Time-Dependent Hamiltonian Simulation Using Discrete-Clock Constructions

Jacob Watkins,[1,*] Nathan Wiebe,[2,3,4] Alessandro Roggero,[5,6,7] and Dean Lee[1]

[1] *Facility for Rare Isotope Beams and Department of Physics and Astronomy, Michigan State University, East Lansing, Michigan 48824, USA*

[2] *Department of Computer Science, University of Toronto, Toronto, Ontario M5S 2E4, Canada*

[3] *Pacific Northwest National Laboratory, Richland, Washington 99354, USA*

[4] *Department of Physics, University of Washington, Seattle, Washington 98195, USA*

[5] *InQubator for Quantum Simulation (IQuS), Department of Physics, University of Washington, Seattle, Washington 98195, USA*

[6] *Dipartimento di Fisica, University of Trento, I-38123 Povo, Trento, Italy*

[7] *INFN-TIFPA Trento Institute of Fundamental Physics and Applications, University of Trento, I-38123 Povo, Trento, Italy*

(Received 7 May 2024; revised 13 August 2024; accepted 1 October 2024; published 31 October 2024)

Compared with time-independent Hamiltonians, the dynamics of generic quantum Hamiltonians $H(t)$ are complicated by the presence of time ordering in the evolution operator. In the context of digital quantum simulation, this difficulty prevents a direct adaptation of many time-independent simulation algorithms for time-dependent simulation. However, there exists a framework within the theory of dynamical systems that eliminates time ordering by adding a "clock" degree of freedom. In this work, we provide a computational framework, based on this reduction, for encoding time-dependent dynamics as time-independent systems. As a result, we make two advances in digital Hamiltonian simulation. First, we create a time-dependent simulation algorithm based on performing qubitization on the augmented clock system and, in doing so, provide the first qubitization-based approach to time-dependent Hamiltonians that goes beyond Trotterization of the ordered exponential. Second, we define a natural generalization of multiproduct formulas for time-ordered exponentials and then propose and analyze an algorithm based on these formulas. Unlike other algorithms of similar accuracy, the multiproduct approach achieves commutator scaling, meaning that this method outperforms existing methods for physically local time-dependent Hamiltonians with sufficient smoothness. Our work reduces the disparity between time-dependent and time-independent simulation and indicates a step toward optimal quantum simulation of time-dependent Hamiltonians.

DOI: 10.1103/PRXQuantum.5.040316

I. INTRODUCTION

Quantum simulation, an original motivator for the quantum computer [1], has arguably retained its preeminence among known applications for the technology. In the context of closed systems, quantum dynamics are described by a Hamiltonian that can, in general, depend explicitly on time. There now exists a large family of digital Hamitonian simulation algorithms [2–8], each of which successfully compiles complicated quantum dynamics into elementary sequences of gates. This has raised the possibility of solving problems in the domain sciences spanning chemistry [9–12], materials science [13], nuclear and neutrino physics [14–16], field theory [17–19], and beyond. However, not all of these algorithms are effective at simulating time-dependent Hamiltonians due to the presence of time ordering in the exponentials. A notable example is qubitization, which achieves optimal scaling for simulating time-independent Hamiltonians but cannot directly simulate general time-dependent Hamiltonians without significant reduction in performance.

There are a number of efficient quantum algorithms designed for time-dependent simulation. Product formulas are naturally suited for this task and have been characterized extensively in a work by Wiebe, Berry, Høyer, and Sanders [20]. Also, Poulin, Qarry, Somma, and Verstraete have introduced a Monte Carlo approach that, unlike

*Contact author: j.a.watkins@hotmail.com

deterministic Trotter, had no smoothness requirements on the time dependence [21]. Following the introduction of the first "post-Trotter" methods, a simulation method based on a truncated Dyson expansion has been introduced by Kieferove, Scherer, and Berry [22] and has shown exponential improvement over product formulas in terms of accuracy. Subsequent work has introduced a "rescaled" Dyson method and a "continuous" version of QDrift, which have had the advantage of depending on the time-averaged size of the Hamiltonian rather than the maximum size [23]. Using Floquet theory, Mizuta and Fujii have successfully applied qubitization to time-dependent periodic Hamiltonians and demonstrated optimal performance [24]. Despite these advances, there remains a relatively limited set of approaches for time-dependent simulation as compared to the time-independent case.

Our work reduces the discrepancy between time-dependent and time-independent Hamiltonian simulation techniques by providing a computational framework, based on the (t, t') formalism [25,26], for encoding time-dependent Hamiltonian dynamics into a time-independent Hamiltonian. Essentially, this is accomplished by promoting the t in $H(t)$ to a degree of freedom and introducing a new evolution parameter. Unlike existing approaches based on a continuous time parameter [27,28], our clock spaces are all finite dimensional, providing the advantages of direct simulability as well as formal well-behavedness. After proving the accuracy of clock-space evolutions, we propose an algorithm for time-dependent Hamiltonian simulation based on direct simulation of the augmented clock system. We focus on qubitization because of its inability to handle time ordering directly and due to its asymptotically superior performance in the time-independent setting. We perform a rigorous error- and query-complexity analysis starting from a Hamiltonian input as a linear combination of unitaries (LCU), with coefficients computed by an oracle. While our asymptotic bounds do not match simulation lower bounds, we suspect that this may be improved upon with better analysis. Regardless, we provide, to our knowledge, the first nontrivial qubitization algorithm for general time-dependent simulation.

Additionally, we take advantage of the finite-clock-space formalism to propose a natural generalization of multiproduct formulas (MPFs) to the time-ordered setting and offer evidence that these are valid extrapolations to the exact time evolution. We then create an algorithm based on these "time-dependent MPFs" and analyze its performance. The algorithm is, to the authors' knowledge, the most efficient existing quantum algorithm for simulating time-dependent Hamiltonians that also exhibits "commutator scaling." By this, we mean that the algorithm exhibits zero error for a time-independent Hamiltonian with easily simulable and commuting terms. In practice, this constitutes a superpolynomial advantage over the error scaling provided by product-formula methods for solving the same problems. We perform two rudimentary numerical demonstrations that indicate that the approach works as anticipated in reducing product-formula error.

The rest of the paper is laid out as follows. In Sec. II, we summarize our main findings and compare with leading algorithms for time-dependent Hamiltonian simulation. In Sec. III, we review the notions of time evolution, product formulas, the (t, t') formalism, and the MPFs needed for our work, while also introducing some notation regarding the various vector and functional norms. In Sec. IV, we introduce the notion of finite-dimensional clock spaces and show that they encode simulations of $H(t)$ with an appropriate time-independent clock Hamiltonian. We then apply these constructions, first to direct simulation by qubitization in Sec. V and then as a tool for proof of time-dependent MPFs in Sec. VI. We conclude in Sec. VII with remarks on the broader impact of this work and directions for future research.

II. MAIN RESULTS

In Table I, we summarize the query complexity of our proposed algorithms (green) and, for comparison, display several leading algorithms for time-dependent simulation. To facilitate comparison, the Hamiltonian is taken in an LCU model,

$$H(t) = \sum_{i=1}^{L} \alpha_i(t) U_i, \tag{1}$$

for unitary and Hermitian U_i and with each α_i being real valued. The Trotter, QDrift, and MPF approaches apply for more general linear combination of Hamiltonians (LCH) inputs but Trotter and MPF also have stronger smoothness assumptions on α_i. The various norms are defined in Sec. III and the scalar function Λ characterizes the Hamiltonian "difficulty" at each time in terms of the size of H and its derivatives (see Definition 3). The tilde in "$\widetilde{O}$" indicates the exclusion of subdominant multiplicative logarithmic factors from the complexity.

The qubitization algorithm is obtained from simulating the finite-clock-space construction of Sec. IV. We carry out a procedure for taking the LCU input $H(t)$ and producing a time-independent LCU input on the larger space. Unfortunately, our analysis shows a Trotter-like error term in the query complexity, as can be seen in Table I. Thus, our analysis does not show that our qubitization algorithm improves over Trotter simulation. In Sec. V C, we discuss how this error term might be eliminated with improved analysis and modification of the clock construction. If this term were absent, the query complexity of our approach would match the T and ϵ lower bounds set by the no-fast-forwarding theorem [3,30].

The time-dependent MPFs that we use for simulation do not necessarily require a clock space but do in the

TABLE I. A summary of our results (green) and a comparison with the leading quantum simulation methods for time-dependent Hamiltonians. We assume that $H = \sum_{i=1}^{L} \alpha_i(t) U_i$ for Hermitian unitaries U_i and real-valued $\alpha_i(t)$. Λ is a positive time-dependent function with dimensions of H that quantifies the size of α_i and its derivatives (see Definition 3). $\|\alpha\|_{p,q}$ refers to a nested vector-p and functional q-norm for the coefficients $\alpha = (\alpha_i)_{i=1}^{L}$ and $\|\alpha\|_{p,q}^{\mathrm{rev}}$ indicates that these are taken in the reverse order. The commutator scaling ("CS") here means that the simulation error vanishes in the limit where H is time independent and $[U_j, U_k] = 0$ for all $j, k \in [L]$.

Method	Query complexity	Auxiliary qubits	CS?
Trotter [29]	$O\big(L(\|\Lambda\|_1)^{1+o(1)}/\epsilon^{o(1)}\big)$	0	Yes
QDrift [23]	$O(\|\alpha\|_{1,1}^2/\epsilon)$	0	No
Dyson [23]	$O\big(L^2\|\alpha\|_{1,1}\log(1/\epsilon)\big)$	$\widetilde{O}\big(\log\big(\|\dot{\alpha}\|_{1,1}/\epsilon\big) + \log\big(\|\alpha\|_{1,\infty}T/\epsilon\big)\big)$	No
Qubitization	$\widetilde{O}\big(\|\alpha\|_{\infty,1}^{\mathrm{rev}}T + \log 1/\epsilon\big)$ $+ \max_t \|\dot{H}\| T^2/\epsilon\big)$	$O\big(\log\big(L\|\alpha\|_{\infty,1}^{\mathrm{rev}}T/\epsilon\big)$ $+ \log\big(L\max_t\|\dot{H}\|T^2/\epsilon\big)\big)$	No
MPF	$\widetilde{O}\big(L\|\Lambda\|_1\log^2(1/\epsilon)\big)$	$\widetilde{O}\big(\log(L\log(\|\Lambda\|_1/\epsilon))$ $+ \max\big\{0, \log\big(L\|\Lambda\|_1\|\dot{\alpha}\|_{\infty,\infty}T^2/\epsilon\big)\big\}\big)$	Yes

oracular setting that we consider. Moreover, the discrete clock construction provides a heuristic argument that the formulas provide high-accuracy simulations as in the time-independent case. We articulate the needed fact in Conjecture 1 and outline a possible path to proof using the clock-space formalism. The generalization from standard MPFs is rather intuitive and the numerics of Sec. VI E support the truth of the conjecture. Assuming this, we go on to analyze the error and query complexity of an algorithm based on these formulas, using adaptive time steps to handle harder parts of the simulation with a greater share of resources.

In the query model, the MPF algorithm performs overall comparably to the Dyson method, as can be seen in Table I. However, a detailed comparison of the two methods yields more interesting insight. In favor of MPFs, they exhibit commutator scaling, meaning that they are errorless in the time-independent commuting limit. Additionally, the dependence on the number of terms L is quadratically improved for MPFs compared to the Dyson approach, though it is not as good as for QDrift. With regard to the weaknesses, like product formulas, there are smoothness requirements on H for MPF simulations to work. In particular, there need to be $2m+1$ bounded derivatives for an m-term MPF. Additionally, the MPF scaling goes, loosely speaking, as the "hardest" coefficient, rather than the preferable "average-hardness" in the case of Dyson.

Like the product formulas on which they are based, the complexity of the MPF scales as the derivatives of the Hamiltonian and does not perform well for highly oscillatory dynamics. In such regimes, QDrift-based methods [23,31] or techniques such as qHOP [32] are likely preferable.

III. BASIC IDEAS AND NOTATION

Here, we review the fundamental concepts behind our work and meanwhile introduce notation. The reader is encouraged to read as desired or reference this section and continue to new results, starting in Sec. IV.

A. Nested norms

In our characterization of simulation errors and computational complexity, we will make frequent use of the vector p or functional q-norm and often both in conjunction. For a vector $v \in \mathbb{C}^n$, the vector p-norm is given by

$$\|v\|_p := \left(\sum_{i=1}^{n} |v_i|^p\right)^{1/p}. \tag{2}$$

For a function $f : [a,b] \to \mathbb{C}$, the functional p-norm is analogously defined as

$$\|f\|_p := \left(\int_a^b |f(t)|^p dt\right)^{1/p}, \tag{3}$$

provided that the integral exists. These expressions hold for p positive integers. We also use $p = \infty$ in the standard way:

$$\|v\|_\infty = \max_j |v_j|, \qquad \|f\|_\infty = \sup_{t\in[a,b]} |f(t)|. \tag{4}$$

For a vector-valued function $v : [a,b] \to \mathbb{C}^n$, the norm $\|v\|_{p,q}$ is just the nesting of these two, starting with the vector norm. For integer p and q,

$$\|v\|_{p,q} := \left(\int_a^b \left(\sum_{j=1}^{n} |v_j(t)|^p\right)^{q/p} dt\right)^{1/q}, \tag{5}$$

and similar expressions hold for p or q being ∞. The norm $\|v\|_{p,q}^{\mathrm{rev}}$ ("rev" for "reverse") indicates that the functional

p-norm is taken, followed by the vector q-norm:

$$\|v\|_{p,q}^{\text{rev}} := \left(\sum_{j=1}^{n}\left(\int_a^b |v_j(t)|^p\right)^{q/p}\right)^{1/q}. \tag{6}$$

For example,

$$\|v\|_{\infty,1}^{\text{rev}} = \sum_j \sup_t |v_j(t)|. \tag{7}$$

B. Digital Hamiltonian simulation

According to the postulates of quantum mechanics, a closed system with Hilbert space $\mathcal{H}$ has dynamics that are generated by some self-adjoint operator H on the space, called the *Hamiltonian*. In certain cases of physical interest, the parameters of the physical system may change over time, or one may work in a "noninertial frame" such as an interaction picture. In such instances, a proper description requires that H be a function of time. In saying that $H(t)$ generates the dynamics, we mean that there exists a unique unitary-operator-valued function U, termed the *time-evolution operator*, that solves the following Schrödinger initial-value problem:

$$\begin{aligned} i\partial_t U(t,t_0) &= H(t)U(t,t_0), \\ U(t_0,t_0) &= \mathbb{1}. \end{aligned} \tag{8}$$

(Here and throughout, $\hbar = 1$.) Often, we set $t_0 = 0$ and use $t = T$ to denote the final time of interest. For any initial state $|\psi(t_0)\rangle = |\psi_0\rangle$ (only pure states are needed), one obtains the time-evolved quantum state as $|\psi(t)\rangle = U(t,t_0)|\psi_0\rangle$. Hence, U "propagates" our state in time and is sometimes called the *propagator*.

In the case in which $H(t)$ is a constant function, we say that it is *time independent*. Such behavior naturally arises in systems the dynamical laws of which exhibit time-translation invariance. In this case, the solution U to Eq. (8) takes the expression

$$U(T,0) = e^{-iHT}. \tag{9}$$

We say that U is an *ordinary* (operator) exponential of H. For more arbitrary $H(t)$, in contrast, the solution U is typically written as an *ordered* (operator) exponential of $H(t)$:

$$U(T,0) = \mathcal{T}\exp\left\{-i\int_0^T H(\tau)d\tau\right\}. \tag{10}$$

There are several different ways to understand the meaning of Eq. (10) but, for our purposes, the most insightful is through product integration [33]. Given a family of partitions $\{t_j\}_{j=1}^n$ of the interval $[s,t]$, with maximum width δ_n tending to zero as $n \to \infty$, a solution is given by the product integral

$$\mathcal{T}\exp\left\{-i\int_0^T H(\tau)d\tau\right\} = \lim_{n\to\infty}\prod_{j=1}^{n-1} e^{-iH(t_j)\Delta t_j}, \tag{11}$$

where $\Delta t_j = t_{j+1} - t_j$. One feature of this approach is that, for sufficiently large but finite n, we achieve an approximation that is amenable to simulation by time-independent methods. This is exactly the approach taken for product-formula simulation. However, simulation of this product by LCU or qubitization is generally pointless, since the errors of discretization outweigh any accuracy benefits achieved by using these protocols over Trotter.

Roughly speaking, a Hamiltonian simulation protocol is a method for approximating $U(t,t_0)$ as a quantum circuit. A metric of accuracy $d(U,V)$ is necessary to evaluate the quality of an approximation V against the exact propagator. In this work, we will take $d(U,V) = \|U - V\|$, where $\|\cdot\|$ will denote the spectral norm (also called the induced 2-norm):

$$\|A\| := \max_{v\neq 0}\frac{\|Av\|}{\|v\|}. \tag{12}$$

The norm appearing on the right is the Euclidean norm. The spectral error $\|U - V\|$ should be thought of as a worst-case simulation error for any initial state. For various reasons, such as partial measurements with postprocessing, V may not be unitary but the actual underlying channel will still be a valid quantum operation. The parameter ϵ is called the *error tolerance* and $d(U,V)$ is the *simulation error*. We will refer to ϵ interchangeably as the error, accuracy, or precision of the simulation.

Any decent approximation to $U(t,t_0)$ should become arbitrarily accurate as $t \to t_0$ (and approaches the identity). The quality of the approximation, in the context of product and multiproduct formulas, is typically quoted in terms of a power-law convergence. This is captured by the following definition.

Definition 1. For finite-dimensional $\mathcal{H}$, let $L : [0,T]^2 \to L(\mathcal{H})$. We say that $L_p : [0,T]^2 \to L(\mathcal{H})$ is a pth-order approximation to L if, for all $t \in [0,T)$,

$$\|L(t+\tau,t) - L_p(t+\tau,t)\| \in O(\tau^{p+1}),$$

where τ is taken asymptotically to 0.

As an important example, product formulas approximate operator exponentials of sums, by splitting exponentials to match terms in a power series of error operators. The

simplest example is 1st-order Trotter, with

$$e^{\lambda(A+B)} = e^{\lambda A}e^{\lambda B} + O(\lambda^2). \tag{13}$$

Such splittings are not exact because A and B do not generally commute.

A function $L : [0,T]^2 \to L(\mathcal{H})$ is said to be *symmetric* if it possesses the following "time-reversal symmetry": $L(t_1,t_0) = L^\dagger(t_0,t_1)$. Symmetric operators are closed under addition and scalar multiplication by a real number. They are also closed under multiplication of the form

$$L^{(2)}(t,t_0) := L(t,t')L(t',t_0), \tag{14}$$

for any $t' \in [0,T]$. This symmetry is valuable, because approximation schemes for U involving symmetric operations are ensured to have an odd error series [3]. Consequently, any symmetric formula is of order $2n$ for integer $n > 0$.

C. Continuous clock space

Mathematically, and more broadly than the Hamiltonian setting considered here, the distinction between time-dependent and time-independent systems can be cast as a distinction between autonomous and nonautonomous dynamical systems. Dynamical systems are differential equations in a single evolution parameter t, which can always be expressed as a first-order differential equation,

$$\dot{x} = f(x,t), \qquad x(0) = x_0, \tag{15}$$

possibly by standard reduction-of-order techniques. Here, $x \in \mathbb{R}^n$ consists of n evolution parameters that implicitly depend on time t. It has long been recognized that a simple transformation allows for the reduction of nonautonomous systems to autonomous ones [25]. The trick is to promote t to a coordinate, thereby making $f(x,t)$ satisfy the requirement of only depending on coordinates. Letting τ take the place of the evolution parameter (time), we still want t and τ to be essentially the same. This is supplied by the simple initial-value problem

$$\frac{dt}{d\tau} = 1, \qquad t(0) = 0. \tag{16}$$

With this, we have the following autonomous system:

$$(\dot{x},\dot{t}) = (f(x,t),1), \qquad (x(0),t(0)) = (x_0,0), \tag{17}$$

the solution of which encodes the solution to the original system, given in Eq. (15).

When the dynamics of interest are generated by a classical Hamiltonian $H(q,p,t)$, the autonomous systems are those in which H has no time dependence. Playing the same trick as before and promoting t to a coordinate, we want $dt/d\tau = 1$ as above. By Hamilton's equations, we require a Hamiltonian H_c such that

$$\frac{\partial H_c}{\partial(-E)} = 1, \tag{18}$$

where $-E$ is the conjugate momentum to t (the minus sign chosen so that E represents energy; we anticipate our results). But we also want H_c to reproduce the same dynamics for q and p as for H. This, altogether, implies the choice

$$H_c(q,p;t,E) := H(q,p;t) - E, \tag{19}$$

an operator that may be familiar from Floquet theory in condensed-matter physics. Simply put, the term $-E$ pulls the t coordinate at constant velocity to the right, changing $H(q,p,t)$ just as needed to enact the appropriate effect on the other coordinates and momenta.

For *quantum* Hamiltonians, we can simply employ a standard quantization procedure on H_c:

$$[\hat{t}, -\hat{E}] = iI. \tag{20}$$

A natural choice of Hilbert space for t is $L^2([0,T])$: square integrable functions on the interval of simulation. We then get a representation of t and E as t multiplication and t derivatives, respectively:

$$(\hat{t}\psi)(t) = t\psi(t), \qquad (\hat{E}\psi)(t) = i\partial_t\psi. \tag{21}$$

The augmented "clock" Hamiltonian, so named because of the time coordinate t, has the form

$$H_c = H - i\partial_t, \tag{22}$$

which is essentially a rearranged Schrödinger operator.

The framework presented, sometimes called the (t,t') formalism because of the two distinct "times" in play, finds use in periodically driven quantum systems [26]. But for our purposes, the elimination of an explicit dependence on the evolution parameter in H is most exciting, because it implies that the time-evolution operator requires no time ordering, while still encoding the full time dynamics [27].

D. Multiproduct formulas

MPFs are a generalization of the celebrated product formulas and span two of the great pillars of quantum simulation. The aim of the MPF is to approximate the time-evolution operator U as a linear combination of lower-order Trotter formulas, in such a way that higher-order errors are cancelled [4,34,35]. They are, fundamentally, nothing more than a Richardson extrapolation of a product formula $\mathcal{P}$ to Trotter step size $s \to 0$. This extrapolation is

done to address the primary deficiency of product formulas, which is that the number of exponentials used in the $2n^{\text{th}}$-order formula scales as 5^n. Product formulas, unfortunately, cannot be easily optimized beyond this. As the MPF is a sum of product-formula approximations, the number of error terms in the expansion does not grow exponentially. This allows us to approximate the quantum dynamics using polynomially many, rather than exponentially many, operator exponentials.

We now reference a theorem [35] that justifies the effectiveness of MPFs in the time-independent setting, while also implicitly defining them. Recall that a (multi)product formula $\mathcal{P}(t)$ is order $p \in \mathbb{Z}_+$ if $U(t) - \mathcal{P}(t) \in O(t^{p+1})$.

Theorem 1 (Time-Independent MPFs (Theorem 1 of Ref. [35])). Let H be a bounded time-independent Hamiltonian and let $U_2(t)$ be the second-order Suzuki-Trotter formula for the time-evolution operator $U(t) = e^{-iHt}$. Let $a = (a_1, a_2, \ldots, a_m) \in \mathbb{R}^m$ and $\vec{k} = (k_1, k_2, \ldots, k_m) \in \mathbb{Z}_+^m$. There exist choices of a and $\vec{k}$ such that the multiproduct formula,

$$U_{2,m}(t) := \sum_{j=1}^{m} a_j U_2^{k_j}(t/k_j),$$

is order $2m$ and satisfies

$$\max_j k_j \in O(m^2), \qquad \|a\|_1 \in O(\text{polylog}(m)).$$

The details of the proof can be seen in Ref. [35] but, at a high level, the MPF $U_{2,m}$ is a Richardson extrapolation of U_2 with respect to the Trotter step-size parameter $1/k$. Such an extrapolation is possible for arbitrary m because there exists an error series [36],

$$U_2^k(t/k) - U(t) = \sum_{j=1}^{\infty} E_{2j+1} \frac{t^{2j+1}}{k^{2j}}, \tag{23}$$

with E_{2j+1} independent of k (but not t generically). The existence of this series suffices for a $1/k \to 0$ Richardson extrapolation [37]. In particular, cancellation occurs for coefficients a_j satisfying the following Vandermonde linear system:

$$\begin{pmatrix} 1 & \cdots & 1 \\ {k_1}^{-2} & \cdots & {k_m}^{-2} \\ \vdots & \ddots & \vdots \\ {k_1}^{-2m+2} & \cdots & {k_m}^{-2m+2} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{24}$$

Though the matrix is ill conditioned, this is irrelevant to the matrix inversion, as the inverse Vandermonde matrix admits an analytic solution that may be reasoned from the theory of polynomial interpolation. What matters for our application is the 1-norm $\|a\|_1$ of the coefficients, which serves as our condition number because of the need to amplify an amplitude of size $1/\|a\|_1$ in the LCU procedure. The content of Theorem 1 is that Trotter steps $\vec{k}$ may be chosen such that $\|a\|_1$ is not too large. For time-ordered U, the analysis of Ref. [36] does not carry over, although reasonable "time-dependent" MPFs can be defined heuristically. One of our motivations in constructing a clock space is to be able to eliminate time ordering and rigorously show that these formulas work.

As discussed in Ref. [35], specific choices of k_j can be found numerically to minimize $\|a\|_1$ and this may be the best approach in practice. However, for our analytic results, it will be most appropriate to utilize the specific k_j chosen in the authors' constructive proof of well-conditioned MPFs. Thus, for all results, we will take the powers k_j as follows:

$$k_j = \left\lceil \frac{\sqrt{8}m}{\pi} \left| \sin\left(\frac{\pi(2j-1)}{8m} \right) \right|^{-1} \right\rceil, \qquad j = 1, \ldots, m. \tag{25}$$

We will use these same coefficients even in the time-dependent MPFs to be introduced in Sec. VI. For error analysis, it will be useful to have simple concrete bounds on k_j. We can achieve this by noting that $\sin(x) \le x$ and $\sin(x) \ge 4x/5$ for $x \in [0, 1]$. This gives the lower bound

$$k_j \ge \left\lceil \frac{8^{3/2}m^2}{\pi^2(2j-1)} \right\rceil \ge \left\lceil \frac{8^{3/2}m^2}{\pi^2(2m-1)} \right\rceil > \frac{\sqrt{128}m}{\pi^2} > m \tag{26}$$

and the upper bound

$$k_j \le \left\lceil \frac{5 \times 8\sqrt{8}m^2}{4(2j-1)\pi^2} \right\rceil \le \left\lceil \frac{5 \times 8\sqrt{8}m^2}{4\pi^2} \right\rceil < 3m^2. \tag{27}$$

Note the consistency of Eq. (27) with the big-O scaling of Theorem 1.

