

Quasi Monte Carlo method for linear combination unitaries via classical postprocessing

Yuya Kawamata^{1,*}, Kosuke Mitarai^{1,2,†} and Keisuke Fujii^{1,2,3,‡}

¹*Department of Systems Innovation, Graduate School of Engineering Science,
The University of Osaka, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan*

²*Center for Quantum Information and Quantum Biology, The University of Osaka, 1-2 Machikaneyama, Toyonaka 560-0043, Japan*

³*RIKEN Center for Quantum Computing (RQC), Hirosawa 2-1, Wako, Saitama 351-0198, Japan*



(Received 19 September 2025; accepted 18 April 2026; published 13 May 2026)

We propose the quasi Monte Carlo method for linear combination of unitaries via classical postprocessing (LCU-CPP) on quantum applications. The LCU-CPP framework has been proposed as an approach to reduce hardware resources, expressing a general target operator $F(A)$ as $F(A) = \int_V f(t) G(A, t) dt$, where each $G(A, t)$ is proportional to a unitary operator. On a quantum device, $\text{Re}[\text{Tr}(G(A, t)\rho)]$ can be estimated using the Hadamard test and then combined through classical integration, allowing for the realization of nonunitary functions with reduced circuit depth. While previous studies have employed the Monte Carlo method or the trapezoid rule to evaluate the integral in LCU-CPP, we show that the quasi Monte Carlo method can achieve even lower errors. In two numerical experiments, ground-state property estimation and Green's function estimation, the quasi Monte Carlo method achieves the lowest errors with a number of Hadamard test shots per unitary that is practical for real hardware implementations. These results indicate that quasi Monte Carlo is an effective integration strategy within the LCU-CPP framework.

DOI: [10.1103/nmky-mykb](https://doi.org/10.1103/nmky-mykb)

I. INTRODUCTION

Quantum computers hold the promise of surpassing classical computers in terms of computational power, offering revolutionary advances in fields such as quantum chemistry [1,2], financial engineering [3,4], and combinatorial optimization [5]. Nevertheless, most algorithms with provable quantum speedups, such as the quantum linear system solver [6] or Hamiltonian simulation [7], require fault-tolerant quantum computing and extensive error correction, which in turn demand substantial hardware resources [8–10]. To bridge the gap between near-term devices and these resource-heavy algorithms, it is therefore critical to identify techniques that significantly reduce quantum hardware requirements [11].

One promising approach is to off-load as much work as possible to classical postprocessing. In particular, the “linear combination of unitaries via classical postprocessing” (LCU-CPP) framework [12–21] lets us estimate quantities of the form $\text{Tr}[F(A)\rho]$, where ρ is an N -qubit quantum state and A is a Hermitian matrix (e.g., a Hamiltonian), even when $F(A)$ itself is nonunitary. Concretely, one selects an integral representation $F(A) = \int_V f(t) G(A, t) dt$, where $t \in \mathbb{R}^d$, V is a d -dimensional hyperrectangular domain, $f(t)$ is a

probability density function over V , and each $G(A, t)$ is a unitary. Because $\text{Tr}[F(A)\rho]$ is real, one can write $\text{Tr}[F(A)\rho] = \int_V f(t) \text{Re}[\text{Tr}(G(A, t)\rho)] dt$. On a quantum device, each term $\text{Re}[\text{Tr}(G(A, t)\rho)]$ is estimated by a Hadamard test. These estimates are then combined by classical numerical integration. In this way, complicated nonunitary functions, such as A^{-1} for matrix inversion [14,18,20], $e^{-\beta A}$ for Gibbs state sampling [13], or filtered ground-state projectors for property estimation [12,14,19], can be realized with reduced quantum circuit depth. Despite this flexibility, prior LCU-CPP studies simply invoke either the Monte Carlo (MC) method or the trapezoid rule for the integration, without comparing these options in detail. However, the overall accuracy and resource cost of LCU-CPP depend sensitively on how one carries out the integral over t .

In this work, we present the quasi Monte Carlo (QMC) method [22] for LCU-CPP. More specifically, we first show the upper bounds for the errors under QMC, MC, and the trapezoid rule. When the number of Hadamard test shots per unitary is M , the analysis indicates that MC and QMC achieve the lowest error when $M = 1$, QMC is most advantageous when M is in a moderate range, and the trapezoid rule becomes favorable in the limit as $M \rightarrow \infty$. To clarify what range of M is practically moderate, we carry out numerical experiments on ground-state property estimation and Green's function estimation. The results show that QMC attains the smallest error around $M = 10^3$ in the former and around $M = 10^2$ in the latter. These shot counts can be considered reasonable for current hardware once execution and communication latencies are taken into account [23,24]. In conclusion, QMC provides an effective integration strategy within the LCU-CPP framework.

The rest of the paper is organized as follows. In Sec. II, we review the LCU-CPP framework and the error analysis of MC

*Contact author: u597035b@ecs.osaka-u.ac.jp

†Contact author: mitarai.kosuke.es@osaka-u.ac.jp

‡Contact author: fujii.keisuke.es@osaka-u.ac.jp

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

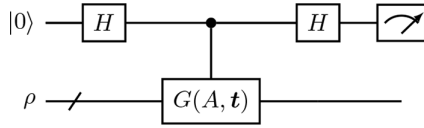


FIG. 1. Quantum circuit for the Hadamard test. The measurement outcome $y \in +1, -1$ satisfies $\Pr[y = +1] = (1 + x)/2$ and $\Pr[y = -1] = (1 - x)/2$, where $x = \text{Re}[\text{Tr}(G(A, t)\rho)]$. For more details, please refer to Appendix A.

and the trapezoid rule. In Sec. III, we introduce LCU-CPP via QMC and provide its advantages. Section IV presents numerical experiments on ground-state properties and Green's function, highlighting the practical gains of QMC. Finally, Sec. V summarizes our conclusion.