IV. FINITE-DIMENSIONAL CLOCK SPACES

We now introduce our finite-dimensional clock space, which we will sometimes call the "clock register" to distinguish it from the continuous version. We discretize the clock variable t into $N_c = N_p \times N_q$ basis states, where $N_p \in \mathbb{Z}_+$ will represent the number of "Trotter steps" used in the simulation. Each is further divided into $N_q \in \mathbb{Z}_+$ steps for reasons that will be discussed shortly. We label these orthonormal basis states $|j\rangle$ for $j \in [0, N_c - 1] \cap \mathbb{Z}$. We will find it useful to consider, for our purposes, only periodic Hamiltonians. This is natural since generators of translation, like the clock-momentum operator E from Sec. III C, act most naturally on $\mathbb{R}$ or with periodic boundary conditions. Nonperiodic Hamiltonians can

be accommodated by a simple reflection, defining $H(T+t) := H(T-t)$ for $t \in [0,T]$. In our work below, we will want $H(t)$ to be a differentiable bounded function within the grid points and although the reflection introduces nonsmoothness, we can simply take one of the grid points to be the midpoint of the simulation.

For simplicity, and for lack of a compelling alternative, we will take these grid points $(t_j)_{j=0}^{N_c-1}$ to be uniformly spaced over the interval $[0,T]$: $t_j = Tj/N_c$ (taking N_c to be an even integer, so that the midpoint requirement discussed directly above is satisfied). We let $\delta t := T/N_c$ denote the grid width. We also take the natural discretization of $H(t)$ onto the clock space:

$$H(t) \mapsto C(H) := \sum_{j=0}^{N_c-1} H_j \otimes |j\rangle\langle j|, \tag{28}$$

where $H_j := H(t_j)$. Observe that $C(H)$ has no dependence on the evolution parameter; it is time independent. The notation $C(H)$ is used to suggest a controlled operation, where the control is on the clock register.

Choosing the appropriate discretization of E is somewhat more tricky, though the choice appears obvious in hindsight. Since E acts as a derivative, it makes sense to take the discretized version to be a finite-difference operator. For example,

$$\Delta := -i\frac{U_+ - U_-}{2\delta t}, \tag{29}$$

where U_+ is the shift operator defined by $U_+|j\rangle = |j+1\rangle$ and $U_- = U_+^\dagger$ is the backward shift (all increments taken modulo N_c). This is the approach that we ultimately take. However, we note that we began by considering a distinct approach via the logarithm of the translation operator,

$$\tilde{\Delta} = i \log U_+. \tag{30}$$

While apparently sensible, given the analogous relation between E and shifts on the clock space, this operator is not nicely behaved. For example, its commutator with the "position operator" $\sum_j t_j |j\rangle\langle j|$, rather than being near identity, has long off-diagonal tails. This behavior may be of independent interest but from now on we will concern ourselves with Δ as the discrete version of $-E$.

With these choices, our full clock Hamiltonian becomes

$$H_c := C(H) - \Delta. \tag{31}$$

Already, we can show that some reasonable properties carry over to this setting.

Lemma 1. In the above notation, let $H : [0,T] \to \mathrm{Herm}(\mathcal{H})$ be a time-dependent Hamiltonian on a finite-dimensional vector space $\mathcal{H}$. Then,

$$[\Delta, C(H)] = i\,\mathrm{Re}\left(U_+ \sum_j \frac{H_{j+1} - H_j}{\delta t} \otimes |j\rangle\langle j|\right),$$

where $\mathrm{Re}(A) := (A + A^\dagger)/2$ denotes the Hermitian part of A. If H is differentiable in each subinterval with bounded derivative, then we further have

$$\|[\Delta, C(H)]\| \le \max_{t\in[0,T]} \|\dot{H}(t)\|.$$

We remark here on the connections to the canonical commutation relation $[f(x), p] = if'(x)$. The additional shift by U_+ is a relatively small deviation from a finite-difference approximation being performed on the Hamiltonian. The proof is relatively straightforward and is provided in Appendix B.

Having defined the clock space and Hamiltonian, we wish to prepare a suitable initial state. A seemingly adequate and natural choice is to take $|\psi_0\rangle \otimes |0\rangle$, where $|\psi_0\rangle$ is the initial state of the system of interest and $|0\rangle$ is the clock state at the initial time $t = 0$. However, problems immediately arise, which can be traced to the fact that the continuous version of $|0\rangle$ is $\delta(t)$, which is not a normalizable state vector. This formal problem finds its way into the discrete setting, in that the finite difference Δ does not properly compute a derivative of $|0\rangle$. Thus, Δ fails to translate $|0\rangle$ properly into later times and the time-dependent simulation fails.

To fix this issue, we take a cue from the continuous setting, where the best we can do is take a wave packet of small enough width to suit our purposes. For simplicity, this wave packet may as well be Gaussian, with some width σ to be chosen with care. Thus, we introduce Gaussian functions

$$\phi_\mu(t;\sigma) = \frac{1}{\sqrt{\mathcal{N}}} e^{-|t-\mu|_c^2/\sigma^2}. \tag{32}$$

of width $\sigma \in \mathbb{R}_+$ and center $\mu \in [0,T)$. Here, $|\cdot|_c$ is the shortest distance to 0 modulo T,

$$|t|_c := \min\{|t|, |T-t|\}, \tag{33}$$

so that, with 0 and T identified, ϕ_μ is smooth everywhere except for $\mu + T/2$ modulo T. Though we specify Gaussians, our findings should generally apply to other localized packets provided that they have bounded second derivative. The coefficient $\mathcal{N} \in \mathbb{R}_+$ is chosen such that the

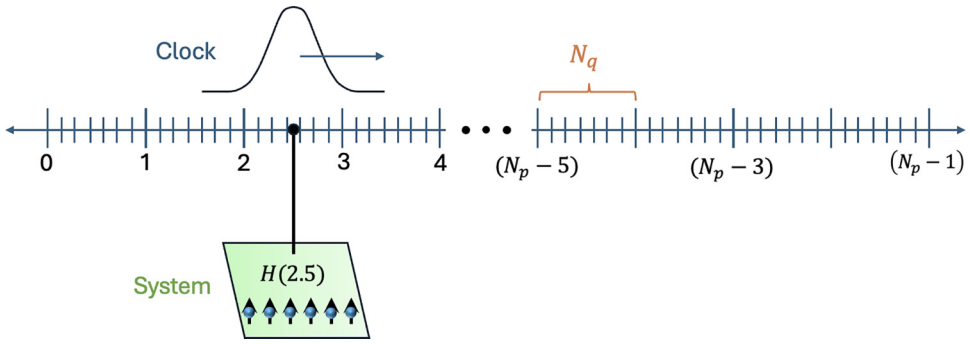

FIG. 1. A schematic of the discrete-clock Hilbert space. The clock register has an initially prepared Gaussian state that is translated uniformly under the clock Hamiltonian. Its location controls the Hamiltonian applied to the system of interest. The Hamiltonian varies little over each of the N_p large steps and the Gaussian is wide compared to the N_q subdivisions within each large step.

discretized vector

$$|\phi_\mu\rangle = \sum_j \phi_\mu(t_j;\sigma)|j\rangle \tag{34}$$

is normalized in the Euclidean sense (i.e., a quantum state vector). Technically, $\mathcal{N}$ has some dependence on μ but in our case we will only consider $\mu = t_j$ for some j, in which case $\mathcal{N}$ only depends on parameters such as N_c and σ. Because of this choice, we will more simply write $|\phi_j\rangle := |\phi_{t_j}\rangle$.

We are now ready to more clearly elucidate the overall strategy of the clock-space construction. In Fig. 1, we give a schematic of the relevant components. As stated above, each of the N_p should be thought of as a single Trotter step in the evolution under $H(t)$. The N_q subintervals ensure that δt is sufficiently small such that the approximation of Δ to a derivative of ϕ_j holds. In particular, we will desire $\sigma \gg \delta t$. On the other hand, we want the variation of H within the envelope of ϕ_j to be small. That is, we want $\sigma < T/N_p$. Because, presumably, we have chosen each Trotter step sufficiently small, this ensures that H is approximately constant over the bulk of $|\phi_j\rangle$. Of course, we will want to ensure all of the above conditions with as few resources, such as clock-register states, as possible.

We now begin to characterize the simulation error in using the clock space for approximating $U(T,0)$. First, it will be helpful to have a characterization of the size of the normalization $\mathcal{N}$.

Lemma 2. In the above notation, the normalization constant $\mathcal{N} \in \mathbb{R}_+$ for Gaussian states $|\phi_j\rangle$ peaked at $\mu = t_j$ satisfies

$$\frac{1}{\sqrt{\mathcal{N}}} \in O(\sqrt{\delta t/\sigma}).$$

The proof is provided in Appendix B. With this technical lemma in hand, we turn to showing that Δ indeed acts as a generator of translations on the clock space for $|\phi_j\rangle$, provided that σ is large relative to δt and that the Gaussian is not truncated by small T.

Lemma 3. In the notation introduced in this section, for any $m \in \mathbb{Z}_+$, we have

$$e^{i\Delta m\delta t}|\phi_j\rangle = |\phi_{j+m}\rangle + O\left(m(\delta t/\sigma)^2 + m\sqrt{\delta t/\sigma}\, e^{-(T/2\sigma)^2}\right),$$

where the asymptotics O are understood to be taken as $\delta t/\sigma \to 0$ and $\sigma/T \to 0$.

Proof. Performing a first-order Taylor expansion of the exponential,

$$e^{i\Delta\delta t}|\phi_j\rangle = |\phi_j\rangle + i\delta t\Delta|\phi_j\rangle + R_1(\delta t)|\phi_j\rangle, \tag{35}$$

where R_1 is the Taylor-remainder operator

$$R_1(\delta t) = \delta t \int_0^{\delta t} \frac{\partial^2}{\partial\tau^2} e^{i\Delta\tau} d\tau = -\int_0^{\delta t} e^{i\Delta\tau} d\tau\,(\delta t\Delta^2). \tag{36}$$

Thus, the error can be bounded, via the triangle inequality for integrals, as

$$\|R_1(\delta t)|\phi_j\rangle\| \le \delta t^2 \|\Delta^2|\phi_j\rangle\|. \tag{37}$$

The action of Δ on discretized functions $|g\rangle$ of the clock space is given by

$$\begin{aligned}\Delta\,|g\rangle &= -i\sum_{j=0}^{N_c-1} g(t_j)\left(\frac{|j+1\rangle - |j-1\rangle}{2\delta t}\right)\\ &= i\sum_j \frac{g(t_{j+1}) - g(t_{j-1})}{2\delta t}|j\rangle\\ &= i\,|D_{\delta t}g\rangle\,.\end{aligned} \tag{38}$$

Here, $D_{\delta t}f(x) := ((f(x+\delta t) - f(x-\delta t))/2\delta t)$ is the symmetric finite difference of half width δt at point x. Thus,

$\Delta^2|\phi_j\rangle = -|D_{\delta t}^2\phi_j\rangle$. We consider the error of this finite difference in terms of an approximation to the derivative for values of t within $T/2 - 2\delta t$ of t_j in circle distance. On this part of the domain, $\phi_j(t \pm 2\delta t)$ is smooth; hence

$$|D_{\delta t}^2\phi_j\rangle = |\partial_t^2\phi_j\rangle + O(\delta t^2\phi_j^{(4)}), \tag{39}$$

where the superscript "(4)" indicates a fourth derivative. Near the edge of the Gaussian, the second-derivative property does not hold; however, these parts of the state vector have amplitude that is on the order $O(\mathcal{N}^{-1/2}e^{-(T/2\sigma)^2})$, which by Lemma 2 is $O(\sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2})$. This gets multiplied by δt^{-2} due to the second finite difference $D_{\delta t}$ being taken. Taking the two sources independently as an upper bound, we have

$$\begin{aligned}\|\Delta^2|\phi_j\rangle\| &= \|\,|\partial_t^2\phi_j\rangle\| + O\left(\delta t^2/\sigma^4 + (\sigma\delta t^3)^{-1/2}e^{-(T/2\sigma)^2}\right)\\ &\in O\left(1/\sigma^2 + \delta t^2/\sigma^4 + (\sigma\delta t^3)^{-1/2}e^{-(T/2\sigma)^2}\right),\end{aligned} \tag{40}$$

where σ^{-4} comes from the four derivatives of the Gaussians. Thus, the total Taylor remainder may be upper bounded using Eq. (36) as

$$\|R_1(\delta t)|\phi_j\rangle\| \in O\left((\delta t/\sigma)^4 + \sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2}\right). \tag{41}$$

To complete the proof, we return to the linear Taylor expansion in Eq. (35). Using similar reasoning to above,

$$\begin{aligned}|\phi_j\rangle + i\delta t\Delta|\phi_j\rangle &= |\phi_j\rangle - \delta t|D_{\delta t}\phi_j\rangle\\ &= |\phi_j\rangle - \delta t|\partial_t\phi_j\rangle\\ &\quad + O\left((\delta t/\sigma)^2 + \sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2}\right).\end{aligned} \tag{42}$$

Finally, what remains is a linear approximation to $|\phi_{j+1}\rangle$, with error also $(\delta t/\sigma)^2$. Keeping only the leading terms, note that the Taylor-remainder error is subdominant. Altogether,

$$e^{i\Delta\delta t}|\phi_j\rangle = |\phi_{j+1}\rangle + O\left((\delta t/\sigma)^2 + \sqrt{\delta t/\sigma}e^{-(T/2\sigma)^2}\right). \tag{43}$$

So far, we have proved the result for $m = 1$. The full result follows by noting that $e^{i\Delta m\delta t} = (e^{i\Delta\delta t})^m$ and taking, as upper bound, m times the error of a single step. ■

We note that the error in Δ generating translations comes from two sources: the discretization at small scales and the boundary effects at large scales. We might name these, in the language of lattice field theory, ultraviolet and infrared truncation effects, respectively.

Our next intermediate result will be concerned with the evolution of $C(H)$ controlled on the Gaussian state $|\phi_j\rangle$. We want the result to be, approximately, an evolution under $H(t_j)$ on the main register of interest. In what follows, it will be convenient to take $\tau := T/N_p$ as the time duration of a larger subdivision of steps.

Lemma 4. Let $H : [0, T] \to \mathrm{Herm}(\mathcal{H})$ be a bounded differentiable function with bounded derivative. For any $\eta \in \mathbb{R}$, we have

$$e^{-iC(H)\eta}|\psi\rangle|\phi_j\rangle = e^{-iH(t_j)\eta}|\psi\rangle|\phi_j\rangle + O\left(\eta\tau\max_{t\in[0,T]}\|\dot{H}(t)\| + (1+\eta\max_{t\in[0,T]}\|H(t)\|)e^{-\tau^2/4\sigma^2}\right),$$

where $\tau := T/N_p$.

Proof. We begin by grouping the terms of $C(H)$ into two chunks: one with significant overlap with the Gaussian; the other with small overlap. Specifically, we take $C(H) = H_{\mathrm{av}} + H_\perp$, with

$$\begin{aligned}H_{\mathrm{av}} &:= \sum_{k=j-N_q/2}^{j+N_q/2-1} H_k \otimes |k\rangle\langle k|\\ H_\perp &:= C(H) - H_{\mathrm{av}}.\end{aligned} \tag{44}$$

Because H_{av} and $H_\perp$ commute, we can Trotterize with no error:

$$e^{-iC(H)\eta}|\psi\rangle|\phi_j\rangle = e^{-iH_\perp\eta}e^{-iH_{\mathrm{av}}\eta}|\psi\rangle|\phi_j\rangle. \tag{45}$$

We will show that the H_{av} term gives approximately $H(t_j)$, while $H_\perp$ acts as approximately the identity (with the right parameter values).

First, consider $e^{-iH_{\mathrm{av}}\eta}$. Define $P_j = \sum_k |k\rangle\langle k|$ as the projector onto the clock states on which H_{av} has support

($k \in \mathbb{Z} \cap [j - N_q/2, j + N_q/2 - 1]$). We have

$$\|e^{-iH_{\text{av}}\eta} - e^{-iH_j \otimes P_j \eta}\| \leq \eta \|H_{\text{av}} - H_j \otimes P_j\|. \tag{46}$$

Meanwhile,

$$\begin{aligned}\left\|H_{\text{av}} - H_j \otimes P_j\right\| &= \left\|\sum_{k=j-N_q/2}^{j+N_q/2-1} (H_k - H_j) \otimes |k\rangle\langle k|\right\| \\ &= \max_k \|H_k - H_j\|. \end{aligned} \tag{47}$$

By a simple Taylor bound, $\|H_k - H_j\| \leq (\tau/2) \max_t \|\dot{H}(t)\|$, where the "max" is over $[t_k, t_j]$ (taking the appropriate ordering of t_j, t_k if needed). We can therefore say

$$\|e^{-iH_{\text{av}}\eta} - e^{-iH_j \otimes P_j \eta}\| \leq \eta\tau \max_{t\in[0,T]} \|\dot{H}\|, \tag{48}$$

so that, up to this error, we can replace a simulation by H_{av} with $H_j \otimes P_0$. Moving on to this situation, we have

$$e^{-iH_j \otimes P_0 \eta}|\psi\rangle \otimes |\phi_j\rangle = e^{-iH_j \eta}|\psi\rangle P_j |\phi_j\rangle + |\psi\rangle(I - P_j)|\psi_j\rangle. \tag{49}$$

Thinking of $\sigma < \tau$ and taking τ/σ increasing, we have $P_0|\phi_j\rangle = |\phi_j\rangle + O\left(e^{-\tau^2/4\sigma^2}\right)$. Thus,

$$\begin{aligned} &e^{-iH_j \otimes P_0 \eta}|\psi\rangle|\phi_j\rangle \\ &\quad = e^{-iH_j \eta}|\psi\rangle|\phi_j\rangle + O\left(\eta\tau \max_{t\in[0,T]} \|\dot{H}\| + e^{-\tau^2/4\sigma^2}\right). \end{aligned} \tag{50}$$

For the remainder of the proof, take $|\psi'\rangle = e^{-iH_j \eta}|\psi\rangle$ for notational convenience. We now consider the action of $H_\perp$ on the remaining state, which we anticipate to be small. First,

$$\left\|e^{-iH_\perp \eta}|\psi'\rangle|\phi_j\rangle - |\psi'\rangle|\phi_j\rangle\right\| \leq \eta \|H_\perp|\psi'\rangle|\phi_j\rangle\|. \tag{51}$$

Let $\mathcal{J}$ be an index set for all the time steps included in the summation H_{av}. We have

$$\begin{aligned}\|H_\perp|\psi'\rangle|\phi_j\rangle\| &= \left\|\sum_{k\notin\mathcal{J}} H_k|\psi'\rangle|k\rangle\langle k|\phi_j\rangle\right\| \\ &\leq \sqrt{\sum_{k\notin\mathcal{J}} \frac{1}{\mathcal{N}} e^{-2|t_j - t_k|_c^2/\sigma^2} \|H_k\|^2}. \end{aligned} \tag{52}$$

Employing a Hölder inequality on the inner product, followed by Lemma 2,

$$\begin{aligned} &\sqrt{\sum_{k\notin\mathcal{J}} \frac{1}{\mathcal{N}} e^{-2|t_j - t_k|_c^2/\sigma^2} \|H_k\|^2} \\ &\quad \leq \max_{k\notin\mathcal{J}} \|H_k\| \sum_{k\notin\mathcal{J}} \frac{e^{-|t_j - t_k|_c^2/\sigma^2}}{\mathcal{N}} \\ &\quad \in O\left(\max_t \|H(t)\|(\delta t/\sigma) \sum_{k=N_q/2}^{\infty} e^{-k^2\delta t^2/\sigma^2}\right). \end{aligned} \tag{53}$$

Following a similar procedure to before, we convert to an error function "erf" and take an exponential upper bound. Doing so gives

$$\|H_\perp|\psi'\rangle|\phi_j\rangle\| \in O\left(\max_{t\in[0,T]} \|H(t)\| e^{-\tau^2/2\sigma^2}\right). \tag{54}$$

Thus, $e^{-iH_\perp \eta}$ acts trivially on this state up to $O\left(\eta \max_t \|H(t)\| e^{-(\tau/2\sigma)^2}\right)$.

Combining the errors together, we take the widest exponential $e^{-\tau^2/4\sigma^2}$ as a simple upper bound for all exponentials that appear. Putting all the error sources together gives us the result of the lemma statement. ■

With the previous two lemmas, we have the ingredients needed for a clock-space simulation: controlled operations and time shifts. We combine them to show that our clock space indeed encodes time-dependent dynamics.

Theorem 2. Let $H : [0, T] \to \text{Herm}(\mathcal{H})$ be a time-dependent Hamiltonian on a finite-dimensional vector space $\mathcal{H}$, such that $H(t)$ as a function is bounded and differentiable with bounded derivative. Then, the clock Hamiltonian, with Gaussian input $|\phi_0\rangle$, approximately applies the time-evolution operator $U(T, 0)$ to an initial state $|\psi_0\rangle \in \mathcal{H}$. To be precise,

$$e^{-iH_c T}|\psi_0\rangle|\phi_0\rangle = (U(T,0)|\psi_0\rangle)\,|\phi_0\rangle + O\left(T\delta t/\sigma^2 + \sqrt{N_c T/\sigma}\, e^{-T^2/4\sigma^2} + \max_t \|\dot{H}\| \frac{T^2}{N_p} + e^{-\tau^2/4\sigma^2}(N_p + \max_t \|H\| T)\right).$$

Proof. Let $\tau = T/N_p$. We begin with a first-order Trotterization of H_c into N_p steps:

$$e^{-iH_cT} = \left(e^{i\Delta\tau}e^{-iC(H)\tau}\right)^{N_p} + O\left(\max_{t\in[0,T]}\|\dot{H}(t)\|\frac{T^2}{N_p}\right). \tag{55}$$

With initial state $|\psi_0\rangle|\phi_0\rangle$, combining Lemmas 4 and 3 gives the following error for a single Trotter step:

$$e^{i\Delta\tau}e^{-iC(H)\tau}|\psi_0\rangle|\phi_0\rangle = e^{-iH_0\tau}|\psi_0\rangle|\phi_{N_q}\rangle + O\left(\tau\delta t/\sigma^2 + \sqrt{N_q\tau/\sigma}e^{-T^2/4\sigma^2} + \tau^2\max_t\|\dot{H}\| + (1+\tau\max_t\|H\|)e^{-\tau^2/4\sigma^2}\right). \tag{56}$$

Thus, after all N_p steps, we can multiply the single-step error above to obtain an upper bound of

$$\left(e^{i\Delta\tau}e^{-iC(H)\tau}\right)^{N_p}|\psi_0\rangle|\phi_0\rangle = e^{-iH_{N_q}(N_p-1)\tau}\ldots e^{-iH_{N_q}\tau}e^{-iH_0\tau}|\psi_0\rangle|\phi_0\rangle + O\left(T\delta t/\sigma^2 + \sqrt{N_cT/\sigma}e^{-T^2/4\sigma^2} + \max_t\|\dot{H}\|\frac{T^2}{N_p} + e^{-\tau^2/4\sigma^2}(N_p + \max_t\|H\|T)\right). \tag{57}$$

The right-hand side, without the error, is a first-order Suzuki Trotter splitting, which approximates $U(T,0)$ to order $\max_{t\in[0,T]}\|\dot{H}(t)\|T^2/N_p$. This can be absorbed into the third term of the big O. This gives the result stated in the theorem. ■

With this result in hand, we now show that the parameters (N_p, N_q, σ) of the clock can be chosen such that any desired degree of approximation to $U(T,0)$ can be achieved.

Theorem 3. In the context of the previous theorem, for any $\epsilon \in \mathbb{R}_+$, there exist clock parameters (N_p, N_q, σ) such that

$$\left\|e^{-iH_cT}|\psi_0\rangle|\phi_0\rangle - U(T,0)|\psi_0\rangle|\phi_0\rangle\right\| < \epsilon,$$

with (N_p, N_q) scaling as

$$N_p \in \Theta\left(\max_{t\in[0,T]}\|\dot{H}\|\frac{T^2}{\epsilon}\right), \quad N_q \in \Theta\left(\frac{\max_t\|\dot{H}\|T^2}{\epsilon^2}x^2\right),$$
$$\sigma \in \Theta\left(\frac{\epsilon}{\max_t\|\dot{H}\|Tx}\right).$$

Here,

$$x := \sqrt{\log\left(\frac{\Gamma\, T}{\epsilon}\right)}$$
$$\Gamma := \max\left\{\max_{t\in[0,T]}\|\dot{H}\|T, \epsilon\max_{t\in[0,T]}\|H\|\right\}.$$

In particular, there exists a sequence $(N_p(j), N_q(j), \sigma(j))$ of clock-space parameters, such that

$$\lim_{j\to\infty}\mathrm{Tr}_c(e^{-iH_cT}|\psi_0\rangle|\phi_0\rangle) = U(T,0)|\psi_0\rangle,$$

where Tr_c is a partial trace over the clock register and $\mathrm{Tr}_c(|\Psi\rangle) := \mathrm{Tr}_c(|\Psi\rangle\langle\Psi|)$.

Proof. To ensure that a total error within ϵ is achievable, it suffices to ensure that each of the five terms constituting the error in Theorem 2 is within $O(\epsilon)$ independently. From the onset, we will choose $N_p \in \Theta\left(\max_{t\in[0,T]}\|\dot{H}\|T^2/\epsilon\right)$ to satisfy the third term.

We next move to understand the necessary σ scaling. We parametrize it as

$$\sigma = \tau/x, \tag{58}$$

with the hope that x can be chosen to increase slowly (i.e., that the Gaussian states have width only slightly smaller than the Trotter step size). For this, we focus on the last two terms, since they have no N_q dependence (which will set the smallest scales). We seek

$$\frac{\max_t\|\dot{H}\|T^2}{\epsilon}e^{-x^2/4} \in O(\epsilon), \qquad \max_t\|H\|Te^{-x^2/4} \in O(\epsilon), \tag{59}$$

which can be satisfied provided that x is asymptotically lower bounded as

$$x^2 \in \Omega\left(\log\max\left\{\frac{\max_t\|\dot{H}\|T^2}{\epsilon^2}, \frac{\max_t\|H\|T}{\epsilon}\right\}\right) = \Omega\left(\log(\Gamma\, T/\epsilon)\right). \tag{60}$$

This sets the scaling for σ.

We move next to the first term to fix N_q, since the second term is expected to be quite small. We require $T\delta t/\sigma^2 \in O(\epsilon)$, which is equivalent to

$$\frac{N_p x^2}{N_q} \in O(\epsilon). \tag{61}$$

Therefore, there exists an $N_q \in \Theta\left(N_p x^2/\epsilon\right)$, satisfying the bound. All that remains is the second term, the contribution of which can be easily shown to be subdominant compared to the other sources. Therefore, the choice of parameter scaling suffices to achieve the desired error ϵ.

We have shown that any desired precision ϵ for dynamical simulation can be accommodated for by an appropriate choice of clock-space parameters. Taking a sequence $\epsilon_j \to 0$, we see that there exists a sequence of clock-space evolutions the limit of which, restricted to the main register, is $U(T,0)$. ■

We have thus shown that finite-clock-space constructions exist which, for $H(t)$ differentiable on $[0,T]$, approximate the dynamics of H to arbitrary precision. One expects that the differentiability condition can be somewhat relaxed, since it does not appear in the continuous setting. Any improvements in error analysis here will enhance the performance guarantees of the qubitization algorithm presented in Sec. V.

V. TIME-DEPENDENT QUBITIZATION

In Sec. IV, we have developed a clock-space construction that encodes a time-dependent Hamiltonian as a time-independent one on an augmented finite-dimensional space. The removal of time ordering using a clock register opens the door for quantum algorithms for time-independent Hamiltonian simulation to simulate the full clock-system dynamics directly. In particular, qubitization is an asymptotically optimal [6] simulation method that can only be applied to time-independent H. In this section, we propose the simulation of time-dependent Hamiltonians using qubitization on the augmented system. To be concrete, we will work with an input model in which $H(t)$ is a linear combination of fixed unitaries with time-varying coefficients. This describes, e.g., Pauli matrices on n qubits with fluctuating coefficients.

A. Overview

In Fig. 2, we give a flowchart for the full qubitization procedure that we propose for time-dependent simulation. We take our main register, encoding the quantum system of interest, and append n_c qubits to provide a clock register of size $N_c = 2^{n_c}$. The product state $|\psi_0\rangle|\phi_0\rangle$ is prepared on the joint register, where $|\psi_0\rangle$ is the initial state of the main register and $|\phi_0\rangle$ is a Gaussian as per Eq. (34). Many protocols for preparing Gaussian states exist [38–42]. For our purposes, we will simply refer to the approach by Kitaev and Webb [43,44]. The Gaussian in our application has non-negligible support over $O(N_q)$ clock states and the authors' algorithm scales polynomially in the number of qubits $n_q = \lceil \log N_q \rceil$ over the Gaussian. This cost is negligible compared to the other simulation costs that we will discuss presently. We neglect the cost of initial-state preparation $|\psi_0\rangle$, despite this step being very important, because the cost is highly application dependent and hard to generally quantify.

After the initial state is prepared, we seek to employ qubitization to approximate $e^{-iH_c T}$ on the full register. Given $H(t)$ in LCU form, we need to express H_c in LCU form as well, which is not immediate. This is done through several applications of the signature-matrix decomposition (see Appendix C). We also truncate Δ at high frequencies to reduce computational cost, with little loss in accuracy. Details of the LCU decomposition are provided in Sec. V B.

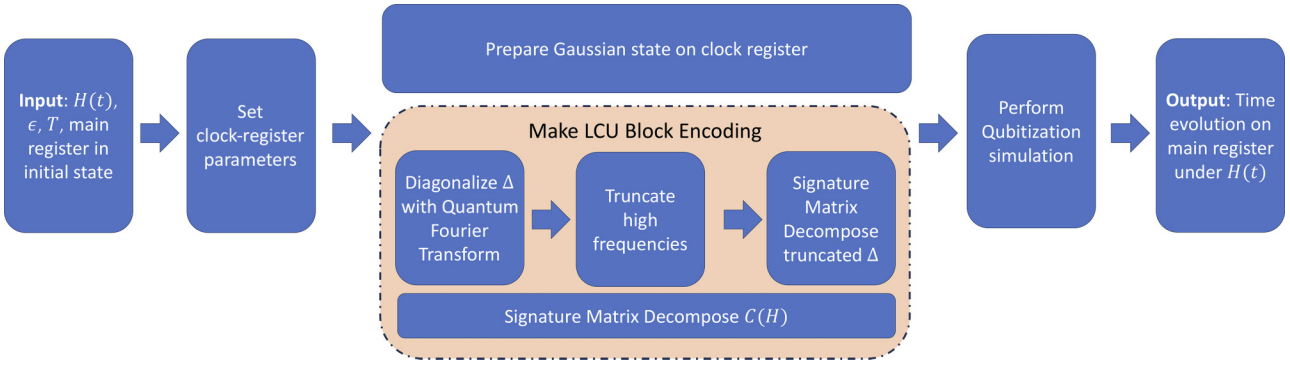

FIG. 2. A schematic work flow of the qubitization-simulation algorithm. The vertically aligned boxes correspond to steps that can be performed in parallel. Given the basic simulation parameters, a clock register of given size needs to be allocated and the Gaussian state $|\phi_0\rangle$ needs to be prepared. Meanwhile, the clock Hamiltonian needs to be brought into LCU form so that it may be block encoded. A qubitization simulation results in an approximate evolution under the original $H(t)$ of interest.

Once H_c is in LCU form, the "select" SEL and "prepare" PREP circuits may be constructed to block encode H_c as

$$H_c/\|c\|_1 = (\langle 0|\,\mathrm{PREP}^\dagger \otimes \mathbb{1})\mathrm{SEL}(\mathrm{PREP}\,|0\rangle \otimes \mathbb{1}), \tag{62}$$

where $\|c\|_1$ is the 1-norm of the LCU coefficients. Standard qubitization can now be done on this block-encoded Hamiltonian [6]. The subsequent Sec. V B will show that the PREP circuit must create a "quasiuniform" distribution over some number N of states, in the sense that, on the LCU auxiliary register,

$$|\mathrm{PREP}\rangle = \sum_{j=1}^{K-1} \sqrt{\delta}\,|j\rangle + \sum_{j=K}^{N} \sqrt{\delta'}\,|j\rangle\,, \tag{63}$$

where δ, δ', K, and N are determined by parameters of simulation. Meanwhile, the SEL circuit will need to apply controlled U_i operations, where U_i is a unitary in the $H(t)$ decomposition, and controlled signature matrices. These second operations can be done with classical reversible comparator circuits implemented quantumly. Each SEL will also require a quantum Fourier transform and its inverse on the clock register.

B. LCU block encoding

We assume that $H(t)$ is of the form

$$H(t) = \sum_{i=1}^{L} \alpha_i(t) U_i, \tag{64}$$

where the U_j are Hermitian and unitary (e.g., n-qubit signed Pauli operators) and the $\alpha_j(t)$ are non-negative real-valued functions on $[0, T]$. When we discretize, the coefficients $\alpha_{ij} := \alpha_i(t_j)$ will be particularly important. Expanding out $C(H)$ from Eq. (28) using Eq. (64),

$$\begin{aligned} C(H) &= \sum_{j=0}^{N_c-1} \left(\sum_{i=1}^{L} \alpha_{ij} U_i \right) \otimes |j\rangle\langle j| \\ &= \sum_{i=1}^{L} U_i \otimes D_i, \end{aligned} \tag{65}$$

where

$$D_i := \sum_{j=0}^{N_c-1} \alpha_{ij}\,|j\rangle\langle j| \tag{66}$$

is a diagonal operator on the clock register. Let $\Lambda_i(\delta) := \lceil \max_j |\alpha_{ij}|/\delta \rceil$. Using a signature-matrix decomposition (see Appendix C), we can write

$$D_i = \sum_{k=1}^{\Lambda_i(\delta)} \delta S_{ik}(\delta) + O(\delta) \tag{67}$$

for $\delta > 0$, where

$$S_{ik}(\delta) = \sum_{j=0}^{N_c-1} (-1)^{k[k>\alpha_{ij}/\delta]}\,|j\rangle\langle j| \tag{68}$$

and $[P]$ is the Boolean function for proposition P, with $[\mathrm{True}] = 1$ and $[\mathrm{False}] = 0$. Thus, we obtain an LCU decomposition of $C(H)$ as

$$C(H) = \delta \sum_{i=1}^{L} \sum_{k=1}^{\Lambda_i(\delta)} U_i \otimes S_{ik}(\delta) + O(L\delta). \tag{69}$$

The prepare circuit PREP is simple enough because the linear combination is uniform. Therefore, it can be accomplished using a Hadamard gate on each of

$$n_{C(H)} \in O\left(\log \sum_{i=1}^{L} \max_j |\alpha_{ij}|/\delta \right) \tag{70}$$

auxiliary qubits needed for a binary encoding. The unitaries $U_i \otimes S_{ik}(\delta)$ can be selected using two different SEL circuits: one for the original U_i (presumed available to us) and one for the signature matrices $S_{ik}(\delta)$. These unitaries can be constructed using classical comparator circuits provided that each α_{ij} is computable.

We now turn our attention to Δ, defined in Eq. (29). Although already in LCU form, the coefficient has size $2/\delta t$ and is too large to be desirable. However, Δ may be truncated at high frequencies without significant loss of accuracy, reducing the coefficient sizes. To show this, we start by converting Δ to Fourier space, i.e., diagonalizing via the quantum Fourier transform (QFT). The result may be computed by diagonalizing U_+ and is found to be

$$\begin{aligned} \Delta &= F \sum_{j=0}^{N_c-1} \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j|\,F^\dagger \\ &= F \sum_{j=-N_c/2}^{N_c/2-1} \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j|\,F^\dagger, \end{aligned} \tag{71}$$

where, in the second line, we define indices $-j = N_c - j$ for $j > 0$ and write the diagonalized Δ symmetrically about $j = 0$. The benefit of this parametrization is that small $|j|$ correspond to low-frequency modes. Let Δ_J be

Δ truncated at frequencies above those of the index $J \in [0, N_c/2] \cap \mathbb{Z}$:

$$\Delta_J := F \sum_{j=-J}^{J} \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| F^\dagger. \tag{72}$$

The error in a clock-space evolution using Δ_J rather than Δ is upper bounded by $T\|\Delta|\phi_0\rangle - \Delta_J|\phi_0\rangle\|$, which can be evaluated and upper bounded as

$$\begin{aligned} &T\|\Delta|\phi_0\rangle - \Delta_J|\phi_0\rangle\| \\ &\quad = \left\| \sum_{|j|>J} N_c \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| F^\dagger |\phi_0\rangle \right\| \\ &\quad \le N_c \sqrt{\sum_{|j|>J} |\langle j| F^\dagger |\phi_0\rangle|^2}. \end{aligned} \tag{73}$$

We thus desire a characterization of $F^\dagger|\phi_0\rangle$, which we naturally expect to be another Gaussian up to errors arising from the difference between discrete and continuous Fourier transforms. This analysis has been performed in Appendix C of Ref. [45] and we adapt that work to our present situation. As the reference shows, the error in each component j arises from three sources:

(1) Truncation of the time variable to $O(T)$, which we denote by ϵ_{trunc}.
(2) Truncation of the frequency variable to $O(N_c/T)$ ("aliasing"), which we denote by ϵ_{alias}.
(3) Differences in normalizing in the continuum versus the discrete setting, which we denote by ϵ_{norm}.

In our notation and setting, Rendon *et al.* [45] show that these errors satisfy the following asymptotic bounds:

$$\begin{aligned} \epsilon_{\text{trunc}} &\in O\left(\sqrt{\frac{\sigma}{T}} e^{-\Omega(T^2/\sigma^2)}\right), \\ \epsilon_{\text{alias}} &\in O\left(\sqrt{\frac{\sigma}{T}} e^{-\Omega(N_c^2\sigma^2/T^2)}\right), \\ \epsilon_{\text{norm}} &\in O\left(e^{-\Omega(N_c)}\right). \end{aligned} \tag{74}$$

Let us take these errors to all be $O(\epsilon_F)$, with the required ϵ_{QFT} to be determined. The results from Theorem 16 and Appendix C of Ref. [45] imply that

$$F^\dagger|\phi_0\rangle = \sum_{j=-N_c/2}^{N_c/2-1} \left(\sqrt{\frac{\pi N_c}{\mathcal{N}}} \frac{\sigma}{T} e^{-(\pi j\sigma/T)^2} + O(\epsilon_{\text{QFT}})\right) |j\rangle. \tag{75}$$

With this in hand, we return to Eq. (73). First,

$$\begin{aligned} |\langle j| F^\dagger |\phi_0\rangle|^2 &= \frac{\pi N_c}{\mathcal{N}} \frac{\sigma^2}{T^2} e^{-2(\pi j\sigma/T)^2} \\ &\quad + O\left(\sqrt{\frac{\sigma}{T}} e^{-(\pi j\sigma/T)^2} \epsilon_{\text{QFT}}\right), \end{aligned} \tag{76}$$

where we assume that the error ϵ_{QFT} is smaller asymptotically than the amplitude itself, to be justified. Taking the sum over high frequencies,

$$\begin{aligned} &\sqrt{\sum_{|j|>J} |\langle j| F^\dagger |\phi_0\rangle|^2} \\ &\quad \in O\left(\sqrt{\frac{N_c}{\mathcal{N}}} \frac{\sigma}{T} e^{-\Omega(J^2\sigma^2/T^2)} + \sqrt{\frac{\sigma}{T}} e^{-\Omega(J^2\sigma^2/T^2)} \epsilon_{\text{QFT}}\right) \\ &\quad \subseteq O\left(\sqrt{\frac{\sigma}{T}} e^{-\Omega(J^2\sigma^2/T^2)} (1 + \epsilon_{\text{QFT}})\right). \end{aligned} \tag{77}$$

We next observe that $\epsilon_{\text{QFT}} \in O(1)$ by previous assumptions and can now be removed. From Eq. (73), we obtain the full simulation error by multiplying by N_c:

$$\epsilon_J \in O\left(N_c \sqrt{\frac{\sigma}{T}} e^{-\Omega(J^2\sigma^2/T^2)}\right). \tag{78}$$

In order for $\epsilon_J \in O(\epsilon)$, we want the cutoff J to satisfy

$$e^{-J^2\sigma^2/T^2} \in O\left(\sqrt{\frac{T}{\sigma}} \frac{\epsilon}{N_c}\right), \tag{79}$$

which can be satisfied provided that J scales as

$$J \in \Theta\left(\frac{T}{\sigma}(\log T/\sigma + \log N_c + \log 1/\epsilon)\right) \subseteq \tilde{\Theta}(T/\sigma). \tag{80}$$

Letting $\tilde{\Delta} := \Delta_J$ for this choice of J, we now switch to considering the simulation of $\tilde{\Delta}$. Let $\delta' > 0$ and let $\Gamma(\delta') := \lceil (N_c/T\delta') \sin(2\pi J/N_c) \rceil$. We have

$$\sum_{j=-J}^{J} \frac{N_c}{T} \sin\left(2\pi \frac{j}{N_c}\right) |j\rangle\langle j| = \delta' \sum_{k=1}^{\Gamma(\delta')} S_k^{(\Delta)}(\delta') + O(\delta'), \tag{81}$$

where

$$S_k^{(\Delta)}(\delta') := \sum_{j=-N_c/2}^{N_c/2-1} \text{sgn}(j)(-1)^{k[k>\chi_j]} |j\rangle\langle j|, \tag{82}$$

and

$$\chi_j := \begin{cases} (N_c/T\delta')\sin(2\pi j/N_c)], & |j| \le J, \\ 0, & \text{else.} \end{cases} \tag{83}$$

Defining the unitary $V_\ell(\delta') := F\,S_\ell^{(\Delta)}(\delta')F^\dagger$, we have obtained an LCU decomposition of Δ. The PREP circuit is, as with $C(H)$, only a column of Hadamards on

$$n_\Delta \in O\left(\log\left((N_c/T\delta')\sin(2\pi J/N_c)\right)\right) \subseteq \tilde{O}\left(\log\frac{1}{\sigma\delta'}\right) \tag{84}$$

auxiliary qubits. Meanwhile, the SEL circuit may be constructed as $F\,\mathrm{SEL}'\,F^\dagger$, where SEL′ is a select circuit using the $S_\ell^{(\Delta)}$ signature matrices that can, as before, be implemented with comparator circuits that compute the sine.

Combining with Eq. (69), we obtain an approximate LCU decomposition of the approximate clock Hamiltonian $\tilde{H}_c$:

$$\begin{aligned}\tilde{H}_c &= \delta\sum_{i=1}^{L}\sum_{k=1}^{\Lambda_i(\delta)} U_i\otimes S_{ik}(\delta) + \delta'\sum_{\ell=1}^{\Gamma(\delta')}\mathbb{1}\otimes V_\ell(\delta') \\ &= H_c + O(\epsilon/T + L\delta + \delta')\end{aligned} \tag{85}$$

To achieve an ϵ-accurate simulation, we will require $\delta \in O(\epsilon/LT)$ and $\delta' \in O(\epsilon/T)$. The 1-norm $\|c\|_1$ of all of the coefficients is given by

$$\begin{aligned}\|c\|_1 &= \delta\sum_{i=1}^{L}\Lambda_i(\delta) + \delta'\Gamma(\delta') \\ &\in O\left(\sum_{i=1}^{L}\max_j|\alpha_{ij}| + \frac{N_c}{T}\sin(2\pi J/N_c)\right) \\ &\subseteq O\left(\|\alpha\|_{\infty,1}^{\mathrm{rev}} + J/T\right) \\ &\subseteq \tilde{O}\left(\|\alpha\|_{\infty,1}^{\mathrm{rev}} + \sigma^{-1}\right) \\ &\subseteq \tilde{O}\left(\|\alpha\|_{\infty,1}^{\mathrm{rev}} + \frac{\max_t\|\dot{H}\|T}{\epsilon}\right),\end{aligned} \tag{86}$$

where $\|\alpha\|_{\infty,1} := \sum_{i=1}^{L}\max_t|\alpha_i(t)|$ and $\tilde{O}$ suppresses multiplicative logarithmic factors. Thus, the number of queries to SEL and PREP circuits in an LCU encoding scales as

$$\boxed{Q \in \tilde{O}\left(\|\alpha\|_{\infty,1}^{\mathrm{rev}}T + \frac{\max_t\|\dot{H}\|T^2}{\epsilon} + \frac{\log 1/\epsilon}{\log\log 1/\epsilon}\right).} \tag{87}$$

The number of auxiliary qubits needed for the clock register is

$$n_c = \log N_p + \log N_q \in O\left(\log\left(\max_t\|\dot{H}\|T^2\right) + \log 1/\epsilon\right), \tag{88}$$

while the number of auxiliary qubits needed for the LCU block encoding is given by

$$\begin{aligned}n_{\mathrm{LCU}} &= n_{C(H)} + n_\Delta \\ &\in O\left(\log\frac{\|\alpha\|_{\infty,1}^{\mathrm{rev}}}{\delta} + \log\frac{1}{\sigma\delta'}\right) \\ &\subseteq O\left(\log\frac{L\|\alpha\|_{\infty,1}^{\mathrm{rev}}T}{\epsilon} + \log\frac{\max_t\|\dot{H}\|T^2}{\epsilon^2}\right) \\ &\subseteq O\left(\log L + \log\left(\|\alpha\|_{\infty,1}^{\mathrm{rev}}T\right)\right. \\ &\quad \left. + \log\left(\max_t\|\dot{H}\|T^2\right) + \log 1/\epsilon\right)\end{aligned} \tag{89}$$

for a total number of auxiliary qubits $n \in O(n_{\mathrm{LCU}})$.

C. Discussion

In this section, we have provided an algorithm for time-dependent simulation by qubitization for instances when the Hamiltonian is input as a linear combination of unitaries. We provide a procedure for constructing an LCU-block encoding on the augmented clock space and use the error analysis of Sec. IV to provide a query complexity for the method.

The presence of the Trotter term in the complexity given in Eq. (87) is unfortunate because, if it were absent, the query complexity would match the lower bounds for simulation in T and ϵ. As a note of optimism, we believe that this term is not due to the method itself but is a fault of the analysis. Specifically, forcing our Hamiltonian to vary slowly over the N_p larger subdivisions should prove unnecessary. This has been done essentially to make the evolution consistent across the clock Gaussian state. In reality, the Hamiltonian should only need to vary smoothly over the smallest increment δt. We are currently investigating modifications to the clock scheme that would make this more apparent.

Besides an LCU encoding, other natural block encodings of H_c may be possible. For example, a very general input model for $H(t)$ is to take it as a d-sparse matrix with query access to the nonzero entries. This seems quite a promising avenue to take, because then $H_c = C(H) + \Delta$ is $d+2$ sparse and there is a natural way to query the entries of H_c. Hence, such a Hamiltonian should immediately be simulable by qubitization (or other quantum walk methods). The trouble is that the largest entry in absolute value $\|H_c\|_{\max}$ of H_c comes from Δ, which is of size

$N_c/2T$. This is too large to yield an effective simulation algorithm. Of course, there is something odd about the need to care for the operator norm $\|\Delta\|$, since the typical state being acted on is a Gaussian $|\phi_j\rangle$. Thinking of Δ in frequency space, modes of frequency $\Omega(\sigma^{-1})$ should not be relevant for Gaussian states of width $O(\sigma)$ on the clock register. This suggests that a high-frequency truncation of Δ—say, $\tilde{\Delta}$—would act in approximately the same way on the Gaussians while decreasing the norm. However, there is no guarantee that the modified operator, $\tilde{\Delta}$, is sparse in the basis of clock times. Perhaps considering a reduced clock Hamiltonian $\tilde{H}_{ij} = \langle\phi_i|H_c|\phi_j\rangle$, with all small elements set to zero, would have the sparseness conditions required, along with a subspace norm of $\|\Delta\|_\phi \in O(\sigma^{-1})$.

VI. TIME-DEPENDENT SIMULATION BY MULTIPRODUCT FORMULAS

As suggested in Sec. III D, MPFs have already been considered extensively in the Hamiltonian-simulation community [4,46,47]. However, one of the deficiencies of MPFs is that they have yet to be generalized, formally, for use in time-dependent Hamiltonian simulations. Because U generally has time ordering, the techniques used in Ref. [36] involving Baker-Campbell-Hausdorff- (BCH) type expansions do not carry over directly. An approach based instead on the Magnus expansion might be expected to work in its place but no subset of terms in the expansion represents the exact evolution separated from error terms. Without this generalization, MPFs cannot be applied to interaction-picture algorithms as well as simulations of physical systems that have intrinsic time dependence.

It is relatively easy to propose a generalization of MPFs that would be expected to work well in the time-dependent case, by Trotterizing the continuous clock Hamiltonian in Eq. (22). When this is done in k_j steps, this amounts to replacing the k_jth power in Theorem 1 with a sequence of k_j unitaries at each time slice. This heuristic argument motivates the following definition.

Definition 2 (Time-Dependent Multiproduct Formulas). For finite-dimensional $\mathcal{H}$ and $L : [0,T]^2 \to L(\mathcal{H})$, let $L_p : [0,T]^2 \to L(\mathcal{H})$ be a pth-order formula for L. Given $m \in \mathbb{Z}_+$, $\vec{k} \in \mathbb{Z}_+^m$ and $a \in \mathbb{R}^m$, define the time-dependent multiproduct formula $L_{m,p} : [0,T]^2 \to L(\mathcal{H})$ to be

$$L_{p,m}(t,t_0) := \sum_{j=1}^{m} a_j L_p^{(k_j)}(t,t_0),$$

where

$$L_p^{(k)}(t,t_0) := \prod_{\ell=0}^{k-1} L_p(t_{\ell+1},t_\ell)$$

and $t_\ell = t_0 + (t-t_0)\ell/k$.

As a limiting case, observe that $L_{p,1} = L_p$ with $a_1 = 1$. The choice to take the t_ℓ as equally spaced is not entirely coincidental, for the same reason that, in the time-independent setting, we take $U_2^k(t/k)$ instead of, say,

$$\prod_{j=1}^{k} U_2(s_j t), \tag{90}$$

where $s = (s_1,\ldots,s_k)$ is a probability vector. Taking a simple power of k makes working with the BCH expansion much simpler. While these definitions could be applied in more general contexts, our interest in Hamiltonian simulation means that we will consider $L = U$ to be a time-evolution operator.

We finally turn to the question of whether the time dependent MPFs of Definition 2 may be constructed for improved approximants. At the beginning of this section, we mentioned the difficulty presented by time ordering in adopting the techniques from Ref. [36]. The reader of the previous section may recognize that clock spaces may be used to remove time ordering, circumventing the issue. However, when the clock variable t is continuous, the shift term $-E$ in the clock Hamiltonian is an unbounded operator, complicating a BCH-type analysis. We conjecture, and provide a heuristic argument, that time-dependent MPFs indeed boost the approximation order for sufficiently smooth Hamiltonians.

Conjecture 1. Let $H = \sum_{i=1}^{L} H_i(t)$ and let

$$U_2(t+\tau,t) = \prod_{i=L}^{1} e^{-iH_i(t+\tau/2)\tau} \prod_{i=1}^{L} e^{-iH_i(t+\tau/2)\tau}$$

be the symmetric second-order Trotterized midpoint formula. Suppose that each H_i is $2m+1$ time differentiable. Then, the time-dependent multiproduct formula $U_{2,m}(t+\tau,t)$ with base formula U_2 approximates $U(t+\tau,t)$ to order $2m$ in t.

We now discuss a potential path to proof of this conjecture. Without loss of generality, we take $t = 0$. Let $k \in \mathbb{Z}_+$ and consider a sequence of discrete clock constructions on interval $[0,\tau]$, with parameters $(N_p(\ell), N_q(\ell), \sigma(\ell))$, such that k always divides $N_c = N_p N_q$, and such that the limit reproduces the dynamics of $H(t)$ on the main register, as per Theorem 3. Consider one of the elements of this sequence. Using the form of H given in the conjecture statement, we may write

$$C(H) = \sum_{i=1}^{L} C(H_i). \tag{91}$$

Thus, the clock Hamiltonian H_c admits the following second-order symmetric Trotter:

$$V_2(\tau) = e^{-i\Delta\tau/2}\left(\prod_{i=L}^{1} e^{-iC(H_i)\tau/2}\prod_{i=1}^{L} e^{-iC(H_i)\tau/2}\right)e^{-i\Delta\tau/2}. \tag{92}$$

From Ref. [36], we have that

$$V(\tau) - V_2^k(\tau/k) = \sum_{j=1}^{m-1}\mathcal{E}_{2j+1}(\tau)\frac{\tau^{2j+1}}{k^{2j}} + \mathcal{E}(\tau,k), \tag{93}$$

where $\mathcal{E} \in O(\tau^{2m+1})$ is analytic in τ. Thus the standard well-conditioned multiproduct formula $V_{2,m}$ of Theorem 1 with base formula V_2 satisfies

$$V(\tau) - V_{2,m}(\tau) = \sum_{j=1}^{m} a_j\,\mathcal{E}(\tau,k_j). \tag{94}$$

We now wish to look at the action on the main register. Applying Eq. (94) to the state $|\psi\rangle|\phi_0\rangle$ of the full register, where $|\psi\rangle$ is arbitrary, and then taking the trace Tr_c over the clock register, one obtains

$$\begin{aligned}&\mathrm{Tr}_c(V(\tau)|\psi\rangle|\phi_0\rangle) - \mathrm{Tr}_c(V_{2,m}(\tau)|\psi\rangle|\phi_0\rangle)\\ &\quad= \sum_{j=1}^{m} a_j E(\tau,k_j)(|\psi\rangle),\end{aligned} \tag{95}$$

where $E(\tau,k)$ is a linear map on the main register, defined by

$$E(\tau,k)(|\psi\rangle) := \mathrm{Tr}_c(\mathcal{E}(\tau,k)|\psi\rangle|\phi_0\rangle). \tag{96}$$

The above holds for every clock space in the sequence defined by $(N_p(\ell), N_q(\ell), \sigma(\ell))$. Taking the limit as $\ell \to \infty$ of Eq. (95), we may pass the limits through the finite sums and scalar multiplications

$$\begin{aligned}&\lim_{\ell\to\infty}\mathrm{Tr}_c(V(\tau)|\psi\rangle|\phi_0\rangle) - \lim_{\ell\to\infty}\mathrm{Tr}_c(V_{2,m}(\tau)|\psi\rangle|\phi_0\rangle)\\ &\quad= \sum_{j=1}^{m} a_j \lim_{\ell\to\infty} E(\tau,k_j)|\psi\rangle,\end{aligned} \tag{97}$$

provided that these limits exist. Indeed, by Theorem 3,

$$\mathrm{Tr}_c(V(\tau)|\psi\rangle|\phi_0\rangle) = U(\tau,0)|\psi\rangle. \tag{98}$$

As for the MPF, taking k steps of the Trotterization, we *should* find that

$$\lim_{\ell\to\infty}\mathrm{Tr}_c(V_2(\tau/k)^k\psi|\phi_0\rangle_c) = U_2^{(k)}(\tau,0)|\psi\rangle \tag{99}$$

though this must be shown. This should not be too hard, as the idea is clear: perform a sequence of clock shifts followed by second-order Trotter on the main register. By passing the limit through the multiproduct sum,

$$\lim_{\ell\to\infty}\mathrm{Tr}_c(V_{2,m}(\tau,0)|\psi\rangle|\phi_0\rangle_c) = U_{2,m}(\tau,0)|\psi\rangle. \tag{100}$$

It remains to show that the limit $\lim_\ell E(\tau,k)$ exists and, moreover, is in $O(\tau^{2m+1})$. This is where the main challenge lies. To show that the limit of a sequence with terms of order $O(\tau^{2m+1})$ is also $O(\tau^{2m+1})$, we can show that the $2m+1$ derivative is bounded at $\tau = 0$. Unfortunately, our current clock constructions have the width σ of the clock state shrinking to infinity, which means that the derivatives grow as well. If a different clock construction can be provided where the clock state can have width $\sigma \in O(1)$, a bound can be placed and thus the limit will be $O(\tau^{2m+1})$.

Ongoing work is being undertaken to fill in the gaps of the previous argument. However, the numerics of Sec. VI E strongly suggest that the time-dependent MPFs indeed work as expected. Moreover, the form of the time-dependent MPF of Definition 2 can be obtained by a naive Trotterization of the continuous clock space, which is very suggestive that, beyond formal issues, the approach is reasonable. Thus, we proceed assuming that Conjecture 1 is true.

A. Statement of protocol

Having argued, informally, that good time-dependent MPFs exist, we now propose an algorithm for Hamiltonian simulation using these formulas. We will provide some accompanying discussion to explain our choices and at the end we will more directly state the approach.

A natural input model for $H(t)$ is a linear combination of Hamiltonians

$$H(t) = \sum_{i=1}^{L}\alpha_i(t)H_i, \tag{101}$$

where each $\alpha_i(t) \in \mathbb{R}$ is assumed to be $2m+1$ differentiable for an m-term MPF. Without loss of generality, we take $\|H_i\| \le 1$. Because we utilize the well-conditioning results of Ref. [35], we want the base formula to be second order and symmetric. A reasonable choice is

$$\begin{aligned}U_2(t+\tau,t) := &\prod_{i=L}^{1}\exp\left\{-iH_i\alpha_i\left(t+\frac{\tau}{2}\right)\tau\right\}\\ &\times\prod_{i=1}^{L}\exp\left\{-iH_i\alpha_i\left(t+\frac{\tau}{2}\right)\tau\right\},\end{aligned} \tag{102}$$

which is a second-order Trotter splitting of the midpoint formula. Thus, from now on, we will be interested in

the MPF

$$U_{2,m}(t,0) = \sum_{j=1}^{m} a_j U_2^{(k_j)}(t,0). \tag{103}$$

As a caution, we remark that, despite notation, the MPF $U_{2,m}$ is not generally unitary for $m > 1$, though when suitably constructed, it will approximate the unitary U and hence be approximately unitary.

That U_2 is second order can be seen from Taylor expanding the Dyson series of U about $\tau = 0$ (H needs to be at least, say, twice differentiable). Moreover, U_2 is time-reversal symmetric in the same sense as U: $U_2(t,t_0) = U_2(t_0,t)^\dagger$. This gives the nice property that the error series for $U(t+\tau,t) - U^{(k)}(t+\tau,t)$ has only even terms, such that higher-order formulas can be reached with approximately half the number of addends.

From the onset, there are a couple of choices to make. The MPFs, in principle, could approximate the entire interval $[0,T]$ provided that the Trotter steps k_i are sufficiently large. However, this has several disadvantages. First, there is no flexibility to treat some subintervals of $[0,T]$ as more difficult than others and allocate resources appropriately. Second, the well-conditioned scheme of Ref. [35] would have to be abandoned or modified to accommodate larger $\vec{k}$. Instead, we divide $[0,T]$ into a mesh of r subintervals, not necessarily uniform but, rather, constructed to account for more difficult parts of the simulation. We provide a greedy algorithm for constructing such a mesh at the end of this section. The algorithm requires a computable Λ_{2m+1}-bound to work (see Definition 3); however, a practitioner might prefer a more heuristic approach to constructing the time mesh. For the moment, we will simply say that, given t_i, the next time point t_{i+1} is incremented roughly as $1/\Lambda_{2m+1}(t)$ for t in a neighborhood of t_i, where Λ_{2m+1} is a positive real-valued function of H and its derivatives that grows for larger or faster fluctuating H.

Once the mesh points $t_0, t_1, \ldots, t_r$ are determined, a time-dependent MPF is performed over each subinterval $[t_i, t_{i+1}]$ in sequence. We assume that the MPF is implemented using the LCU technique. The base-midpoint formula U_2 must be implemented by some scheme that depends on the structure of $H(t)$, though the approximating unitary $\tilde{U}_2$ should be at least second order and should preserve the time-reversal symmetry of U_2 (and U). We take Eq. (102) as our base formula for the subsequent analysis. It is known that such Trotter formulas exhibit commutator scaling, meaning that, in the limit where all H_j commute pairwise and all α_j are constant functions, the simulation error goes to zero. Hence, the MPF will also inherit this desirable property.

Let us now supply our pseudoalgorithm for the MPF procedure. Given fundamental parameters, $[0,T]$, ϵ, and a description of $H(t)$:

(1) Compute a Λ_{2m+1}-bound (Definition 3) for some m larger than the expected number of MPF terms. This is more or less a bound on the "difficulty" of $H(t)$ at various times.
(2) Construct a time mesh of r steps using the algorithm of Appendix F.
(3) Perform a sequence of time-dependent MPFs $U_{2,m}(t_j, t_{j-1})$ [see Eq. (103)] over each time slice $[t_{j-1}, t_j]$. These MPFs are performed using the well-conditioned MPF scheme of Ref. [35], involving a select-prepare encoding and oblivious amplitude amplification (OAA).

Figure 3 provides a high-level schematic of the quantum circuits involved. The PREP circuit acts on an initial state $|0\rangle$ of the auxiliary register as

$$\text{PREP}\,|0\rangle = \sum_j \sqrt{\frac{a_j}{\|a\|_1}}\,|j\rangle\,, \tag{104}$$

while the SEL circuit acts as

$$\text{SEL}\,|j\rangle\,|\psi\rangle = |j\rangle\,U_2^{(k_j)}(t_i,t_{i-1})\,|\psi\rangle\,. \tag{105}$$

Specific information about the parameter choices, such as m and r, is provided in the subsequent error analysis, though sometimes only in a big-O sense.

B. Error analysis

In this section, we analyze the errors arising between the exact unitary U and the MPF approximation $\tilde{U}$ given by

$$\tilde{U}(T,0) = \prod_{i=1}^{r} U_{2,m}(t_i,t_{i-1}). \tag{106}$$

This analysis will ignore hardware imperfections and decoherence, assume that U_2 is implemented perfectly, and assume exact coefficients a_j. In the query-complexity analysis of Sec. VI D, we will consider additional algorithmic errors arising from a more precise specification of the Hamiltonian input model.

We introduce a useful definition to quantify errors succinctly. It is well understood that MPFs, like regular product formulas, have smoothness requirements to ensure convergence. To quantify the errors and costs of MPFs, we provide a metric that captures the "size" of H and its derivatives at each point in time, in order to characterize the difficulty of simulation.

Definition 3. Let $H(t) = \sum_{i=1}^{L} \alpha_i(t) H_i$ be a time-dependent finite-dimensional Hamiltonian with H_i Hermitian and $\alpha_i(t) \in \mathbb{R}$ having $n \in \mathbb{N} \cup \{\infty\}$ continuous derivatives. For each i, define a $\Lambda_{i,n}$-bound as any continuous function $\Lambda_{i,n} : [0,T] \to \mathbb{R}_+$ satisfying the following

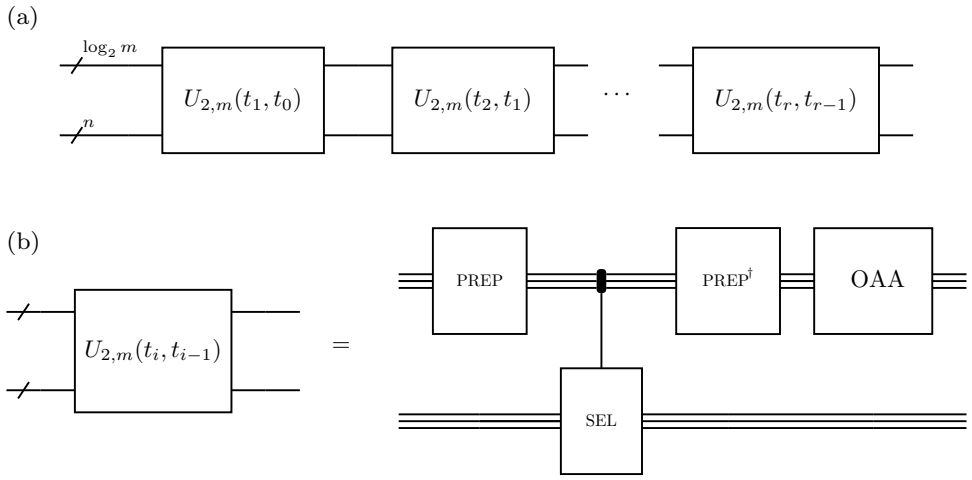

FIG. 3. A schematic of a quantum circuit implementing the time-dependent MPF algorithm. The lower register of n qubits represents the system of interest, while the upper auxiliary register consisting of $\log_2 m$ is for the LCU procedure. (a) The long-time MPF algorithm: multiple time steps. As with product formulas, the full interval $[0, T]$ is broken into k subintervals, chosen to devote more accuracy to more difficult parts of the simulation. (b) A schematic of a single MPF time step. Each time slice is implemented using the standard LCU approach. The "PREP" and "SEL" boxes represent the standard "prepare" and "select" circuits. Oblivious amplitude amplification (OAA) may be used to boost the success probability to one because the MPF is approximately unitary.

bounds with respect to H and its derivatives:

$$\Lambda_{i,n}(t) \geq \sup_{j\in[n]} \sqrt[j+1]{\|\alpha_i^{(j)}(t)\|} \quad \forall t \in [0, T],$$

where $f^{(n)}$ represents an nth derivative of f and $[n] := \{j \in \mathbb{N} \mid j \leq n\}$. Assuming that such bounds exist for all $i = 1, \ldots, L$, we say that $H(t)$ is Λ_n-bounded. We further say that $H(t)$ is Λ_n-boundable if it admits some Λ_n-bound. For convenience, we define $\Lambda_i := \Lambda_{i,\infty}$. We also define a Λ_n-bound as any continuous on $[0, T]$ satisfying

$$\Lambda_n(t) \geq \max_{i\in[L]} \Lambda_{i,n}(t).$$

For near-constant $\alpha_i(t)$, $\Lambda_{i,n}$ is simply an upper bound on $|\alpha_i|$, while for rapid oscillations, the derivative terms will dominate. Observe that for finite n, our assumptions imply that $\Lambda_{i,n}(t)$ exists (H is $\Lambda_{i,n}$-boundable), since $|\alpha_i^{(j)}|$ is continuous on a compact interval and hence is a bounded function. Also, in the finite case, the supremum may be replaced with a simple "max" and $\Lambda_{i,n}(t)$ may be taken as equal to the right-hand side because it is the maximum of a finite set of continuous functions, which is continuous. For this "minimal choice," $\Lambda_{i,n}(t)$ is a nondecreasing sequence in n. For each n, there also exists a $\Lambda_{i,n}$ that is constant in t. Allowing $\Lambda_{i,n}$ to vary in time, however, takes into consideration the possibility that the expense of simulating H will vary with time. We note that $\Lambda_{i,n}$-bounds are additive in the sense that, for $H(t)$ and $G(t)$ admitting $\Lambda_{i,n}^H$ and $\Lambda_{i,n}^G$-bounds, respectively, $\Lambda_{i,n}^H + \Lambda_{i,n}^G$ is a $\Lambda_{i,n}$-bound on $H + G$.

In contrast to finite n, the existence of a $\Lambda_{i,\infty}$-bound is not guaranteed and amounts to the assumption that the derivatives of H grow at most exponentially for asymptotically large j and fixed t. There are smooth even analytic functions that do not satisfy this, many of which are physically interesting. A simple example is a Gaussian pulse

$$\alpha(t) = e^{-t^2}, \tag{107}$$

the derivatives of which, generating the Hermite polynomials, grow factorially with n at $t = 0$. Other interesting cases, such as harmonic oscillations or exponential growth and decay, do admit a Λ-bound. Despite these restrictions, we adopt this approach for simplicity and in order to facilitate comparison with prior work on general-order Suzuki-Trotter formulas [29]. Admittedly, a modification of Definition 3 to be an upper bound on

$$\max_j j^{-1} \sqrt[j+1]{\|\alpha_i^{(j)}(t)\|} \tag{108}$$

would expand the class of functions admitting Λ_∞-bounds to analytic functions (though not generic smooth functions).

We now begin the error analysis of Eq. (106) in earnest. From a triangle inequality, the error can be bounded as the error in each step:

$$\|U(T, 0) - \tilde{U}(T, 0)\| \leq \sum_{i=1}^{r} \|U(t_i, t_{i-1}) - U_{2,m}(t_i, t_{i-1})\|. \tag{109}$$

Therefore, to ensure an error at most ϵ, it suffices that each subinterval has error at most ϵ/r. We thus focus on a single subinterval. An upper bound on this error is supplied by the following theorem, which is the main technical result of this section.

Theorem 4. Let $H : [t_0, t_1] \to \mathrm{Herm}(\mathcal{H})$ be a time-dependent Hamiltonian on finite-dimensional $\mathcal{H}$ with $2m+1$ continuous derivatives on $[t_0, t_1]$ and Λ_{2m+1}-bound. Suppose further that

$$eL \max_{\tau\in[t_0,t_1]} \Lambda_{2m+1}(\tau)(t_1 - t_0) < 1.$$

Then, for any $m \in \mathbb{Z}_+$, there exist $\vec{k} \in \mathbb{Z}_+^m$ and $a \in \mathbb{R}^m$ such that

$$\|U(t_1, t_0) - U_{2,m}(t_1, t_0)\| < \frac{\|a\|_1}{\sqrt{\pi m}} \left(5L \max_{\tau\in[t_0,t_1]} \Lambda_{2m+1}(\tau)(t_1 - t_0)\right)^{2m+1}$$

and $\|a\|_1 \in O(\log(m))$.

Observe that convergence of the above error bound to zero as $m \to \infty$ is conditioned on a sufficiently small $t_1 - t_0$. This is potentially unsurprising, as the Suzuki-Trotter formulas also do not provide an unconditionally converging sequence of approximations to the time-evolution operator. Note also the parallel roles between m and the Suzuki-Trotter order k in reducing the error. In our case, however, we shall see that the simulation cost increases only polynomially in m, whereas for product formulas the cost is necessarily exponential in k.

The term $\|a\|_1/\sqrt{\pi m}$ is $o(1)$ for large m and can be more or less ignored. Unfortunately, the Λ_{2m+1} scales as the "worst" coefficient α_i multiplied by the number of terms L, which seems too cynical. However, improving on this may greatly complicate the proof of the error bound. Theorem 4 will be the important result that informs the algorithmic choices and complexity analysis of subsequent sections. Having characterized the error on a single subinterval of $[0, T]$, the full error over r subintervals may be found simply by using Eq. (109).

We prove Theorem 4 using a similar strategy to that used to provide error estimates for the Suzuki-Trotter formulas [3,29,48]. As H is continuously differentiable at least $2m+1$ times, $U_{2,m}$ is a valid extrapolant by Conjecture 1 and cancels the first m terms in the error series. We can thus express the difference $U_{2,m} - U$ using the integral Taylor-remainder formulas

$$U_{2,m}(t, t_0) - U(t, t_0) = R_{2m} - \mathcal{R}_{2m}, \tag{110}$$

with

$$\mathcal{R}_{2m} := \frac{1}{2m!} \int_{t_0}^{t} (t-\tau)^{2m} U^{(2m+1)}(\tau, t_0) d\tau, \tag{111}$$

$$R_{2m} := \frac{1}{2m!} \int_{t_0}^{t} (t-\tau)^{2m} U_{2,m}^{(2m+1)}(\tau, t_0) d\tau, \tag{112}$$

where $U^{(n)}$ refers to derivatives in the first argument. By the triangle inequality,

$$\|U_{2,m}(t, t_0) - U(t, t_0)\| \le \|\mathcal{R}_{2m}\| + \|R_{2m}\| \tag{113}$$

and we upper bound each remainder in separate lemmas.

The easier bound is $\mathcal{R}_{2m}$, so we begin with the corresponding lemma.

Lemma 5. The remainder term $\mathcal{R}_{2m}$ in Eq. (112) satisfies

$$\|\mathcal{R}_{2m}\| < \frac{1}{2\sqrt{\pi m}} \left(2L \max_{\tau\in[t_0,t]} \Lambda_{2m+1}(\tau)(t - t_0)\right)^{2m+1}.$$

Proof. Recall that U, as the exact propagator, satisfies the Schrödinger equation [Eq. (8)]. Higher derivatives can easily be found through repeated application of the product rule. The result will be a polynomial in the derivatives of H times U itself. Under the spectral norm, using the triangle and submultiplicative properties, the ordering of terms does not matter and therefore is equivalent to the expression one gets taking derivatives of a scalar exponential. Noting that $\|U\| = 1$, the resulting polynomial is the complete exponential Bell polynomial from Faà di Bruno's formula (see Appendix D). Letting $n = 2m+1$, we have

$$\|\partial_t^n U(t, t_0)\| \le Y_n\left(\|H(t)\|, \|\dot{H}(t)\|, \ldots, \|H^{(n-1)}(t)\|\right). \tag{114}$$

From the definition of $\Lambda_{i,n}$, we have

$$\begin{aligned}\|H^{(j)}(t)\| &\le \sum_{i=1}^{L} |\alpha_i^{(j)}(t)| \\ &\le \sum_i \Lambda_{i,n}(t)^{j+1} \\ &\le (L\Lambda_n(t))^{j+1}\end{aligned} \tag{115}$$

and since the Bell polynomials Y_n are monotonic in each argument,

$$\begin{aligned}&Y_n\left(\|H\|, \|\dot{H}\|, \ldots, \|H^{(n-1)}\|\right) \\ &\quad\le Y_n(L\Lambda_n(t), (L\Lambda_n(t))^2, \ldots, (L\Lambda_n(t))^n) \\ &\quad= (L\Lambda_n(t))^n b_n,\end{aligned} \tag{116}$$

where b_n are the Bell numbers (Appendix D). Thus,

$$\|\partial_t^n U(t, t_0)\| \le (L\Lambda_n(t))^n b_n. \tag{117}$$

Finally, returning to the bound on $\mathcal{R}_{2m}$, we have from the integral triangle inequality that

$$\begin{aligned}\|\mathcal{R}_{2m}\| &\le \frac{1}{(2m)!}\int_{t_0}^{t}(t-\tau)^{2m}\|\partial_\tau^{2m+1}U(\tau,t_0)\|d\tau \\ &\le \frac{1}{(2m)!}\int_{t_0}^{t}(t-\tau)^{2m}(L\Lambda_{2m+1}(\tau))^{2m+1}b_{2m+1}d\tau,\end{aligned} \tag{118}$$

where we have made use of the inequality in Eq. (117). This, in turn, can be bounded by maximizing Λ_{2m+1} over $[t_0,t]$:

$$\begin{aligned}\|\mathcal{R}_{2m}\| &\le \frac{b_{2m+1}}{(2m)!}(L\max_{\tau\in[t_0,t]}\Lambda_{2m+1}(\tau))^{2m+1}\int_{t_0}^{t}d\tau(t-\tau)^{2m} \\ &\le \frac{b_{2m+1}}{(2m+1)!}\left(L\max_{\tau\in[t_0,t]}\Lambda_{2m+1}(\tau)(t-t_0)\right)^{2m+1}.\end{aligned} \tag{119}$$

Finally, we upper bound the prefactor using a Stirling bound and bounds from Ref. [49] on the Bell numbers. For all $m\in\mathbb{Z}_+$,

$$\begin{aligned}\frac{b_{2m+1}}{(2m+1)!} &< \frac{\left(\frac{0.792(2m+1)}{\log(2m+2)}\right)^{2m+1}}{\sqrt{2\pi(2m+1)}\left(\frac{2m+1}{e}\right)^{2m+1}} \\ &= \frac{1}{\sqrt{2\pi(2m+1)}}\left(\frac{.792e}{\log(2m+2)}\right)^{2m+1}.\end{aligned} \tag{120}$$

Plugging this into Eq. (119),

$$\begin{aligned}\|\mathcal{R}_{2m}\| &< \frac{1}{\sqrt{2\pi(2m+1)}} \\ &\quad\times\left(\frac{0.792e}{\log(2m+2)}L\max_{\tau\in[t_0,t]}\Lambda_{2m+1}(\tau)(t-t_0)\right)^{2m+1} \\ &< \frac{1}{2\sqrt{\pi m}}\left(2L\max_{\tau\in[t_0,t]}\Lambda_{2m+1}(\tau)(t-t_0)\right)^{2m+1}.\end{aligned} \tag{121}$$

The last line is the result of the lemma. ■

To prove a similar bound on R_{2m}, we will first need a technical lemma that bounds the size of ordinary exponentials of time-dependent matrices.

Lemma 6. Let $A(t)$ be an anti-Hermitian-valued function of $t\in\mathbb{R}$ with n bounded derivatives. Then,

$$\|\partial_t^n e^{A(t)}\| \le Y_n\left(\|\partial_t A(t)\|,\|\partial_t^2 A(t)\|,\ldots,\|\partial_t^n A(t)\|\right),$$

where Y_n is the complete exponential Bell polynomial.

In the scalar case, Faà di Bruno's bound is an exact expression (see Appendix D), so the content of our result is that a corresponding bound holds even in the nonscalar case. The exponential disappears because $e^{A(t)}$ is unitary. The proof of this is provided in Appendix E. We now state the bound on the Taylor R_{2m} for the time-dependent MPF.

Lemma 7. In the above notation, suppose that

$$eL\max_{\tau\in[t_0,t]}\Lambda_{2m+1}(\tau)(t-t_0)<1.$$

Then, the remainder term R_{2m} in Eq. (112) satisfies

$$\|R_{2m}\| < \frac{\|a\|_1}{2\sqrt{\pi m}}\left(5L\max_{\tau\in[t_0,t]}\Lambda_{2m+1}(\tau)(t-t_0)\right)^{2m+1}.$$

The proof is more technical than the previous bound of Lemma 5 and is relegated to Appendix E. We quickly prove Theorem 4 assuming the truth of the above lemma.

Proof of Theorem 4. First, we note that $\|a\|_1\ge 1$, since a necessarily satisfies $\sum_j a_j=1$ from the Vandermonde constraints in Eq. (24). From Eq. (113), the error $\|U(t,t_0)-U_{2,m}(t,t_0)\|$ is bounded by the sum of the remainder upper bounds derived in Lemmas 5 and 7. Comparing the two, we see that R_{2m} dominates $\mathcal{R}_{2m}$ for all $m\ge 1$. We thus take twice the larger as an upper bound:

$$\begin{aligned}\|U(t,t_0)-U_{2,m}(t,t_0)\| &< 2\|R_{2m}\| \\ &< \frac{\|a\|_1}{\sqrt{\pi m}}\left(5L\max_{\tau\in[t_0,t]}\Lambda_{2m+1}(\tau)(t-t_0)\right)^{2m+1}.\end{aligned} \tag{122}$$

This completes the proof. ■

C. Time-step analysis

The next ingredient we need for a complexity analysis is asymptotic bounds on the number of subintervals r needed in the time mesh. This will be the concern of this section. Unfortunately, in pursuing best-case bounds on r, we eschew a practical procedure for generating the time points t_i. Appendix F provides a concrete procedure that is based on the analysis of this section.

For time-dependent Hamiltonians, because the cost per unit time can vary with t in general, one should adaptively choose the step size depending on the cost. For our purposes, this means choosing a step size that is inversely proportional to the energy measure $\Lambda_{2m+1}(t)$. We will explore this adaptive time stepping and show L^1-norm scaling with $\Lambda_{2m+1}(t)$ here.

To derive bounds on r, we will need to assume something about the size of the derivative $\dot{\Lambda}_{2m+1}$ compared to Λ_{2m+1} itself. Given a Λ_n-bound, a differentiable (or even smooth) Λ_n-bound exists. From now on, we consider Λ_n-bounds for which there exists a $K \in \mathbb{R}_+$ such that $|\dot{\Lambda}_n(t)| \leq K\Lambda_n(t)^2$ for all $t \in [0, T]$. Given H that is Λ_n-boundable, there is always, in fact, a Λ_n-bound such that K exists and is arbitrarily close to zero. For example, we may take a constant bound $\Lambda_n' := \max_t \Lambda_n(t)$, noting that Λ_n is continuous on a compact interval. Of course, Λ' does not capture the changing behavior of $H(t)$ and is therefore suboptimal. Nevertheless, we have demonstrated that our additional assumptions are not much more restrictive than those we have already made. Note that (in natural units) K is dimensionless.

With these preliminaries in place, the following result provides an upper bound on the number of time steps needed for our MPF algorithm.

Lemma 8. Let H satisfy the assumptions of Theorem 4 and let Λ_{2m+1} be a Λ_{2m+1}-bound for H such that, for some $K \in \mathbb{R}_+$, $|\dot{\Lambda}_{2m+1}(t)| \leq K\Lambda_{2m+1}(t)^2$ for all $t \in [0, T]$. For every $\epsilon > 0$, there exists a list $(t_0, t_1, \ldots, t_r)$ of monotonically increasing times $t_j \in [0, T]$, with $t_0 = 0$ and $t_r = T$, such that

$$\|U(T,0) - \prod_{i=1}^{r} U_{2,m}(t_i, t_{i-1})\| \leq \epsilon,$$

with the number of time steps r bounded above as

$$r \leq \left\lfloor \left(5\left(1 + \frac{3}{2}K\right) L \|\Lambda_{2m+1}\|_1\right)^{1+\frac{1}{2m}} \left(\frac{\|a\|_1}{\epsilon\sqrt{\pi m}}\right)^{\frac{1}{2m}} \right\rfloor.$$

Here, $\|\Lambda_{2m+1}\|_1$ is the L^1 norm over $[0, T]$:

$$\|\Lambda_{2m+1}\|_1 := \int_0^T \Lambda_{2m+1}(t)dt.$$

Proof. As discussed in Sec. VI B, in order to satisfy the ϵ-error constraint of Lemma 8, it suffices that the error on each subinterval is less than ϵ/r. Using Theorem 4, the sum is bounded as

$$\sum_{i=1}^{r} \|U(t_i, t_{i-1}) - U_{2,m}(t_i, t_{i-1})\| \leq \frac{\|a\|_1}{\sqrt{\pi m}} \sum_{i=1}^{r} \left(5L \max_{\tau\in[t_{i-1},t_i]} \Lambda_{2m+1}(\tau)(t_i - t_{i-1})\right)^{2m+1}. \tag{123}$$

To ensure an overall error ϵ, it therefore suffices to produce a mesh such that, for each i,

$$\frac{\|a\|_1}{\sqrt{\pi m}} \left(5L \max_{\tau\in[t_{i-1},t_i]} \Lambda_{2m+1}(\tau)(t_i - t_{i-1})\right)^{2m+1} \leq \epsilon/r. \tag{124}$$

Rearranging, this corresponds to choosing t_i, given all other parameters, that satisfy

$$L \max_{\tau\in[t_{i-1},t_i]} \Lambda_{2m+1}(\tau)(t_i - t_{i-1}) \leq \frac{1}{5}\left(\frac{\epsilon\sqrt{\pi m}}{\|a\|_1 r}\right)^{1/(2m+1)}. \tag{125}$$

We now digress in order to relate $\max_\tau \Lambda_{2m+1}(\tau)$ and its average. Here is where we will make use of the K-bounds on the derivative $\dot{\Lambda}$: we closely follow arguments found in Ref. [29]. From the inequality in the lemma statement, we have

$$\begin{aligned} \left|\frac{\dot{\Lambda}_{2m+1}(t)}{\Lambda_{2m+1}(t)^2}\right| &\leq K, \\ \left|\frac{d}{dt}\frac{1}{\Lambda_{2m+1}(t)}\right| &\leq K. \end{aligned} \tag{126}$$

Suppose that the time t_{i-1} has been chosen by the previous iteration (if $i = 1$, $t_0 = 0$). Let $t > t_{i-1}$ and integrate the above inequality from t_{i-1} to t:

$$\begin{aligned} \int_{t_{i-1}}^{t} \left|\frac{d}{d\tau}\frac{1}{\Lambda_{2m+1}(\tau)}\right| d\tau &\leq K(t - t_{i-1}), \\ \left|\int_{t_{i-1}}^{t} \frac{d}{d\tau}\frac{1}{\Lambda_{2m+1}(\tau)} d\tau\right| &\leq K(t - t_{i-1}), \\ \left|\frac{1}{\Lambda_{2m+1}(t)} - \frac{1}{\Lambda_{2m+1}(t_{i-1})}\right| &\leq K(t - t_{i-1}). \end{aligned} \tag{127}$$

Let us rearrange this in terms of $\Lambda_{2m+1}(t)$ alone:

$$\begin{aligned} -K(t - t_{i-1}) \leq \frac{1}{\Lambda_{2m+1}(t)} - \frac{1}{\Lambda_{2m+1}(t_{i-1})} &\leq K(t - t_{i-1}), \\ \frac{1}{\Lambda_{2m+1}(t_{i-1})} - K(t - t_{i-1}) \leq \frac{1}{\Lambda_{2m+1}(t)} &\leq \frac{1}{\Lambda_{2m+1}(t_{i-1})} + K(t - t_{i-1}), \\ \frac{\Lambda_{2m+1}(t_{i-1})}{1 + K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})} \leq \Lambda_{2m+1}(t) &\leq \frac{\Lambda_{2m+1}(t_{i-1})}{1 - K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})}. \end{aligned} \tag{128}$$

The lower-bound inequality holds for all $t > t_{i-1}$, while the upper bound only holds when

$$(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})K < 1. \tag{129}$$

We restrict our attention to t for which both bounds hold. Consider, for the moment, only the leftmost inequality. The lower bound on the left is monotonically decreasing with t. This means that it is also a uniform lower bound on $\Lambda_{2m+1}(t')$ for any $t' \in [t_{i-1}, t]$. Therefore, it is a lower bound for the average $\bar{\Lambda}_{2m+1}(t)$ on the interval $[t_{i-1}, t]$:

$$\bar{\Lambda}_{2m+1}(t, t_{i-1}) := \frac{1}{t - t_{i-1}} \int_{t_{i-1}}^{t} \Lambda_{2m+1}(\tau) d\tau \tag{130}$$

That is,

$$\frac{\Lambda_{2m+1}(t_{i-1})}{1 + K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})} \leq \bar{\Lambda}_{2m+1}(t, t_{i-1}) \tag{131}$$

or, after isolating for $\Lambda_{2m+1}(t_{i-1})$,

$$\Lambda_{2m+1}(t_{i-1}) \leq \frac{\bar{\Lambda}_{2m+1}(t, t_{i-1})}{1 - K(t - t_{i-1})\bar{\Lambda}_{2m+1}(t, t_{i-1})}. \tag{132}$$

At this point, let us now consider the upper bound in Eq. (128). This bound is monotonically *increasing* in t and therefore also upper bounds $\Lambda_{2m+1}(\tau)$ for any τ in $[t_{i-1}, t]$. Therefore, it is also a bound for the maximum:

$$\max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau) \leq \frac{\Lambda_{2m+1}(t_{i-1})}{1 - K(t - t_{i-1})\Lambda_{2m+1}(t_{i-1})}. \tag{133}$$

Substituting bounds for $\Lambda_{2m+1}(t_{i-1})$ from Eq. (132) gives us a bound on the maximum value in terms of the average:

$$\max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau) \leq \frac{\bar{\Lambda}_{2m+1}(t, t_{i-1})}{1 - \frac{3}{2}K\bar{\Lambda}_{2m+1}(t, t_{i-1})(t - t_{i-1})}. \tag{134}$$

Solving for the average value of Λ_{2m+1} and multiplying by $t - t_{i-1}$ on both sides,

$$(t - t_{i-1})\bar{\Lambda}_{2m+1}(t, t_{i-1}) \geq \frac{(t - t_{i-1}) \max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau)}{1 + \frac{3}{2}K(t - t_{i-1}) \max_{\tau \in [t_{i-1}, t]} \Lambda_{2m+1}(\tau)}. \tag{135}$$

Let us finally choose a $t = t_i$ that will serve as the next time step in the adaptive scheme. We would like to come as close as possible to saturating Eq. (125) while staying within the constraint imposed by the maximum bound of Eq. (128). Thus, we choose t_i such that

$$\max_{\tau \in [t_{i-1}, t_i]} \Lambda_{2m+1}(\tau)(t_i - t_{i-1}) = \min\left\{\frac{1}{K}, \frac{1}{5L}\left(\frac{\epsilon\sqrt{\pi m}}{\|a\|_1 r}\right)^{1/(2m+1)}\right\}. \tag{136}$$

Since K is a constant, for asymptotic purposes we will assume sufficiently small ϵ such that the right-hand term is smaller. Plugging in to Eq. (135) yields

$$\bar{\Lambda}_{2m+1}(t_i, t_{i-1})(t_i - t_{i-1}) \geq \frac{\frac{1}{5L}\left(\frac{\epsilon\sqrt{\pi m}}{\|a\|_1 r}\right)^{1/(2m+1)}}{1 + \frac{3}{2}\frac{1}{5L}\left(\frac{\epsilon\sqrt{\pi m}}{\|a\|_1 r}\right)^{1/(2m+1)} K}. \tag{137}$$

We then find, by using the fact that $(1/5L)\left((\epsilon\sqrt{\pi m})/(\|a\|_1 r)\right)^{1/(2m+1)} < 1$ and by summing over $i = 1, \ldots, r$ in Eq. (137), that

$$\|\Lambda\|_1 \geq r^{\frac{2m}{2m+1}} \frac{1}{5L}\left(\frac{\epsilon\sqrt{\pi m}}{\|a\|_1}\right)^{1/(2m+1)} \left(1 + \frac{3}{2}K\right)^{-1}. \tag{138}$$

Finally, rearranging the above, this implies that the number of steps required for the MPF algorithm is upper bounded as

$$r \leq \left(5\left(1 + \frac{3}{2}K\right)L\|\Lambda\|_1\right)^{1+\frac{1}{2m}} \left(\frac{\|a\|_1}{\epsilon\sqrt{\pi m}}\right)^{\frac{1}{2m}}. \tag{139}$$

The result then directly follows from the requirement that r is an integer. ■

To summarize, we have provided an upper bound on the number of steps r needed given assumptions on the derivative of Λ_{2m+1}. What is perhaps objectionable is that, in determining our subsequent time stepping, we have seemed to have needed information about the total number of steps r that we would end up with. While this does not detract from the correctness of our result, it does indicate possible difficulty in constructing a suitable set of t_j for which the lemma holds. One approach is to guess the final number r_{try} of steps needed, construct the mesh according to the proof, then see if r_{try} can be made correct. This approach is considered in Appendix F.

D. Query complexity

With the results of Secs. VI B and VI C in hand, we proceed to bound the query complexity needed to perform a time-dependent MPF simulation. First, we define a set of

oracles that are appropriate for this simulation problem. As discussed above, the most natural input model in our setting is the linear combinations of Hamiltonians model,

$$H = \sum_{j=1}^{L} \alpha_j(t) H_j, \tag{140}$$

where $\alpha_j : [0,T] \to \mathbb{R}$ has $2m+1$ continuous derivatives and $H_j \in \mathrm{Herm}(\mathbb{C}^{2^n})$. Without loss of generality, we assume that $\|H_j\| \leq 1$. We discretize $[0,T]$ into 2^{n_c} uniform grid points $t_k = kT/2^{n_c}$ for $k \in [0, 2^{n_c}) \cap \mathbb{Z}$ and define $\alpha_{jk} := \alpha_j(t_k)$. Let $\delta t := T/2^{n_c}$. Let U_α and U_H be unitary oracles that provide the input Hamiltonian as follows:

$$\begin{aligned} U_\alpha |j\rangle |k\rangle |\tau\rangle |0\rangle &:= |j\rangle |k\rangle |\tau\rangle \left|\alpha_{jk}\tau\right\rangle, \\ U_H |j\rangle \left|\alpha_{jk}\tau\right\rangle |\psi\rangle &:= |j\rangle \left|\alpha_{jk}\tau\right\rangle \exp\{-iH_j\alpha_{jk}\tau\} |\psi\rangle. \end{aligned} \tag{141}$$

The oracle U_α encodes a reversible classical computation and may be taken as self-inverse. Here, $|\tau\rangle$ encodes a step of size $\tau \in \mathbb{R}$ in binary using n_c qubits. Such step sizes are always non-negative for the low-order formulas that we consider and therefore we take $\tau \in [0,T]$. Hence, $\delta t = T/2^{n_c}$ is the rounding error for the step sizes. We neglect rounding effects due to the values $\alpha_{jk}\tau$.

Our first result concerns the approximate implementation of U_2 using the two oracles.

Lemma 9. Let $U_2(\tau+t,t)$ be the second-order Suzuki-Trotter formula for the midpoint formula, with $t \in [0,T]$ and $\tau \in [0, T-t]$. Then, an approximation W_2 can be constructed using at most $6L-3$ queries to U_H and U_α, such that

$$\|U_2(t+\tau,t) - W_2(t+\tau,t)\| \leq L \max_{j,t} |\dot{\alpha}_j(t)| \tau T/2^{n_c}.$$

Proof. Define W_2 as U_2 but with each α_j evaluated at the nearest discrete times in $\{t_k\}$. Using the techniques of Ref. [29], two queries to U_α and one query to U_H are needed to exactly simulate each of the $2L-1$ exponentials present in W_2. Thus $3 \times (2L-1)$ queries are needed in total. To evaluate the discretization error, by Box 4.1 of Ref. [50] we have that

$$\begin{aligned} &\|W_2 - U_2\| \\ &\quad \leq 2 \sum_{j=1}^{L} \|e^{-iH_j\alpha_j(\mathrm{rnd}[t+\tau/2])\tau/2} - e^{-iH_j\alpha_j(t+\tau/2)\tau/2}\|, \end{aligned} \tag{142}$$

which in turn is upper bounded, through an application of the fundamental theorem of calculus, by

$$2 \sum_{j=1}^{L} \left\| H_j \alpha_j\left(\mathrm{rnd}\left[t+\frac{\tau}{2}\right]\right)\frac{\tau}{2} - H_j\alpha_j\left(t+\frac{\tau}{2}\right)\frac{\tau}{2} \right\|, \tag{143}$$

where rnd rounds to the nearest n_c-bit value. Since $\|H_j\| \leq 1$, this is merely upper bounded as

$$\tau \sum_{j=1}^{L} \left| \alpha_j\left(\mathrm{rnd}\left[t+\frac{\tau}{2}\right]\right) - \alpha_j\left(t+\frac{\tau}{2}\right) \right|. \tag{144}$$

By the fundamental theorem of calculus, with an integral upper bound, each term is upper bounded as $\delta t \max_{s \in t \pm \delta t} |\partial_t \alpha_j(s+\tau/2)|$. Maximizing over $[0,T]$ instead and making other simplifying choices, we obtain a crude upper bound

$$\begin{aligned} \|W_2 - U_2\| &\leq \tau L \delta t \max_{j,[0,T]} |\dot{\alpha}_j(t)| \\ &\leq L\tau T/2^{n_c} \max_{j,[0,T]} |\dot{\alpha}_j(t)|. \end{aligned} \tag{145}$$

Rearranging this gives the inequality of the lemma statement. ■

Having supplied an approximate base formula W_2 using our base oracles, we next need to implement an approximate MPF $W_{2,m}$ over a subinterval $[t_0,t_1]$. This is conventionally done through the use of "select" SEL and "prepare" PREP circuits:

$$\begin{aligned} \mathrm{PREP}\,|0\rangle &:= \sum_{j=1}^{m} \sqrt{\frac{|a_j|}{\|a\|_1}} |j\rangle, \\ \mathrm{SEL}\,|j\rangle |\psi\rangle &:= \mathrm{sgn}(a_j) |j\rangle W_2^{(k_j)}(t_1,t_0) |\psi\rangle. \end{aligned} \tag{146}$$

The PREP circuit can be implemented without any queries to U_α or U_H, whereas SEL requires $O(L\|\vec{k}\|_\infty)$ queries. Following the well-conditioned MPF scheme of Ref. [35], we have that $k_j \leq 3m^2$. This implies that a query to SEL requires $O(Lm^2)$ queries to U_H and U_α.

We can use SEL and PREP for a standard LCU block encoding in order to construct a time-dependent MPF with base formula W_2.

Lemma 10. Under the assumptions of Theorem 4 and the above query model, for any $[t_0,t_1] \subseteq [0,T]$, the time-dependent MPF $W_{2,m}$ with base formula W_2 satisfies

$$\begin{aligned} &\|W_{2,m}(t_1,t_0) - U(t_1,t_0)\| \\ &\quad \in O\left(\|a\|_1 \left(\max_{t \in [t_0,t_1]} \Lambda_{2m+1}(t)(t_1-t_0) \right)^{2m+1} \right), \end{aligned}$$

provided that

$$n_c \geq \log\left(\frac{3\sqrt{\pi}m^{5/2}L\max_{t,j}|\partial_t\alpha_j(t)|(t_1-t_0)T}{\left(5L\max_{t\in[t_0,t_1]}\Lambda_{2m+1}(t)(t_1-t_0)\right)^{2m+1}}\right),$$

and can be constructed with a number of queries to U_H and U_α scaling as $\widetilde{O}(m^2L)$.

Proof. From Lemma 4 of Ref. [51], we have

$$\begin{aligned}(\langle 0|\otimes I)(\text{PREP}^\dagger)\text{SEL}(\text{PREP})(|0\rangle\otimes I)&=\frac{1}{\|a\|_1}\sum_{j=1}^{m}a_j W_2^{(k_j)}(t_1,t_0)\\&=W_{2,m}(t_1,t_0)/\|a\|_1.\end{aligned}\tag{147}$$

Let $\delta'>0$ be such that, for all j and $\ell\in\{1,\ldots,k_j\}$,

$$\left\|W_2\left(\Delta t\frac{\ell}{k_j}+t_0,\Delta t\frac{\ell-1}{k_j}+t_0\right)-U_2\left(\Delta t\frac{\ell}{k_j}+t_0,\Delta t\frac{\ell-1}{k_j}+t_0\right)\right\|\leq\delta',\tag{148}$$

where $\Delta t=t_1-t_0$. Then, by invoking Box 4.1 from Ref. [50],

$$\left\|U_2^{(k_j)}(t_1,t_0)-W_2^{(k_j)}(t_1,t_0)\right\|\leq k_j\delta',\tag{149}$$

which, since $k_j\leq 3m^2$, implies that

$$\|V_{2,m}(t_1,t_0)-W_{2,m}(t_1,t_0)\|\leq 3m^2\delta'\|a\|_1.\tag{150}$$

We supply δ' using Lemma 9, obtaining

$$3m^2\delta'\|a\|_1\leq 3m^2\|a\|_1 L\max_{j,t\in[0,T]}|\dot\alpha_j(t+\tau/2)|\Delta t\frac{T}{2^{n_c}},\tag{151}$$

which gives us a bound on the discretized MPF $W_{2,m}$ relative to the undiscretized $V_{2,m}$.

It then follows from the triangle inequality and Theorem 4 that

$$\begin{aligned}\|W_{2,m}(t_1,t_0)-U(t_1,t_0)\|&\leq\|U_{2,m}(t_1,t_0)-U(t_1,t_0)\|+\|W_{2,m}(t_1,t_0)-U_{2,m}(t_1,t_0)\|\\&\leq\frac{\|a\|_1}{\sqrt{\pi m}}\left(5L\max_{t\in[t_0,t_1]}\Lambda_{2m+1}(t)\Delta t\right)^{2m+1}+3m^2L\|a\|_1\max_{j,t}|\dot\alpha_j(t)|\Delta t\frac{T}{2^{n_c}}.\end{aligned}\tag{152}$$

Under the assumption that

$$n_c\geq\log\left(\frac{3\sqrt{\pi}m^{5/2}L\max_{j,t}|\dot\alpha_j(t)|\Delta tT}{\left(5L\max_{t\in[t_0,t_1]}\Lambda_{2m+1}(t)\Delta t\right)^{2m+1}}\right),\tag{153}$$

the second term is bounded by the first in Eq. (152), so we have an upper bound

$$\left\|\sum_{j=1}^{M}a_j\prod_{q=1}^{k_j}W_2\left(Tq/k_j,T(q-1)/k_j\right)-U(T,0)\right\|\leq\frac{2\|a\|_1}{\sqrt{\pi m}}\left(5L\max_{t\in[t_0,t_1]}\Lambda_{2m+1}(t)\Delta t\right)^{2m+1}.\tag{154}$$

Since $U(T,0)$ is unitary, we know that the MPF implemented by our algorithm is close to a unitary. This means that we satisfy the necessary conditions for robust OAA as given by Lemma 5 of Ref. [51]. This implies that, using $O(\|a\|_1)$ applications of the unitary $(\text{PREP}^\dagger)\text{SEL}(\text{PREP})$, we can implement an operator $W_{2,m}$ satisfying

$$\|W_{2,m}(t_1,t_0)-U(t_1,t_0)\|\in O\left(\|a\|_1\left(\max_{t\in[t_0,t_1]}\Lambda_{2m+1}(t)(t_1-t_0)\right)^{2m+1}\right).\tag{155}$$

The number of queries scales as

$$Q_{\text{step}}\in O(\|a\|_1m^2L)\subseteq\widetilde{O}(m^2L).\tag{156}$$

■

With the short-time simulation costs in place, we are now ready to state our main theorem, which bounds the number of queries needed to perform the full multiproduct simulation of a time-dependent Hamiltonian.

Theorem 5. Consider the above query setting, under the assumptions of Theorem 4 and Lemma 8 (Λ_{2m+1}-bounded H with K bound on $\dot{\Lambda}_{2m+1}$). Let Q_{tot} be the number of queries to oracles U_α and U_H. There exists an operator $W_{\text{tot}}(T,0)$ that simulates the Hamiltonian $\sum_{j=1}^{L}\alpha_j(t)H_j$ to precision

$$\|(\langle 0|\otimes\mathbb{1})W_{\text{tot}}(T,0)(|0\rangle\otimes\mathbb{1})-U(T,0)\|\leq\epsilon,$$

with a number of queries scaling as

$$Q_{\text{tot}}\in\widetilde{O}\left(L(1+K)\|\Lambda_{2m+1}\|_1\log^2(1/\epsilon)\right)$$

and a number of auxiliary qubits scaling as

$$\widetilde{O}\left(\log\left(L\log\left(\frac{(1+K)\|\Lambda_{2m+1}\|_1}{\epsilon}\right)\right)+\max\{0,\log\left(\frac{L(1+K)\|\Lambda_{2m+1}\|_1\max|\partial_t\alpha_j(t)|T^2}{\epsilon}\right)\}\right).$$

Proof. We construct our approximation W_{tot} as

$$W_{\text{tot}}(T,0):=\prod_{i=1}^{r}W_{2,m}(t_i,t_{i-1}),\tag{157}$$

where the grid points t_i are given by the adaptive scheme of Sec. VI C. From Lemma 8, we have that the number of segments needed to perform the simulation within error ϵ obeys

$$r\in\widetilde{O}\left(\frac{((1+K)\|\Lambda_{2m+1}\|_1)^{1+1/(2m)}}{\epsilon^{1/(2m)}}\right).\tag{158}$$

Therefore, using Lemma 10,

$$Q_{\text{tot}}\in\widetilde{O}(m^2Lr)\subseteq\widetilde{O}\left(\frac{m^2L((1+K)\|\Lambda_{2m+1}\|_1)^{1+1/(2m)}}{\epsilon^{1/(2m)}}\right).\tag{159}$$

The approximate value of the optimal m can be found by equating the exponentially shrinking component of the cost to the polynomially increasing value of m. We choose m to satisfy

$$m^2=\left(\frac{(1+K)\|\Lambda_{2m+1}\|_1}{\epsilon}\right)^{1/2m}.\tag{160}$$

Solving for m yields

$$m=\frac{\log\left((1+K)\|\Lambda_{2m+1}\|_1/\epsilon\right)}{4\text{LambertW}\left(\log\left((1+K)\|\Lambda_{2m+1}\|_1/\epsilon\right)/4\right)}\in\widetilde{O}\left(\log\left(\frac{(1+K)\|\Lambda_{2m+1}\|_1}{\epsilon}\right)\right).\tag{161}$$

This implies that the query complexity Q_{tot} is in

$$\widetilde{O}\left(L(1+K)\|\Lambda_{2m+1}\|_1\log^2(1/\epsilon)\right).\tag{162}$$

The number of auxiliary qubits needed in the construction is in $O(\log(m))$ to implement the MPF and in $\lceil\log L\rceil+n_c$ to implement the U_α oracle. From Lemma 10, and the requirement that $n_c\geq 0$, we have

$$n_c\geq\max\left\{0,\log\left(\frac{3\sqrt{\pi}m^{5/2}L\max_{j,t}|\dot{\alpha}_j(t)|\Delta tT}{\left(5L\max_{t\in[t_{i-1},t_i]}\Lambda_{2m+1}(t)\Delta t_i\right)^{2m+1}}\right)\right\}\tag{163}$$

for all subintervals Δt_i. Using Eq. (124) to simplify the denominator and taking $\Delta t\leq T$, it suffices to choose n_c such that

$$n_c\in O\left(\max\left\{0,\log\left(\frac{m^2r}{\epsilon\|a\|_1}L\max_{j,t}|\dot{\alpha}_j(t)|T^2\right)\right\}\right)\subseteq\widetilde{O}\left(\max\left\{0,\log\left(\frac{L(1+K)\|\Lambda_{2m+1}\|_1\max|\partial_t\alpha_j(t)|T^2}{\epsilon}\right)\right\}\right),\tag{164}$$

where we have used Eqs. (159) and (161). The total number of auxiliary qubits thus scales as

$$\widetilde{O}\left(\log\left(L\log\left(\frac{(1+K)\|\Lambda_{2m+1}\|_1}{\epsilon}\right)\right)+\max\left\{0,\log\left(\frac{L(1+K)\|\Lambda_{2m+1}\|_1\max|\partial_t\alpha_j(t)|T^2}{\epsilon}\right)\right\}\right), \tag{165}$$

as is claimed in the lemma. ■

This shows that the cost of quantum simulation using MPFs broadly conforms to the cost scalings that one would expect of previous methods. In particular, similar to the truncated Dyson-series simulation method [22,52], we obtain that the cost of simulating a time-dependent Hamiltonian scales near linearly with time T and polylogarithmically with $1/\epsilon$.

E. Numerical demonstrations

In the above sections, we have developed and characterized MPFs for time-dependent simulations by showing their existence and proving error bounds. However, these bounds are unlikely to be the final word on the performance of the algorithm. For example, we have already mentioned that, for time-independent H, the MPF of Definition 2 is exact in cases in which the Hamiltonian consists of only commuting terms. Yet this behavior is not captured in the bound of Theorem 4 because Λ_{2m+1} is at least as large as $\|H\|$. This discrepancy is unrelated to the fact that, in practice, the second-order formula U_2 can only be computed approximately.

To begin bridging the gap between the actual performance of the algorithm and our bounds, we investigate time-dependent MPFs empirically through two numerical examples. We compute $U_{2,m}$ for these systems on a classical computer (using matrix computations) and compare the result with the exact propagator (computed within machine ϵ). The vector $\vec{k}\in\mathbb{Z}_+^m$ that we will use comes from the bottom half of Table I of Ref. [35] and minimizes $\|\vec{k}\|$ for $\|a\|_1\le 2$. In general, deriving an analytic solution for the propagator given a time-dependent Hamiltonian is challenging or impossible. To bypass this problem, we will consider a time-independent Hamiltonian that is viewed from a "noninertial" frame, thereby rendering the dynamics time dependent in the new frame (for details, see Appendix G).

1. Example 1: Electron in magnetic field, rotating frame

As a very simple first demonstration, consider a spin-1/2 particle (say, an electron) in a homogeneous magnetic field B in a frame rotating with frequency ω. The Hamiltonian is

$$\begin{aligned}\tilde{H}(t) &= (\omega+B\cos\theta)Z/2\\ &\quad + B\sin\theta(\cos\omega tX/2+\sin\omega tY/2).\end{aligned} \tag{166}$$

Though it is not strictly necessary to run the algorithm, let us compute an appropriate $\Lambda(t)$ upper bound. The spectral norm of $\tilde{H}$ may be upper bounded as

$$\|\tilde{H}\|\le\frac{|\omega+B\cos\theta|}{2}+|B\sin\theta|, \tag{167}$$

while the derivatives $\tilde{H}^{(n)}(t)$ have the bound

$$\begin{aligned}\|\tilde{H}^{(n)}(t)\| &\le |B\sin\theta\omega^n|,\\ \sqrt[n+1]{\|\tilde{H}^{(n)}(t)\|} &\le \omega\left|\frac{B\sin\theta}{\omega}\right|^{1/n+1}.\end{aligned} \tag{168}$$

For ω not too much larger than B, we see then that $\Lambda(t)=\omega$ is an appropriate choice.

The first thing to check will be that the error has the appropriate power-law scaling. Namely, for M-term formulas, the error ϵ_c for small t should scale as $O(t^{2m+1})$ or better. We can check this by computing the "running power" $p(t,t')$:

$$p(t,t'):=\frac{\log\epsilon_t/\epsilon_{t'}}{\log t/t'}. \tag{169}$$

For different but small values of t and t', the value of p should approach the expected order of the error: $2m+1$. Indeed, this is precisely the behavior observed in Fig. 4. For sufficiently small simulation times, a power-law dependence on the simulation error is observed and the corresponding power is as anticipated. Additionally, we see that the error decreases by orders of magnitude with each additional term once the power-law regime is reached. Choosing $m>4$ in this example quickly leads to machine precision being the dominant error source.

Next, we vary the MPF order m for fixed simulation time t. Since $\Lambda=\omega$, our bounds predict an exponential decay in the error but only provided that $t<1/\omega$. Otherwise, the bounds grow exponentially and say nothing useful about performance. In Fig. 5, we fix t at several different times and plot the error dependence on the multiproduct order m. Past a certain threshold value for m (which increases

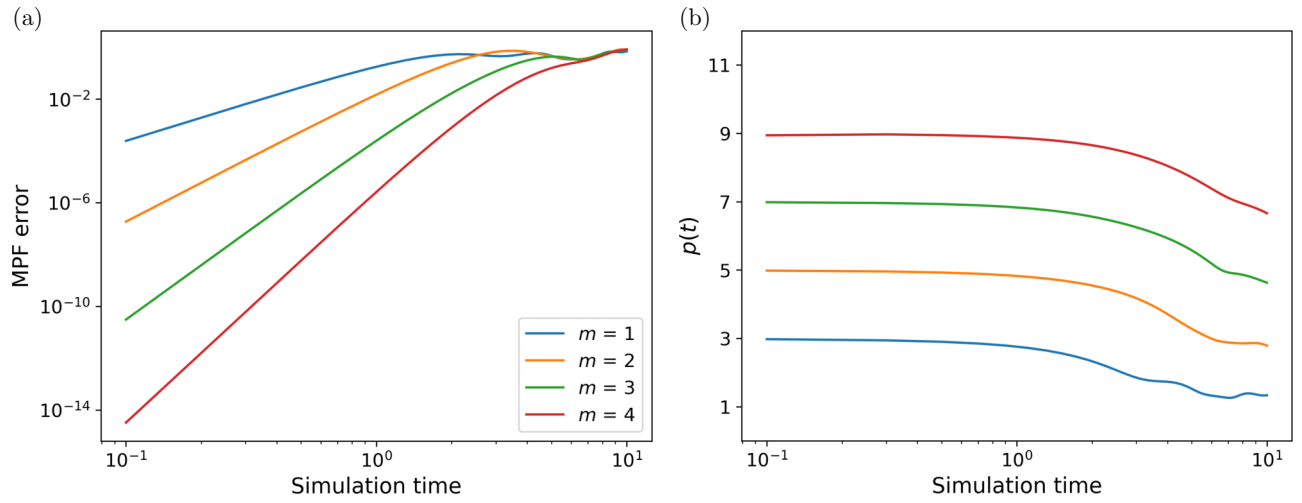

FIG. 4. (a) Multiproduct errors plotted against the simulation time, for MPF orders $m = 1, 2, 3, 4$, on a log-log plot. Note the power-law scaling for small values of t. The parameters used here are $B = 1$, $\omega = 4$, and $\theta = \pi/6$. For larger m, one quickly runs into machine precision becoming the dominant error source. (b) The running power $p(t, t')$ defined in Eq. (169), with $t' = .3$. Note that the plateau corresponds with the anticipated value of $2m + 1$.

with t), an exponential decay in error is observed, possibly superexponential. It is promising that, even for $t = 10$, the exponential decay is eventually achieved at $m \gtrsim 6$. This suggests that our error bounds may be too conservative and, in particular, that MPFs could *absolutely* converge to U as $m \to \infty$ in certain circumstances. This would be a notable improvement to product formulas alone, which tend to lead to errors that diverge as $m \to \infty$ if the time step t remains fixed [3,29,48]. In contrast, Theorem 4 shows that if the time step is sufficiently small, then the MPF converges to the exact result. However, such convergence is not anticipated from the bounds for a large value such as $t = 10$.

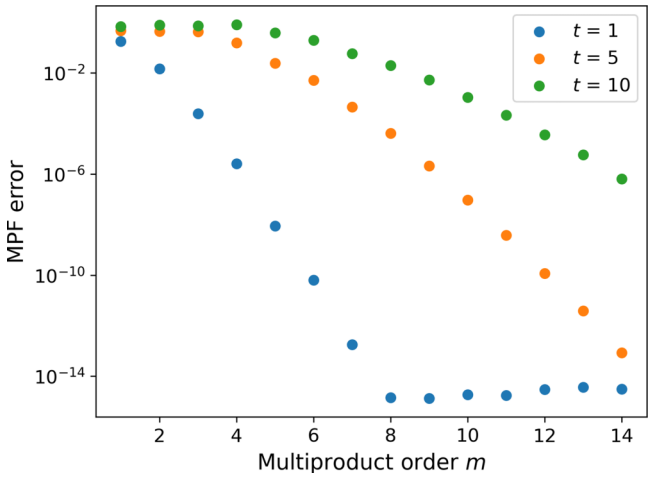

FIG. 5. The multiproduct error shows a (super)exponential decrease in error for sufficiently large order m. The threshold for this regime is seen to increase as the simulation time increases. This behavior surpasses the expectation of our proven bounds, since there are no guarantees if the time step is too large. Note that, in practice, one should typically split a longer simulation time into smaller steps. The plateau for $t = 1, m > 8$ is a result of machine-precision limitations. Parameter values: $B = 1$, $\omega = 4$, $\theta = \pi/6$.

Indeed, there are good reasons to believe that the absolute-convergence property holds more generically than this example. No matter how large the order m, we are still using a low-order formula (such as the midpoint formula U_2) as a base. Moreover, recall that the MPF is essentially a sum of product formulas with different numbers of time steps (for the same time interval). As the order m increases, higher weight is given to terms in the multiproduct sum with finer meshes. Correspondingly, terms that have larger time steps, and therefore may not converge properly, become suppressed at large m. Such behavior is not reflected in our derived error bounds, so there is likely room for improvement.

Practitioners in quantum simulation will likely want to know how MPFs fare against the more familiar and simpler Trotter techniques. To facilitate this, numerical studies across a broad range of physically interesting systems would be desirable. Such a comprehensive analysis must be left to future work; here, we will be satisfied with comparing MPFs with Trotterization for our spin-1/2 example. Our Trotterization is just an MPF with $m = 1$, corresponding to a midpoint-formula approximation. To facilitate as fair a comparison as possible, we will keep the number of midpoint-formula queries between the two methods the same. That is, we will enforce the requirement

$$r_{\text{trot}} = r_{\text{mpf}} \max_j |k_j|, \tag{170}$$

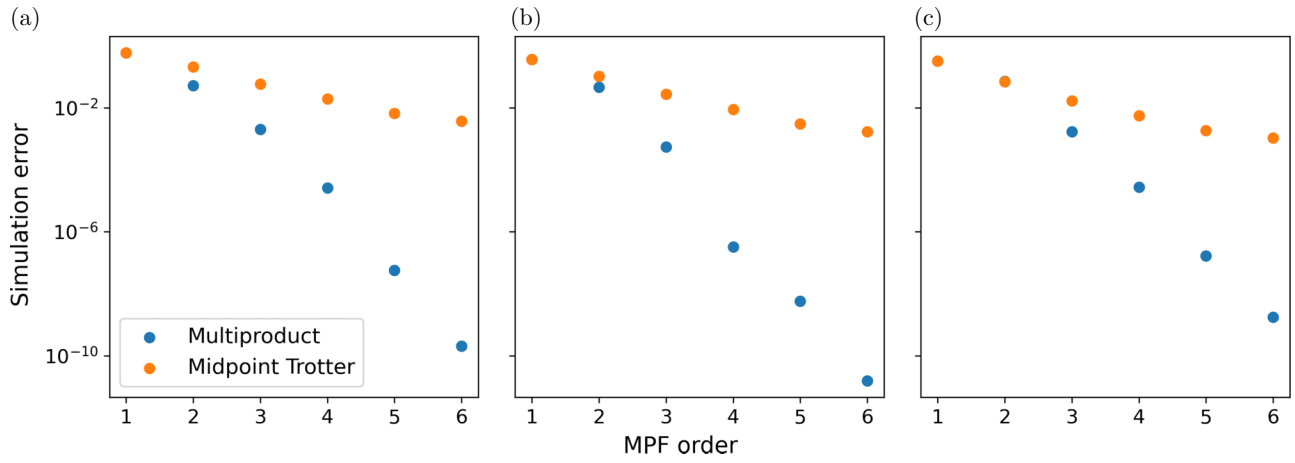

FIG. 6. The simulation error (spectral norm) of MPFs and midpoint-formula Trotterization, for the spin-1/2 system, with number of midpoint-formula queries kept fixed between the two: (a) $B = 10$, $\omega = 40$; (b) $B = 20$, $\omega = 20$; (c) $B = 40$, $\omega = 10$. Each plot corresponds to different values for the parameters B and ω, always with $\theta = \pi/6$. The number of MPF steps r_{mpf} is fixed at 10. The crossover point tends to occur for error $\epsilon > 10^{-3}$, which is large enough for practical significance. Such error tolerances can be orders of magnitude larger than those required in many quantum simulation proposals [10,12].

where r_{trot} and r_{mpf} are the number of time steps for Trotter and MPF, respectively. Note that the number of midpoint queries per time step for Trotter and MPFs are 1 and $\max_j |k_j|$, respectively.

In Fig. 6, we show the results of these comparisons for the several values of the magnetic field B and rotation frequency ω. The number of MPF steps r_{mpf} is fixed at 10, a reasonable value since it makes $\Lambda \Delta t \sim 1$ on each subinterval. As the MPF order increases, so does the number of Trotter steps r_{trot}, by the condition given in Eq. (170). These results show that, for m not too large, MPFs outperform Trotterization, at a value of the error ϵ that is large enough to be of practical significance for scientific or industrial applications. Admittedly, the spin-1/2 system considered above is rather simplistic. However, we anticipate most of the inferences drawn above to hold even as we increase the dimensionality of the Hilbert space. For example, though the complexity of simulating U_2 generally increases as $\dim(H)$ grows, it does so both for MPFs and Trotterization. Nevertheless, benchmarking MPFs on more complex systems would be desirable.

2. Example 2: Spin chain in interaction picture

As a first step toward more complicated many-body quantum systems, we investigate the use of MPFs for a particular one-dimensional (1D) chain of spins with nearest-neighbor interactions. As before, we will take advantage of a change of reference frame, allowing us to compare the multiproduct simulations with a machine-precise simulation in an equivalent time-independent frame. The Hamiltonian of interest is

$$\tilde{H}(t) = \frac{J}{2}\big(\cos(2\omega t) G_1 + \sin(2\omega t) G_2\big), \tag{171}$$

where

$$\begin{aligned} G_1 &= \sum_{k=1}^{N} X_k X_{k+1} + Y_k Y_{k+1}, \\ G_2 &= \sum_{k=1}^{N} (-1)^k (X_k Y_{k+1} - Y_k X_{k+1}). \end{aligned} \tag{172}$$

The observable of interest will be the magnetization $\mu := \sum_k Z_k$. One can verify that G_1 and G_2 do not commute with each other, yet they both commute with μ.

Assuming that $\tilde{H}$ commutes with an observable μ, to what degree does the MPF $U_{2,m}$ conserve μ? Since $U_{2,m}$ is an algebraic combination of exponentials of $\tilde{H}$, $U_{2,m}$ also commutes with μ. Thus, the operator μ evolves in the Heisenberg picture as

$$\mu_{2,m}(t) := U_{2,m}^{\dagger}(t) \mu U_{2,m}(t) = U_{2,m}^{\dagger}(t) U_{2,m} \mu. \tag{173}$$

However, $U_{2,m}$ is not necessarily unitary:

$$U_{2,m}^{\dagger}(t) U_{2,m}(t) \neq \mathbb{1}. \tag{174}$$

This implies that the conservation laws are only approximately conserved.

In Fig. 7, we plot the deviations in the conserved μ, $\|\mu - \mu_{2,m}(t)\|$, with respect to the simulation time. As the simulation time tends to zero, we see the expected power-law scaling, as evidenced by the linear relationship on a log-log plot. For larger m, the slope and hence the power p increases, corresponding to improved performance. We can extract the power as the slope of the line and this is plotted in the right-hand frame. Note that there are sudden

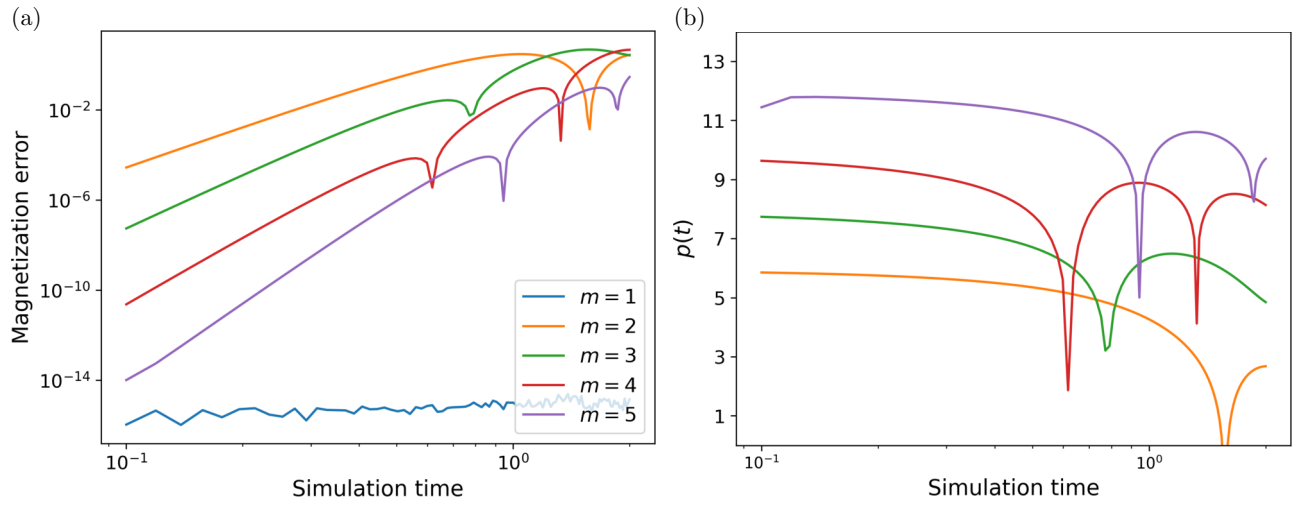

FIG. 7. (a) Deviations from the conservation of magnetization μ under time evolution by MPFs. Note that the order $m = 1$ is simply a product-formula evolution, which conserves μ exactly. For small simulation times, the expected power-law scaling is observed, with larger powers as m increases. (b) The running power $p(t, t')$ as defined in Eq. (169), with $t' = .3$. Note the plateau at $2m + 2$, which indicates slightly better convergence than naively expected ($p = 2m + 1$). This phenomenon generalizes to other systems and is formalized by Theorem 6. Parameter values: $N = 4, J = 1, \omega = 4$.

dips in the error at specific simulation times, which tend to occur before reaching the power-law-scaling regime. This could be due to cancellation between two terms of comparable magnitude in an error series. Similar phenomena occur in several other contexts, such as the error from adiabatic evolution [53]. Conclusive identification of these phenomena will require further study.

Naively, we would expect $p = 2m + 1$, but here we actually get slightly better: $p = 2m + 2$. In fact, this scaling can be justified.

Theorem 6. The deviation of $U_{2,m}$ from being unitary obeys

$$\|U_{2,m}^{\dagger}(t)U_{2,m}(t) - \mathbb{1}\| \in O(t^{2m+2}).$$

We provide a proof of this in Appendix G and a variant of this argument can be found in Ref. [34].

In summary, though MPFs do not inherently preserve commutation laws, the error is due to nonunitarity in $U_{2,m}$. This can be bounded and reduced in a systematic way, either by decreasing the time step or increasing the MPF order.

VII. CONCLUSIONS

The main contribution of this paper is a computational reduction, based on the (t, t') formalism, of time-dependent systems to time-independent ones, allowing for the replacement of ordered-operator exponentials with ordinary-operator exponentials acting on a higher- but finite-dimensional space. This augmented clock system may be directly simulated by quantum algorithms designed for time-independent Hamiltonians, thus extending their domain of applicability. In particular, we provide the first nontrivial application of qubitization to time-dependent Hamiltonians. Though our analysis does not show improvements over alternatives for time-dependent simulation, such as Trotter, we expect the fault to lie within the analysis rather than in the method itself. Simple numerics may elucidate whether this claim is plausible. Besides direct simulation, the clock framework provides a useful conceptual tool for developing algorithms. As a demonstration, we adapt the multiproduct formalism to the time-dependent case and in turn provide a simulation algorithm that not only has commutator scaling but also outperforms time-dependent Trotter-Suzuki methods. We support our theoretical findings with numerical demonstrations, which indicate the improved performance of time-dependent MPFs over low-order product formulas.

This work opens up a number of interesting possibilities. The most obvious, in our view, is to determine whether the clock formalism can lead to algorithms for time-dependent Hamiltonian simulation that match proven lower bounds through the use of qubitization. Another open question is whether these techniques could be used to translate commutator bounds for product formulas [5] over to the time-dependent case. This would be a significant step toward the development of a complete understanding of the error in Trotter-Suzuki formulas, since for the first time we would have a bound on the error of ordered-operator exponentials that yields the anticipated commutator scaling.

Regarding the MPF algorithm specifically, there is a possibility that MPFs converge to the propagator in the limit of large-order m regardless of the time-step size,

assuming sufficient smoothness in H. The corresponding statement is not true for Trotter-Suzuki: smaller and smaller time intervals must be taken to ensure convergence as one reaches higher-order formulas. Proving (or disproving) absolute convergence would be a valuable avenue for future research. On the numerical side, more convincing demonstrations of time-dependent MPFs, using larger systems, would be desirable.

ACKNOWLEDGMENTS

We thank Jeffrey Schenker and Dominic Berry for helpful discussions. This work was supported in part by the U.S. Department of Energy (DOE), Office of Science, Office of Nuclear Physics, Inqubator for Quantum Simulation (IQuS) under DOE (NP) Award No. DE-SC0020970, as well as Awards No. DE-SC0021152, No. DE-SC0013365 and No. DE-SC0023658, by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1848739, and by NSF award PHY-2310620. N.W.'s work on this project is supported by the "Embedding Quantum Computing into Many-Body Frameworks for Strongly Correlated Molecular and Materials Systems" project, which is funded by the DOE Office of Science, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences.

APPENDIX A: DETAILS OF CONTINUOUS CLOCK SPACE

Having taken a developmental approach in the main text, let us provide a more concrete characterization of the continuous clock space. The Hilbert space is given by

$$\mathcal{H} = \mathcal{H}_s \otimes \mathcal{H}_c, \tag{A1}$$

where $\mathcal{H}_c \cong L^2(\mathcal{M})$ and $\mathcal{M}$ is the connected one-dimensional smooth manifold representing t, which contains the interval $[0, T]$. On $\mathcal{H}_c$, E acts as a generator of translations but is an unbounded operator. Nevertheless, the exponentials of E above are well defined through the spectral theorem and functional calculus for unbounded operators [54]. States $\psi \in \mathcal{H}$ can be expressed as a certain class of integrable functions on $\mathcal{M}$, the values $\psi(t)$ of which are states on $\mathcal{H}_s$. The inner product on $\mathcal{H}$ is the natural one,

$$\langle \phi | \psi \rangle := \int_{\mathcal{M}} \langle \phi(t) | \psi(t) \rangle_s \, dt, \tag{A2}$$

where $\langle \cdot | \cdot \rangle_s$ denotes the inner product on $\mathcal{H}_s$.

If $\mathcal{M}$ is not exactly $[0, T]$, then a time-dependent observable $A(t)$ acting on $\mathcal{H}_s$ will need to be defined on the entire clock space. Once done, $A(t)$ is promoted to a parameter-independent observable $\mathbf{A}$ on $\mathcal{H}$, acting on $\psi \in \mathcal{H}$ in a manner corresponding with the original space:

$$(\mathbf{A}\psi)(t) := A(t)\psi(t) \tag{A3}$$

We observe that $\mathbf{A}$ is *local* in $\mathcal{H}_c$ in the sense of acting via multiplication in t space. Let $\mathbf{H}$ be the promoted Hamiltonian operator and let $\mathbf{U}(\tau)$ be the unitary operator given by

$$\mathbf{U}(\tau) = e^{iE\tau} e^{-i(\mathbf{H}-E)\tau}. \tag{A4}$$

One can verify that $\mathbf{U}$ solves the following Schrödinger equation:

$$\begin{aligned} i\partial_\tau \mathbf{U}(\tau) &= \mathbf{H}(\tau)\mathbf{U}(\tau) \\ \mathbf{U}(0) &= \mathbb{1}. \end{aligned} \tag{A5}$$

Here,

$$\mathbf{H}(\tau) := e^{iE\tau} \mathbf{H} e^{-iE\tau} \tag{A6}$$

is a τ-dependent Hamiltonian corresponding to simple uniform translation along the clock space. For any state $\Psi_0 \in \mathcal{H}$, the function

$$\Psi(\tau) := \mathbf{U}(\tau)\Psi_0 \tag{A7}$$

solves the Schrödinger equation generated by $\mathbf{H}(\tau)$ but, more importantly, it encodes solutions to the dynamics under $H(t)$. Indeed, for any $t \in \mathcal{M}$, we have a state $\psi(\tau, t) \in \mathcal{H}_s$ defined by

$$\psi(\tau, t) := (\Psi(\tau))(t) = (\mathbf{U}(\tau)\Psi_0)(t), \tag{A8}$$

which solves the Schrödinger equation of interest:

$$\begin{aligned} i\partial_\tau \psi(\tau, t) &= i\partial_\tau (\mathbf{U}(s)\Psi_0)(t) \\ &= (\mathbf{H}(\tau)\Psi_0)(t) \\ &= H(\tau + t)\psi(0, t). \end{aligned} \tag{A9}$$

The interpretation is that each t constitutes an initial time for performing the simulation, so we have a family of solutions parametrized by t with initial state $\psi(0, t)$. The evolution parameter acts, as expected, as the total time elapsed in the simulation. Finally, we can obtain a collection of time-evolution operators $U(t + \tau, t)$ on $\mathcal{H}_s$ for each $t \in [0, T]$ as follows:

$$U(t + \tau, t)\psi_0 := [\mathbf{U}(\tau)\Psi_0](t), \tag{A10}$$

where $\Psi_0 \in \mathcal{H}$ is any state for which $\Psi_0(t) = \psi_0$. This operator is unitary and solves the operator Schrödinger equation [Eq. (8)].

To summarize, the operator $\mathbf{U}$ of Eq. (A4) encodes a one-parameter family of time-evolution operators for

the system of interest, parametrized by the initial time. Thus, we have shown how the propagator U generated by a time-dependent H can be cast as an ordinary-operator exponential on an augmented space. Interesting in its own right, this framework also allows for a natural unification of ideas regarding "Trotterization." This term is used to refer to both (a) the splitting up of an (ordinary-)operator exponential of $H = \sum_j H_j$ into exponentials of the various H_j or (b) the simulation of a time-dependent Hamiltonian by time-independent simulations over small time intervals. These can be seen as manifestations of the same phenomenon. To illustrate with a pertinent example, consider a symmetric Trotterization of the unitary $\mathbf{U}$:

$$\begin{aligned} U_2(t+\tau,t) &:= e^{iE\tau}\left(e^{-iE\tau/2}e^{-iH(t)\tau}e^{-iE\tau/2}\right) \\ &= e^{-iH(t+\tau/2)\tau}. \end{aligned} \tag{A11}$$

We have just derived the *midpoint formula* [28,29] from scratch. The Trotter product theorem says that

$$\lim_{k\to\infty} e^{iE\tau}\left(e^{-iE\tau/2k}e^{-iH(t)\tau/k}e^{-iE\tau/2k}\right)^k = U(t+\tau,t). \tag{A12}$$

Note that this holds even though E is unbounded [55]. Thus,

$$U_2^{(k)}(t+\tau,t) = e^{iE\tau}\left(e^{-iE\tau/2k}e^{-iH(t)\tau/k}e^{-iE\tau/2k}\right)^k \tag{A13}$$

constitutes a good approximation to U for sufficiently large $k \in \mathbb{Z}_+$. This opens up the possibility of a more unified approach to Hamiltonian-simulation algorithms that has not yet been properly considered.

APPENDIX B: CLOCK-SPACE TECHNICAL LEMMAS

Here, we provide the proofs of several technical lemmas that are listed in Sec. IV.

Proof of Lemma 1. We proceed in several steps, first by computing $[U_+, C(H)]$. We have

$$\begin{aligned} [U_+, C(H)] &= \sum_{j=0}^{N_c-1} H_j \otimes [U_+, |j\rangle\langle j|] \\ &= \sum_{j=0}^{N_c-1} H_j \otimes (|j+1\rangle\langle j| - |j\rangle\langle j-1|. \end{aligned} \tag{B1}$$

By splitting the sum and reindexing (all increments modulo N_c), we can move the difference to the H_j, giving

$$\begin{aligned} [U_+, C(H)] &= \sum_j (H_j - H_{j+1}) \otimes |j+1\rangle\langle j| \\ &= -U_+ \sum_j (H_{j+1} - H_j) \otimes |j\rangle\langle j|. \end{aligned} \tag{B2}$$

Next, we have that $[U_-, C(H)] = -[U_+, C(H)]^\dagger$. Thus,

$$[U_+ - U_-, C(H)] = -2\,\mathrm{Re}\left(U_+ \sum_j (H_{j+1} - H_j) \otimes |j\rangle\langle j|\right) \tag{B3}$$

and the full result follows almost immediately from the definition of Δ given in Eq. (29).

As for the upper bound, we note that $\|\mathrm{Re}(A)\| \le \|A\|$ for any finite-dimensional A and by unitary invariance of the spectral norm, we have

$$\begin{aligned} \|[\Delta, C(H)]\| &\le \left\| \sum_j \frac{H_{j+1} - H_j}{\delta t} \otimes |j\rangle\langle j| \right\| \\ &= \max_j \left\| \frac{H_{j+1} - H_j}{\delta t} \right\|. \end{aligned} \tag{B4}$$

The upper bound then follows from the claim

$$\left\| \frac{H_{j+1} - H_j}{\delta t} \right\| \le \max_{t\in[t_j, t_{j+1}]} \|\dot{H}(t)\|, \tag{B5}$$

coming from the fundamental theorem of calculus and the triangle inequality. ■

Proof of Lemma 2. By cyclicity, the normalization $\mathcal{N}$ is the same for all $|\phi_j\rangle$, so we consider $j = 0$. Because $|\phi_0\rangle$ is normalized in the Euclidean norm, we have

$$\begin{aligned} \mathcal{N} &= \sum_{j=0}^{N_c-1} e^{-2|\delta t^2 j|_c^2/\sigma^2} \\ &= \sum_{j=0}^{N_c/2-1} e^{-2j^2\delta t^2/\sigma^2} + \sum_{j=N_c/2}^{N_c-1} e^{-2(N_c-j)^2\delta t^2/\sigma^2} \\ &= 1 + \sum_{j=1}^{N_c/2-1} e^{-2j^2\delta t^2/\sigma^2} + \sum_{j=1}^{N_c/2} e^{-2j^2\delta t^2/\sigma^2} \\ &= \sum_{j=0}^{\frac{N_c}{2}-1} e^{-2j^2\delta t^2/\sigma^2} + \sum_{j=0}^{\frac{N_c}{2}} e^{-2j^2\delta t^2/\sigma^2} - 1. \end{aligned} \tag{B6}$$

We may lower bound the sums as Riemann approximations to a Gaussian integral, giving error functions "erf":

$$\begin{aligned} \mathcal{N} &\ge \sqrt{\frac{\pi}{8}}\left(\mathrm{erf}\left(\frac{T+2\delta t}{\sqrt{2}\sigma}\right) + \mathrm{erf}\left(\frac{T}{\sqrt{2}\sigma}\right)\right) - 1 \\ &> \sqrt{\frac{\pi}{2}}\frac{\sigma}{\delta t}\mathrm{erf}\left(\frac{T}{\sqrt{2}\sigma}\right) - 1, \end{aligned} \tag{B7}$$

which then implies

$$\frac{1}{\mathcal{N}} \le \sqrt{2/\pi}(\delta t/\sigma)\frac{1}{\operatorname{erf}\left(\frac{T}{\sqrt{2}\sigma}\right) - \sqrt{\frac{2}{\pi}}\frac{\delta t}{\sigma}}$$
$$= \sqrt{\frac{2}{\pi}}(\delta t/\sigma) + O\left((\delta t/\sigma)\left(\frac{\delta t}{\sigma} + e^{-\frac{T^2}{2\sigma^2}}\right)\right)$$
$$\in O(\delta t/\sigma). \tag{B8}$$

The result follows simply from taking a square root. ■

APPENDIX C: SIGNATURE-MATRIX DECOMPOSITION

Here, we provide an overview of the signature-matrix decomposition, also known as the alternating-sign trick, which has been used in our clock-space qubitization algorithm of Sec. V for achieving an LCU expression for diagonal (or easily diagonalized) linear operators. While this technique has been a part of the digital quantum simulation toolbox for some time, unfortunately the literature leaves no clear trace of it. Because of this, we hope the reader will find this overview helpful beyond our present application, by filling in a needed record.

Let H be a Hermitian operator on D dimensions, with diagonal decomposition $H = \sum_{j=1}^{D} \lambda_j |j\rangle\langle j|$. The question is how we can write this operator as a sum of unitaries, at least to some approximation. In looking for an appropriate set $\{U_j\}$, it makes sense to restrict our attention to those diagonals in the same basis as H. We may naturally restrict U_j to be Hermitian because H is as well. These stringent requirements force the U_j to be so-called signature matrices: diagonal matrices with nonzero entries ± 1.

To state the idea clearly, we focus on a single eigenvalue λ_j. We count up to λ_j by units of 1 until $\lceil \lambda_j \rceil$ is reached. Then, we alternate between adding units of -1 and $+1$. The last step may seem odd but is necessary because, with unitaries, we cannot simply add zero. Nor can we stop the adding procedure before *all* of the eigenvalues have been reached by additions of 1.

Let us now proceed more formally. Let $L = \lceil \|H\| \rceil$ be the first integer larger than the largest eigenvalue of H. Define a signature matrix U_k for each $k \in \{1, \ldots, L\}$ as follows:

$$U_k = \sum_{j=1}^{D} (-1)^{k[k>\lambda_j]} |j\rangle\langle j|. \tag{C1}$$

Here, $[P]$ is the Boolean function for proposition P assigning 1 to true, 0 to false. We see that, for k even, $U_k = \mathbb{1}$ is the identity operator, while for odd k, U_k has eigenvalue -1 whenever j is such that $k > \lambda_j$.

Let $G = \sum_{k=1}^{L} U_k$. Then G is also diagonal in the $|j\rangle$ basis and, moreover, the associated eigenvalue η_j is given by

$$\eta_j = \sum_{k=1}^{L} (-1)^{k[k>\lambda_j]} = \lfloor \lambda_j \rfloor + O(1), \tag{C2}$$

where $O(1)$ in fact denotes an integer from the set $\{-1, 0, 1\}$. Thus, the error between η_j and λ_j is upper bounded by 2.

This might not seem like a good approximation, especially when λ_j is small. But we can artificially increase the size of λ_j by performing the same procedure for H/δ for suitably small δ and then multiplying by δ. Let $L_\delta = \lceil \|H\|/\delta \rceil$. Then,

$$H/\delta = \sum_{k=1}^{L_\delta} U_k + O(1), \tag{C3}$$

so

$$H = \sum_{k=1}^{L_\delta} \delta U_k + O(\delta). \tag{C4}$$

We have succeeded at expressing H in LCU form to accuracy $O(\delta)$ using L_δ terms.

What about LCU computation? If H is defined on n qubits, we need H to be efficiently diagonalizable by a unitary circuit W into the computational basis. We then need to construct the PREP and SEL oracles. The PREP is simple enough: after normalization, we just need a uniform superposition. Meanwhile, SEL requires controlled U_k operations. Each U_k can be constructed with the help of a classical comparator circuit to compare each λ_j to the integer k. The number of auxiliary qubits we will need is $\lceil \log L_\delta \rceil \in O(\log \|H\|/\delta)$ to obtain accuracy δ. We will leave the discussion at that: suffice to say that because these constructions exist, our query complexities give an accurate reading on the gate-simulation complexity.

APPENDIX D: TOOLS FROM COMBINATORICS

This appendix is a reference for several tools from combinatorics used, especially in connection to the MPF error analysis of Sec. VI B.

The simple factorial $n!$ counts the number of permutations of n objects and is usefully approximated by *Stirling's approximation*. In the main text, we always make use of a version of the approximation that gives strict bounds for $n \in \mathbb{Z}_+$:

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n < n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{1/(12n)}. \tag{D1}$$

These bounds are extremely tight, even for small n.

The *multinomial coefficient* is a generalization of the more common binomial coefficient and it arises in several combinatorial situations. It is defined by

$$\binom{n}{n_1,\ldots,n_k} := \frac{n!}{n_1!n_2!\ldots n_k!}, \tag{D2}$$

where $n \in \mathbb{Z}_+$ and the $(n_\ell)_{\ell=1}^k$ are non-negative integers that sum to n. It is a positive integer corresponding to the number of distinct ways of placing n distinguishable items into k boxes, where each box has a fixed number n_ℓ of items. In this work, we will find occasion to make use of the multinomial when evaluating high-order derivatives of a product:

$$\left(\frac{d}{dt}\right)^n f_1(t)f_2(t)\ldots f_k(t). \tag{D3}$$

Here, $(f_\ell)_{\ell=1}^k$ are n-differentiable functions of $t \in \mathbb{R}$. Employing the product rule, one is left to count all the possible combinations of derivatives of each f_ℓ. It turns out that the multinomial is suited for this:

$$\left(\frac{d}{dt}\right)^n \prod_{\ell=1}^k f_\ell(t) = \sum_N \binom{n}{n_1,\ldots,n_k} \prod_{\ell=1}^k \left(\frac{d}{dt}\right)^{n_\ell} f_\ell(t). \tag{D4}$$

The sum is taken over the set N of sequences of non-negative integers $(n_\ell)_{\ell=1}^k$ summing to n. A useful property is that

$$\sum_N \binom{n}{n_1,\ldots,n_k} = k^n \tag{D5}$$

for non-negative integers k and n (with the convention $0^0 = \lim_{x\to 0} x^x = 1$).

Besides derivatives of products, we will also need to bound derivatives of ordinary exponentials of a time-dependent matrix. Useful for this purpose is an expression for derivatives of exponentials of a scalar function $a(t)$:

$$\left(\frac{d}{dt}\right)^n e^{a(t)}. \tag{D6}$$

The solution that we rely on is *Faà di Bruno's formula*, which asserts that

$$\left(\frac{d}{dt}\right)^n e^{a(t)} = e^{a(t)} Y_n(a'(t), a''(t), \ldots, a^{(n)}(t)), \tag{D7}$$

where Y_n is the *complete exponential Bell polynomial* [56]. An explicit formula is given by

$$Y_n(x_1, x_2, \ldots, x_n) = \sum_C \frac{n!}{c_1!c_2!\ldots c_n!} \prod_{j=1}^n \left(\frac{x_j}{j!}\right)^{c_j}, \tag{D8}$$

where the sum is taken over the set C of all sequences $(c_j)_{j=1}^n$ such that $c_j \geq 0$ and

$$c_1 + 2c_2 + \cdots + nc_n = n. \tag{D9}$$

Essentially, each coefficient in Y_n counts the ways in which one can partition a set of fixed size n into subsets of given sizes and number. When one simply wants to count the total number of possible partitions, one is led to the *Bell numbers* b_n. These are related to the Y_n by evaluating all arguments to 1:

$$b_n = Y_n(1, 1, \ldots 1) \tag{D10}$$

More generally, for any $x \in \mathbb{R}$,

$$Y_n(x, x^2, \ldots, x^n) = x^n b_n, \tag{D11}$$

which can be seen directly from Eq. (D8) along with the sum rule given in Eq. (D9). The Bell numbers b_n grow combinatorially; in particular, the following upper bound [49] is useful:

$$b_n < \left(\frac{.792n}{\log(n+1)}\right)^n, \quad \forall n \in \mathbb{Z}_+. \tag{D12}$$

More generally, the single-variable *Bell polynomial*, or *Touchard polynomial* $B_n(x)$, is simply Y_n with all arguments evaluated to x:

$$B_n(x) = Y_n(x, x, \ldots, x) \tag{D13}$$

Of course, $b_n = B_n(1)$. The nth Bell polynomial $B_n(x)$ is also the value of the nth moment of the Poisson distribution with mean x. From Ref. [57], we have the following upper bound on B_n:

$$B_n(x) \leq \left(\frac{n}{\log\left(1+\frac{n}{x}\right)}\right)^n, \quad \forall x \geq 0, \tag{D14}$$

which we observe is very close to that for the Bell numbers ($x = 1$) in Eq. (D12). From their definitions, Y_n, B_n, and b_n all grow monotonically, both in their functional arguments and their index n. This is intuitive from being combinatorial functions the coefficients of which count something according to the size of n.

APPENDIX E: PROOFS OF MPF TECHNICAL LEMMAS

In this appendix, we prove several technical lemmas used in the analysis of Sec. VI B.

Proof of Lemma 6. From the Trotter product theorem, we have

$$\partial_t^n \exp(A(t)) = \partial_t^n \lim_{r\to\infty} (\exp(A(t)/r))^r. \tag{E1}$$

Using the fact that the series converges uniformly, we may interchange the order of differentiation and the limit. This leads to

$$\|\partial_t^n \exp(A(t))\| \le \lim_{r\to\infty} \sum_S \binom{n}{s_1,\ldots,s_r} \prod_{q=1}^r \left\|\partial_t^{s_q} \exp(A(t)/r)\right\|. \tag{E2}$$

Here, the sum over S is constrained such that $s_j \ge 0$ and $s_1 + \cdots + s_r = n$. Then, using Taylor's theorem, we have

$$\left\|\partial_t^{s_q} \exp(A(t)/r)\right\| \le \frac{\|A^{(s_q)}(t)\|}{r} + O(1/r^2) \tag{E3}$$

for $s_q > 0$, where the $O(1/r^2)$ terms will vanish as $r \to \infty$. The $s_q = 0$ case has upper bound 1 by unitarity. Hence, put together,

$$\|\partial_t^n \exp(A(t))\| \le \lim_{r\to\infty} \sum_S \binom{n}{s_1,\ldots,s_r} \prod_{q=1}^r \left(\frac{\|A^{(s_q)}(t)\|(1-\delta_{s_q,0})}{r} + \delta_{s_q,0}\right). \tag{E4}$$

Now let us define a scalar function $a(x)$ defined for x in a neighborhood of t such that, for any k such that $0 \le k \le n$,

$$a^{(k)}(t) = \|A^{(k)}(t)\|(1-\delta_{k,0}) \tag{E5}$$

for a particular $x = t$. Such a function can be seen to exist by considering the nth-degree Taylor polynomial. We may apply the standard Faà di Bruno formula given in Eq. (D7) to a, so that

$$\partial_x^n e^{a(x)}\Big|_{x=t} = e^{a(t)} Y_n(\|A^{(1)}(t)\|,\ldots,\|A^{(n)}(t)\|) = Y_n(\|A^{(1)}(t)\|,\ldots,\|A^{(n)}(t)\|). \tag{E6}$$

On the other hand, we can split $a(t)$ into r steps and compute the nth derivative, just as for the Trotter product theorem:

$$\partial_x^n e^{a(x)}\Big|_{x=t} = \lim_{r\to\infty} \sum_S \binom{n}{s_1,\ldots,s_r} \prod_{q=1}^r \left(\frac{\|A^{(s_q)}(t)\|(1-\delta_{s_q,0})}{r} + \delta_{s_q,0}\right) \tag{E7}$$

By comparing the expressions given in Eqs. (E4) and (E7), we see that

$$\|\partial_t^n \exp\{A(t)\}\| \le \partial_x^n e^{a(x)}\Big|_{x=t}. \tag{E8}$$

Applying Eq. (E6), we reach our desired bound:

$$\|\partial_t^n \exp(A(t))\| \le Y_n(\|A^{(1)}(t)\|,\ldots,\|A^{(n)}(t)\|) \tag{E9}$$

■

Proof of Lemma 7. Without loss of generality, we take $t_0 = 0$. The relevant expressions are

$$U_{2,m}(t,0) = \sum_{j=1}^{m} a_j U_2^{(k_j)}(t,0) \tag{E10}$$

and

$$U_2^{(k)}(t,0) := \prod_{\ell=1}^{k} U_2(t_\ell, t_{\ell-1}), \tag{E11}$$

with $t_\ell := t\ell/k$. The Taylor remainder in integral form is given by

$$\begin{aligned} R_{2m} &= \frac{1}{(2m)!}\int_0^t (t-\tau)^{2m} \frac{d^{2m+1}}{d\tau^{2m+1}} U_{2,m}(\tau,0)d\tau \\ &= \frac{1}{(2m)!}\sum_{j=1}^{m} a_j \int_0^t (t-\tau)^{2m} \frac{d^{2m+1}}{d\tau^{2m+1}} U_2^{(k_j)}(\tau,0)d\tau. \end{aligned} \tag{E12}$$

With a couple of triangle inequalities, this is upper bounded as

$$\begin{aligned} \|R_{2m}\| &\le \frac{1}{(2m)!}\sum_{j=1}^{m} |a_j| \frac{t^{2m+1}}{2m+1} \max_{\tau\in[0,t]} \|\partial_\tau^{2m+1} U_2^{(k_j)}(\tau,0)\| \\ &\le \frac{\|a\|_1}{(2m+1)!} t^{2m+1} \max_{j,\tau} \|\partial_\tau^{2m+1} U_2^{(k_j)}(\tau,0)\|, \end{aligned} \tag{E13}$$

where in the last line we have employed a Hölder inequality. Our focus is now on bounding the derivative, which we unravel layer by layer using frequent multinomial expansions. First,

$$\partial_\tau^n U_2^{(k)}(\tau,0) = \sum_N \binom{n}{n_1,\ldots,n_k} \prod_{\ell=1}^{k} \partial_\tau^{n_\ell} U_2(\tau_\ell, \tau_{\ell-1}). \tag{E14}$$

Next, we write

$$\begin{aligned} U_2(\tau_\ell, \tau_{\ell-1}) &= \prod_{i=L}^{1} e^{-iH_i\alpha_i(\tau_{\ell-1/2})\tau/k} \prod_{i=1}^{L} e^{-iH_i\alpha_i(\tau_{\ell-1/2})\tau/k} \\ &= \prod_{i=1}^{2L} e^{A_{i,\ell}}, \end{aligned} \tag{E15}$$

where

$$A_{i,\ell} := -iH_i\alpha_i(\tau_{\ell-1/2})\tau/k \tag{E16}$$

and i is defined by reflection for $i > L$. Once again, performing a multinomial expansion,

$$\partial_\tau^n U_2(\tau_\ell, \tau_{\ell-1}) = \sum_N \binom{n}{n_1,\ldots,n_{2L}} \prod_{i=1}^{2L} \partial_\tau^{n_i} e^{A_{i,\ell}}. \tag{E17}$$

We now bound the individual ordinary-operator exponentials. Invoking Lemma 6,

$$\|\partial_\tau^n e^{A_{i,\ell}}\| \le Y_n\left(\|\partial_\tau A_{i,\ell}\|, \ldots, \|\partial_\tau^n A_{i,\ell}\|\right). \tag{E18}$$

In turn, we have

$$\begin{aligned}\partial_\tau^n A_{i,\ell} &= -i\frac{H_i}{k}\partial_\tau^n(\alpha_i(\tau_{\ell-1/2})\tau)\\ &= -i\frac{H_i}{k}\left[\left(\frac{\ell-1/2}{k}\right)^n \tau\alpha_i^{(n)}(\tau_{\ell-1/2}) + n\left(\frac{\ell-1/2}{k}\right)^{n-1}\alpha_i^{(n-1)}(\tau_{\ell-1/2})\right],\end{aligned} \tag{E19}$$

where $\alpha^{(n)}(x)$ refers to the nth derivative of α with respect to its argument, then evaluated at x (i.e., not a τ derivative). Since $\|H_i\| \le 1$, we have

$$\|\partial_\tau^n A_{i,\ell}\| < \frac{1}{k}(\ell/k)^{n-1}\left((\ell/k)\tau|\alpha_i^{(n)}(\tau_{\ell-1/2})| + n|\alpha_i^{(n-1)}(\tau_{\ell-1/2})|\right). \tag{E20}$$

From Definition 3, $\alpha_i^{(j)}(t) \le \Lambda_{i,n}(t)^{j+1}$. Dropping the n and t dependence for the moment,

$$\begin{aligned}\|\partial_\tau^n A_{i,\ell}\| &< (\ell/k)^n\left((\tau/k)\Lambda_i^{n+1} + (n/\ell)\Lambda_i^n\right)\\ &= (\Lambda_i\ell/k)^n\left(\Lambda_i\tau/k + n/\ell\right).\end{aligned} \tag{E21}$$

We have reached the bottom and now proceed to work our way back up to the Taylor remainder R_{2m}, starting with Eq. (E18). Using Eq. (D7) of Appendix D,

$$\begin{aligned}\|\partial_\tau^n e^{A_{i,\ell}}\| &\le \sum_C \frac{n!}{c_1!c_2!\ldots c_n!}\prod_{j=1}^n\left(\frac{\|\partial_\tau^j A_{i,\ell}\|}{j!}\right)^{c_j}\\ &< \sum_C \frac{n!}{c_1!c_2!\ldots c_n!}\prod_{j=1}^n\left(\frac{(\Lambda_i\ell/k)^j\,(\Lambda_i\tau/k + j/\ell)}{j!}\right)^{c_j}.\end{aligned} \tag{E22}$$

Using the sum rule for C, we can pull out a factor of $(\Lambda_i\ell/k)$. Using the upper bound $j \le n$ and the monotonicity of Y_n, we obtain the bound

$$\|\partial_\tau^n e^{A_{i,\ell}}\| < (\Lambda_i\ell/k)^n B_n(\Lambda_i\tau/k + n/\ell), \tag{E23}$$

where B_n is the Bell polynomial (see Appendix D). For simplicity, define

$$x_{i,\ell,n} = \Lambda_i\tau/k + n/\ell \tag{E24}$$

as the argument to B_n. Employing the bound given in Eq. (D14),

$$\|\partial_\tau^n e^{A_{i,\ell}}\| < (\Lambda_i\ell/k)^n\left(\frac{n}{\log\left(1 + n/x_{i,\ell,n}\right)}\right)^n, \tag{E25}$$

which is valid for all $n > 0$ and for $n = 0$ when defined by the 0^+ limit. We can simplify the reciprocal log with the bound

$$\begin{aligned}\frac{1}{\log\left(1 + n/x_{i,\ell,n}\right)} &< \left(\frac{1}{2} + \frac{x_{i,\ell,n}}{n}\right)^n\\ &= \frac{1}{2^n}\left(1 + \frac{2x_{i,\ell,n}}{n}\right)^n.\end{aligned} \tag{E26}$$

This gives us the simplified exponential derivative

$$\|\partial_\tau^n e^{A_{i,\ell}}\| < (\Lambda_i\ell/2k)^n\,(n + 2x_{i,\ell,n})^n. \tag{E27}$$

We now move up a level to reconsider Eq. (E17). Employing a triangle inequality,

$$\begin{aligned}\|\partial_\tau^n U_2(\tau_\ell,\tau_{\ell-1})\| &\le \sum_N \binom{n}{n_1,\ldots,n_{2L}} \prod_{i=1}^{2L} \|\partial_\tau^{n_i} e^{A_{i,\ell}}\| \\ &< \sum_N \binom{n}{n_1,\ldots,n_{2L}} \prod_{i=1}^{2L} (\Lambda_i \ell/2k)^{n_i} (n_i + 2x_{i,\ell,n_i})^{n_i}. \end{aligned} \tag{E28}$$

We maximize Λ_i over all $i = 1,\ldots,L$ and call it Λ. We can factor out the corresponding term and, with some rewriting, obtain

$$(\Lambda\ell/2k)^n \sum_N \binom{n}{n_1,\ldots,n_{2L}} \prod_{i=1}^{2L} (n_i + 2x_{\ell,n_i})^{n_i}, \tag{E29}$$

where we have also let x_{ℓ,n_i} be x_{i,ℓ,n_i}, with the subscript dropped on Λ_i.Focusing on the rightmost product over i, one can show using a Lagrange multiplier that the maximum is given by $n_i = n/2L$ for all i (we maximize over $n_i \in \mathbb{R}_+$, which is an upper bound). This is intuitive from the symmetry of the product as well. Taking this as an upper bound, we have

$$\begin{aligned}\|\partial_\tau^n U_2(\tau_\ell,\tau_{\ell-1})\| &< (\Lambda\ell/2k)^n \left(\frac{n}{2L} + \frac{2\Lambda\tau}{k} + \frac{n}{L\ell}\right)^n \sum_N \binom{n}{n_1,\ldots,n_{2L}} \\ &= (\Lambda\ell/2k)^n \left(n + \frac{4\Lambda\tau L}{k} + \frac{2n}{\ell}\right)^n \\ &= (\Lambda/k)^n \left(n + n\ell/2 + \frac{2\Lambda\tau L\ell}{k}\right)^n, \end{aligned} \tag{E30}$$

where in going to the second line we have evaluated the multinomial sum as $(2L)^n$ and simplified.

With this in hand, we return to Eq. (E14) and bound it as

$$\begin{aligned}\|\partial_\tau^n U_2^{(k)}(\tau,0)\| &\le \sum_N \binom{n}{n_1,\ldots,n_k} \prod_{\ell=1}^{k} \|\partial_\tau^{n_\ell} U_2(\tau_\ell,\tau_{\ell-1})\| \\ &< \sum_N \binom{n}{n_1,\ldots,n_k} \prod_{\ell=1}^{k} (\Lambda/k)^{n_\ell} \left(n_\ell + n_\ell\ell/2 + \frac{2\Lambda\tau L\ell}{k}\right)^{n_\ell}. \end{aligned} \tag{E31}$$

Using the upper bound $\ell \le k$ and factoring out the $(\Lambda/k)^{n_\ell}$ using the sum rule,

$$\|\partial_\tau^n U_2^{(k)}(\tau,0)\| < (\Lambda/k)^n \sum_N \binom{n}{n_1,\ldots,n_k} \prod_{\ell=1}^{k} (n_\ell + n_\ell k/2 + 2\Lambda\tau L)^{n_\ell}. \tag{E32}$$

Similarly, we upper bound the product using $n_\ell = n/k$ for all ℓ, which can be justified through a maximization using Lagrange multipliers. The corresponding bound is

$$\begin{aligned}\|\partial_\tau^n U_2^{(k)}(\tau,0)\| &< (\Lambda/k)^n (n/k + n/2 + 2\Lambda\tau L)^n \sum_N \binom{n}{n_1,\ldots,n_k} \\ &= (n\Lambda)^n \left(\frac{1}{k} + \frac{1}{2} + \frac{2\Lambda\tau L}{n}\right)^n. \end{aligned} \tag{E33}$$

We are finally ready to return to Eq. (E13) and bound R_{2m}. We recall that Λ has τ dependence and let $\Lambda_{\max} := \max_{\tau\in[0,t]} \Lambda(\tau)$. We also upper bound any appearance of τ otherwise by t, because these are always in the numerator. So far, using $n = 2m + 1$, these reductions give

$$\|R_{2m}\| < \frac{\|a\|_1}{(2m+1)!}((2m+1)\Lambda_{\max}t)^{2m+1} \max_j \left(\frac{1}{k_j}+\frac{1}{2}+\frac{2\Lambda tL}{2m+1}\right)^{2m+1}. \tag{E34}$$

Employing a Stirling bound on the factorial and factoring out an additional L from the rightmost term,

$$\|R_{2m}\| < \frac{\|a\|_1}{\sqrt{2\pi(2m+1)}}(eL\Lambda_{\max}t)^{2m+1} \max_j \left(\frac{1}{Lk_j}+\frac{1}{2L}+\frac{2\Lambda t}{2m+1}\right)^{2m+1}. \tag{E35}$$

We now apply the assumption that $eL\Lambda_{\max}t < 1$ to upper bound the $\max_j$ term, along with $k_j, L \geq 1$:

$$\max_j \left(\frac{1}{Lk_j}+\frac{1}{2L}+\frac{2\Lambda t}{2m+1}\right)^{2m+1} < \left(\frac{3}{2}+\frac{2}{3e}\right)^{2m+1}. \tag{E36}$$

Thus,

$$\begin{aligned}\|R_{2m}\| &< \frac{\|a\|_1}{2\sqrt{\pi m}}\left(\left(\frac{3e}{2}+\frac{2}{3}\right)L \max_{\tau\in[0,t]}\Lambda_{2m+1}(\tau)t\right)^{2m+1} \\ &< \frac{\|a\|_1}{2\sqrt{\pi m}}\left(5L \max_{\tau\in[0,t]}\Lambda_{2m+1}(\tau)t\right)^{2m+1}.\end{aligned} \tag{E37}$$

In these last lines, we remind ourselves that Λ has the subscript $2m+1$ as per Definition 3. ■

APPENDIX F: GREEDY ALGORITHM FOR ADAPTIVE TIME STEPS

Here, we discuss schemes for constructing the adaptive nonuniform mesh of time steps used in the MPF algorithm described in Sec. VI A. Specifically, we seek a decomposition of the desired simulation interval $[0,T]$ into a monotonically increasing sequence of times $t_0, t_1, \ldots, t_r$, with $t_0 = 0$, $t_r = T$. The mesh construction of Sec. VI C, although theoretically sound, is not directly implementable, since it requires knowing the total number of steps while constructing each new point based on local data. To avoid this issue, as well as the restriction $|\dot{\Lambda}(\tau)| \leq K\Lambda^2(\tau)$, we seek a simple-to-use greedy algorithm.

One possibility is to use a direct approach that first selects a candidate number of steps r_{try}. Starting from $r_{\text{try}} = 1$, we then build recursively a sequence of times using the inequality given in Eq. (125) of Sec. VI C,

$$L \max_{t\in[t_{i-1},t_i]}\Lambda_{2m+1}(t)\,(t_i - t_{i-1}) \leq \frac{1}{5}\left(\frac{\epsilon\sqrt{\pi m}}{\|a\|_1 r},\right)^{1/(2m+1)} \tag{F1}$$

with $r = r_{\text{try}}$. Starting from $t_0 = 0$ and looking for the largest t_i that satisfies the condition, we finally check whether the generated number of intervals is greater than r_{try}, in which case we increase r_{try} by one and repeat. When the algorithm stops at the optimal value r_{opt}, we have performed a total of $r_{\text{opt}}(r_{\text{opt}}+1)/2$ nonlinear optimization steps, each one requiring multiple evaluations of the left-hand side of Eq. (F1). This can be very demanding when the left-hand side of Eq. (F1) is expensive to evaluate and the optimal number of intervals is around half the upper bound

$$r_{\max} = \left(5LT \max_{t\in[0,T]}\Lambda_{2m+1}(t)\right)^{\frac{2m+1}{2m}}\left(\frac{\|a\|_1}{\epsilon\sqrt{\pi m}}\right)^{\frac{1}{2m}} \tag{F2}$$

obtained considering identical intervals and bounding $\Lambda(t)$ with its maximum value over the whole simulation interval $[0,T]$. In this case, finding an approximation to the optimal decomposition requires $O(r_{\max}^2)$ optimization steps, each one requiring multiple evaluations of the left-hand side of Eq. (F1).

We now describe an alternative approach that determines r_{opt} within a factor of 2 and uses only $r_{\max}$ evaluations of $\max_{t\in[t_{i-1},t_i]}\Lambda(t)$ and additional $O(\log(r_{\max})r_{\max})$ simple arithmetic operations. This procedure can be used to find a viable, and approximately optimal, decomposition of the time interval or as a good starting point to find the optimal one using a procedure such as the one described above. The idea is to start by decomposing the interval $[0,T]$ into $r_{\max}$ segments with equal length and storing the maximum of $\Lambda(t)$ in each segment in an array A of size $r_{\max}$. We then introduce an additional array of the

same size,

$$L_m = \left[\max_{k\le m} A_k\right] m\frac{T}{r_{\max}}, \tag{F3}$$

together with an additional set of vectors of the same size,

$$R_m^{(n)} = \left[\max_{n\ge k>m} A_k\right] (n-m)\frac{T}{r_{\max}}, \tag{F4}$$

where n is an additional index between 1 and $r_{\max}$. The first vector stores the left-hand side of the inequality given in Eq. (F1) for the interval up to the mth time, while the second vector stores the same information for the interval starting at the mth time and ending at the nth one. The algorithm proceeds by splitting the time interval recursively into two parts, so that the left-hand side of Eq. (F1) takes (approximately) the same value on both halves (i.e., we are splitting the error equally on both sides). At every iteration, the number of intervals doubles and the right-hand side of Eq. (F1) shrinks accordingly. We stop the procedure once the inequality in Eq. (F1) is satisfied on one interval (since we are guaranteed that it will in all others). The procedure will stop at some r_K, at which point we know that the optimal value r_{opt} is in $[\lceil r_K/2\rceil, r_K]$. The algorithm can then be described as follows:

(1) Compute L_m for all $m = 1,\ldots,r_{\max}$.
(2) Set $n = r_{\max}$ and $r = 2$.
(3) Compute the elements of $R_m^{(n)}$ for all $m = 1,\ldots, n-1$.
(4) Initialize an auxiliary array D_m as $D_m = L_m - R_m^{(n)}$.
(5) Find the least index k for which $D_k > 0$.
(6) If L_k is less than the right-hand side of the bound in Eq. (F1) with the current value of r, set $r_K = r$ and exit.
(7) If $2r \ge r_{\max}$, set $r_K = r_{\max}$ and exit.
(8) Set $r = 2r$ and $n = k$ and repeat from step (3).

Step (1) requires $r_{\max}$ operations while steps (3) and (4) cost n operations each. Since the number of iterations is bounded by $\log_2(r_{\max})$, their combined cost is bounded by $2\log_2(r_{\max})r_{\max}$. If we use binary search, step (5) costs $\log_2(n)$ operations, so its total cost is at most $\log_2(r_{\max})^2$ operations. From this analysis, we see that steps (3) and (4) are the most expensive ones and they dominate the cost of the scheme. On exit, we have $r_K \approx r_{\text{opt}}$ together with the first interval $[t_0, t_1]$. The rest of the intervals can then be found keeping $r = r_K$ fixed with additional $O(r_{\max})$ operations.

APPENDIX G: DETAILS FOR NUMERICAL DEMONSTRATIONS

We begin by discussing the frame transformation that allows for a direct comparison to the exact (machine-precision) time evolution. Suppose that H is a time-independent Hamiltonian with propagator $U(t)$. Under a unitary frame transformation $T(t)$, the Hamiltonian and propagator transform as

$$\begin{aligned}\tilde{U}(t) &= T(t)U(t),\\ \tilde{H}(t) &= i\frac{\partial T(t)}{\partial t}T(t)^\dagger + T(t)HT(t)^\dagger.\end{aligned} \tag{G1}$$

Thus, in order to benchmark the error of the MPF, we compute $\tilde{U}_k$ for Hamiltonian $\tilde{H}$ and then compare with the exact propagator (accurate to machine precision):

$$\epsilon_c = \|\tilde{U}_{\tilde{k}}(t) - T(t)U(t)\|. \tag{G2}$$

Choosing such a special form of the time dependence has implications for the Λ-bound. In our examples, we will choose $T(t) = e^{-iKt}$ for some K, and in such a case we find $\tilde{H}(t) = K + e^{-it\mathrm{ad}_K}H$, where $\mathrm{ad}_K^n(\cdot) = [K,\cdot]$. Taking derivatives, we find that Λ can be chosen such that

$$\Lambda \ge \max_n \left\{\|K+H\|,\ \sqrt[n+1]{\|\mathrm{ad}_K(H)\|}\right\} \tag{G3}$$

for $n > 0$. Having such a simple form for Λ is nice but clearly does not capture the full range of time dependencies that one expects to find in practice.

1. Example 1 details

Choose a coordinate system such that B makes an angle θ with respect to the z axis and lies within the x-z plane. The spin-1/2 system can be described by the Hamiltonian

$$H = \mu B(\cos\theta Z/2 + \sin\theta X/2), \tag{G4}$$

where Z and X (and later Y) are Pauli operators and μ is a coupling parameter that will henceforth be set to one. The propagator $U(t) = e^{-iHt}$ is easy to compute and corresponds to precession about the magnetic field axis with frequency B. To obtain a time-dependent problem, let us shift to a reference frame that rotates with angular frequency ω about the z axis. The transformation is given by $R_z(\omega t)$, where R_a is the usual $SU(2)$ rotation operator about axis a. This leads to the Hamiltonian considered in the main text. Because we know that this Hamiltonian is just a transformed time-independent system, it is easy to compute the exact propagator $\tilde{U}(t)$:

$$\tilde{U}(t) = R_z(\omega t)U(t). \tag{G5}$$

2. Example 2 details

We seek a time-independent Hamiltonian $H = H_0 + H_1$ that produces nontrivial time dependence in the interaction picture. We also ask that it satisfies a simple conservation law. A special instance of the 1D XX model will suffice to meet these conditions. Consider a circular chain of N qubits with nearest-neighbor-hopping interactions, with Hamiltonian $H = H_0 + H_1$ of the form

$$\begin{aligned} H_0 &= \sum_{k=1}^{N} \frac{\omega_k}{2} Z_k, \\ H_1 &= \sum_{k=1}^{N} \frac{J_k}{2} \left(X_k X_{k+1} + Y_k Y_{k+1}\right). \end{aligned} \tag{G6}$$

Here, ω_k, J_k are real site-dependent parameters and any index increments are done modulo N. For any value of the parameters, the Hamiltonian conserves the total magnetization $\mu := \sum_k Z_k$:

$$[\mu, H] = 0. \tag{G7}$$

Conceptually, we will think of H_0 as a "base" Hamiltonian, with perturbation H_1 generating interactions, though we make no assumptions as to the smallness of H_1. We will switch to an interaction picture that is comoving with the simple dynamics of H_0. In this frame, the Hamiltonian $\tilde{H}(t)$ is given by

$$\begin{aligned} \tilde{H}(t) &= e^{iH_0 t} H_1 e^{-iH_0 t} \\ &= \sum_{k=1}^{N} \frac{J_k}{2} \left(X_k(t) X_{k+1}(t) + Y_k(t) Y_{k+1}(t)\right), \end{aligned} \tag{G8}$$

where

$$\begin{aligned} X_k(t) &:= e^{iH_0 t} X_k e^{-iH_0 t} = \cos(\omega_k t) X_k - \sin(\omega_k t) Y_k, \\ Y_k(t) &:= e^{iH_0 t} Y_k e^{-iH_0 t} = \cos(\omega_k t) Y_k + \sin(\omega_k t) X_k \end{aligned} \tag{G9}$$

correspond to rotating the Pauli vectors about the z axis with frequency ω_k. We can express Eq. (G8) in terms of the time-independent X_k and Y_k of the original frame,

$$\begin{aligned} \tilde{H}(t) = \sum_{k=1}^{N} \frac{J_k}{2} \big\{ &\cos(\Delta\omega_k t)(X_k X_{k+1} + Y_k Y_{k+1}) \\ &+ \sin(\Delta\omega_k t)(X_k Y_{k+1} - Y_k X_{k+1}) \big\}, \end{aligned} \tag{G10}$$

where $\Delta\omega_k = \omega_{k+1} - \omega_k$. We see that having different qubit frequencies ω_k on neighboring sites should give rise to a nontrivial time dependence in $\tilde{H}$. Another indication is gleaned from the commutator of H_0 and H_1:

$$[H_0, H_1] = -i \sum_k \frac{J_k}{2} (X_k Y_{k+1} - Y_k X_{k+1})(\Delta\omega_k) \tag{G11}$$

The time dependence in H_I will be nontrivial when the commutator does not vanish, as occurs when $\Delta\omega_k \neq 0$. A simple choice is to set

$$J_k = J, \quad \omega_k = (-1)^k \omega. \tag{G12}$$

That is, the qubit frequency alternates sign at each site and the coupling is translation invariant. For simplicity, we consider only even numbers of qubits, to avoid frequency matching at $k = N$. These choices lead to the Hamiltonian considered in the main text.

We conclude this appendix with a proof of the theorem regarding the unitary approximation order of the MPF.

Proof of Theorem 6. We suppress all function evaluations at t when convenient. Let $E := U_{2,m} - U$, so that $U_{2,m} = U + E$. Then, using the unitarity of U and the fact that $E \in O(t^{2m+1})$,

$$U_{2,m}^\dagger U_{2,m} = \mathbb{1} + N + O(t^{4m+2}), \tag{G13}$$

where

$$N := U^\dagger E + E^\dagger U. \tag{G14}$$

Since $N \in O(t^{2m+1})$, all of its derivatives up to degree $2m$ vanish when evaluated at $t = 0$. Hence, it suffices to show that

$$N^{(2m+1)}(0) = 0. \tag{G15}$$

We can expand this derivative in terms of E and U using the binomial theorem. When we evaluate at $t = 0$, those terms with derivative less than degree $2m + 1$ in E vanish. We are left with

$$N^{(2m+1)}(0) = E^{\dagger(2m+1)}(0) U(0) + U^\dagger(0) E^{(2m+1)}(0). \tag{G16}$$

We have $U(0) = U^\dagger(0) = \mathbb{1}$. Moreover, by the time-symmetric property of U and $U_{2,m}$, $E(t)$ is also symmetric. Therefore,

$$E^{\dagger(2m+1)}(0) = E^{(2m+1)}(-t)\Big|_{t=0} = -E^{(2m+1)}(0). \tag{G17}$$

Hence, the two terms in Eq. (G16) cancel, yielding $N^{(2m+1)}(0) = 0$. This completes the proof. ■

[1] R. P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. **21** (1982).

[2] S. Lloyd, Universal quantum simulators: Correction, Science **279**, 1113 (1998).

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

[4] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 901 (2012).

[5] 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).

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

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

[8] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing* (New York, NY, 2014), pp. 283–292.

[9] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri *et al.*, Towards quantum chemistry on a quantum computer, Nat. Chem. **2**, 106 (2010).

[10] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, Proc. Natl. Acad. Sci. **114**, 7555 (2017).

[11] V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, Quantum computing enhanced computational catalysis, Phys. Rev. Res. **3**, 033055 (2021).

[12] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even more efficient quantum computations of chemistry through tensor hypercontraction, PRX Quantum **2**, 030305 (2021).

[13] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-depth quantum simulation of materials, Phys. Rev. X **8**, 011044 (2018).

[14] A. Roggero, A. C. Y. Li, J. Carlson, R. Gupta, and G. N. Perdue, Quantum computing for neutrino-nucleus scattering, Phys. Rev. D **101**, 074038 (2020).

[15] B. Hall, A. Roggero, A. Baroni, and J. Carlson, Simulation of collective neutrino oscillations on a quantum computer, Phys. Rev. D **104**, 063009 (2021).

[16] A. Baroni, J. Carlson, R. Gupta, A. C. Y. Li, G. N. Perdue, and A. Roggero, Nuclear two point correlation functions on a quantum computer, Phys. Rev. D **105**, 074503 (2022).

[17] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum algorithms for quantum field theories, Science **336**, 1130 (2012).

[18] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, Quantum-classical computation of Schwinger model dynamics using quantum computers, Phys. Rev. A **98**, 032331 (2018).

[19] A. F. Shaw, P. Lougovski, J. R. Stryker, and N. Wiebe, Quantum algorithms for simulating the lattice Schwinger model, Quantum **4**, 306 (2020).

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

[21] 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).

[22] M. Kieferová, A. Scherer, and D. W. Berry, Simulating the dynamics of time-dependent Hamiltonians with a truncated Dyson series, Phys. Rev. A **99**, 042314 (2019).

[23] D. W. Berry, A. M. Childs, Y. Su, X. Wang, and N. Wiebe, Time-dependent Hamiltonian simulation with L^1-norm scaling, Quantum **4**, 254 (2020).

[24] K. Mizuta and K. Fujii, Optimal Hamiltonian simulation for time-periodic systems, Quantum **7**, 962 (2023).

[25] J. K. Hale and H. Koçak, *Scalar Nonautonomous Equations* (Springer, New York, 1991), p. 107.

[26] G. Casati and L. Molinari, "Quantum chaos" with time-periodic Hamiltonians, Prog. Theor. Phys. Supp. **98**, 287 (1989).

[27] U. Peskin and N. Moiseyev, The solution of the time-dependent Schrödinger equation by the (t, t') method: Theory, computational algorithm and applications, J. Chem. Phys. **99**, 4590 (1993).

[28] M. Suzuki, Methodology of analytic and computational studies on quantum systems, J. Stat. Phys. **110**, 945 (2003).

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

[30] Y. Atia and D. Aharonov, Fast-forwarding of Hamiltonians and exponentially precise measurements, Nat. Commun. **8**, 1572 (2017).

[31] A. Rajput, A. Roggero, and N. Wiebe, Hybridized methods for quantum simulation in the interaction picture, Quantum **6**, 780 (2022).

[32] D. An, D. Fang, and L. Lin, Time-dependent Hamiltonian simulation of highly oscillatory dynamics and superconvergence for Schrödinger equation, Quantum **6**, 690 (2022).

[33] J. D. Dollard and C. N. Friedman, in *Number V. 10. Section, Analysis in Encyclopedia of Mathematics and Its Applications* (Cambridge University Press, Cambridge, 1984).

[34] S. A. Chin, Multi-product splitting and Runge-Kutta-Nyström integrators, Celest. Mech. Dyn. Astron. **106**, 391 (2010).

[35] G. H. Low, V. Kliuchnikov, and N. Wiebe, Well-conditioned multiproduct Hamiltonian simulation. arXiv: Quantum Physics, 2019.

[36] S. Blanes, F. Casas, and J. Ros, Extrapolation of symplectic integrators, Celest. Mech. Dyn. Astron. **75**, 149 (1999).

[37] A. Sidi, in *Cambridge Monographs on Applied and Computational Mathematics* (Cambridge University Press, Cambridge, UK, 2003), p. 21.

[38] N. Klco and M. J. Savage, Minimally entangled state preparation of localized wave functions on quantum computers, Phys. Rev. A **102**, 012612 (2020).

[39] A. G. Rattew, Y. Sun, P. Minssen, and M. Pistoia, The efficient preparation of normal distributions in quantum registers, Quantum **5**, 609 (2021).

[40] A. G. Rattew and B. Koczor, Preparing arbitrary continuous functions in quantum registers with logarithmic complexity, arXiv preprint arXiv:2205.00519.

[41] J. Iaconis, S. Johri, and E. Y. Zhu, Quantum state preparation of normal distributions using matrix product states, npj Quantum Inf. **10**, 15 (2024).

[42] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv preprint arXiv:quant-ph/0208112.

[43] A. Kitaev and W. A. Webb, Wavefunction preparation and resampling using a quantum computer, arXiv preprint arXiv:0801.0342.

[44] C. W. Bauer, P. Deliyannis, M. Freytsis, and B. Nachman, Practical considerations for the preparation of multivariate Gaussian states on quantum computers, arXiv preprint arXiv:2109.10918.

[45] G. Rendon, J. Watkins, and N. Wiebe, Improved accuracy for Trotter simulations using Chebyshev interpolation, Quantum **8**, 1266 (2024).

[46] P. K. Faehrmann, M. Steudtner, R. Kueng, M. Kieferova, and J. Eisert, Randomizing multi-product formulas for Hamiltonian simulation, Quantum **6**, 806 (2022).

[47] S. Zhuk, N. F. Robertson, and S. Bravyi, Trotter error bounds and dynamic multi-product formulas for Hamiltonian simulation, Phys. Rev. Res. **6**, 033309 (2024).

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

[49] D. Berend and T. Tassa, Improved bounds on Bell numbers and on moments of sums of random variables, Probab. Math. Stat. **30**, 185 (2010).

[50] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).

[51] D. W. Berry, A. M. Childs, and R. Kothari, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (IEEE, Berkeley, CA, 2015), p. 792.

[52] G. H. Low and N. Wiebe, Hamiltonian simulation in the interaction picture. arXiv preprint arXiv:1805.00675.

[53] N. Wiebe and N. S. Babcock, Improved error-scaling for adiabatic quantum evolutions, New J. Phys. **14**, 013024 (2012).

[54] B. C. Hall, in Volume 267 of *Graduate Texts in Mathematics* (Springer New York, NY, 2013).

[55] B. Simon and M. C. Reed, *Methods of Modern Mathematical Physics* (Academic Press, San Diego, CA, 1980), Vol. 1.

[56] L. Comtet, *Advanced Combinatorics: The Art of Finite and Infinite Expansions* (Springer Science & Business Media, New York, NY, 2012).

[57] T. D. Ahle, Sharp and simple bounds for the raw moments of the binomial and Poisson distributions, Stat. Probab. Lett. **182**, 109306 (2022).