II. PRELIMINARIES

In this section, we briefly review prior studies as preliminaries. We first explain the framework of LCU-CPP and then error estimates for LCU-CPP when using MC and the trapezoid rule, respectively.

A. Linear combination of unitaries via classical postprocessing

We consider the problem of estimating $\text{Tr}(F(A)\rho)$ for an N -qubit quantum state ρ , a Hermitian matrix $A \in \mathbb{C}^{2^N \times 2^N}$ such as a Hamiltonian, and a general function $F : \mathbb{C}^{2^N \times 2^N} \rightarrow \mathbb{C}^{2^N \times 2^N}$. To tackle this problem, the LCU-CPP framework relies on the following integral representation:

$$F(A) = \int_V f(\mathbf{t}) G(A, \mathbf{t}) d\mathbf{t}, \quad (1)$$

where $\mathbf{t} \in \mathbb{R}^d$, V is a d -dimensional hyperrectangular domain $V = \prod_{j=1}^d [a_j, b_j]$ of volume $|V| = \prod_{j=1}^d (b_j - a_j)$, $f(\mathbf{t})$ is a probability density function, and $G(A, \mathbf{t})$ is a constant multiplied unitary operator parametrized by A and $\mathbf{t}$. While this integral expansion is not universally applicable for every possible $F(A)$, many important cases do admit such a form, as we see below. From Eq. (1), it follows that

$$\text{Tr}(F(A)\rho) = \int_V f(\mathbf{t}) \text{Tr}(G(A, \mathbf{t})\rho) d\mathbf{t}. \quad (2)$$

Since $F(A)$ is assumed to be Hermitian, $\text{Tr}(F(A)\rho)$ is real, which allows us to write $\text{Tr}(F(A)\rho) = \int_V f(\mathbf{t}) \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)] d\mathbf{t}$. Because $\text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]$ can be estimated via the Hadamard test (see Fig. 1), we can estimate $\text{Tr}(F(A)\rho)$ by numerical integration over V . We refer to this procedure as LCU-CPP. This idea appears in a number of studies, dating back at least to 1999 [25]. It is worth noting that one could also implement these integrals by performing linear combinations of unitaries entirely on a quantum computer, as in Ref. [26]; however, using the Hadamard test together with classical postprocessing typically reduces the hardware resources. Below, we briefly highlight three representative applications of this approach.

Microcanonical ensemble ($d = 1$). First, we consider estimating the expectation value of a unitary O in a microcanonical ensemble at energy E using LCU-CPP [15,16],

which involves calculating $\text{Tr}(Oe^{-(A-E)^2\tau^2}\rho)$ with $\rho = I/2^N$. As in Eq. (2), we use

$$F(A) = Oe^{-(A-E)^2\tau^2}, \quad (3)$$

$$f(t) = \frac{1}{2\tau\sqrt{\pi}} e^{-\frac{t^2}{4\tau^2}}, \quad (4)$$

$$G(A, t) = Oe^{-i(A-E)t}, \quad (5)$$

where τ is a constant. The integration domain is an infinite domain, which is reduced to a finite region by introducing an appropriate cutoff.

Ground-state property estimation ($d = 2$). Next, we consider estimating the expectation value of a unitary O on the ground state ρ_g using LCU-CPP [12,14,19], which involves calculating $\text{Tr}(e^{-A^2\tau^2} Oe^{-A^2\tau^2}\rho)$. We prepare an unnormalized ground state by applying a Gaussian function to a state ρ that has nonzero overlap with the ground state. As in Eq. (2), we use

$$F(A) = e^{-A^2\tau^2} Oe^{-A^2\tau^2}, \quad (6)$$

$$f(\mathbf{t}) = \frac{1}{4\tau^2\pi} e^{-\frac{t_1^2 + t_2^2}{4\tau^2}}, \quad (7)$$

$$G(A, \mathbf{t}) = e^{-iAt_1} Oe^{-iAt_2}, \quad (8)$$

where $\mathbf{t} = (t_1, t_2)$. The integration domain is an infinite domain in two dimensions, which is reduced to a finite region by introducing appropriate cutoffs.

Linear system solver ($d = 4$). Finally, we consider estimating $\text{Tr}(A^{-1}OA^{-1}\rho)$ using LCU-CPP for an invertible Hermitian matrix A and a unitary O [14]. Here, A is rescaled so that $\text{spec}(A) \subset [-1, -\mu] \cup [\mu, 1]$ with $\mu > 0$. The integration domain is $V = [0, L_1] \times [-L_2, L_2] \times [0, L_3] \times [-L_4, L_4]$, and we define $Z(L) := 1 - e^{-L^2/2}$ and $\text{sign}(x)$ as the sign function. With normalization chosen so that f is a probability density after truncation, we set, as in Eq. (2),

$$F(A) = A^{-1}OA^{-1}, \quad (9)$$

$$f(\mathbf{t}) = \frac{1}{4L_1L_3Z(L_2)Z(L_4)} |t_2t_4| e^{-\frac{t_2^2 + t_4^2}{2}}, \quad (10)$$

$$G(A, \mathbf{t}) = -\frac{2Z(L_2)Z(L_4)}{\pi} \times \text{sign}(t_2t_4) L_1L_3 e^{-it_1t_2A} Oe^{-it_3t_4A}, \quad (11)$$

where $\mathbf{t} = (t_1, t_2, t_3, t_4)$. This representation recovers the untruncated one as $L_j \rightarrow \infty$.

B. LCU-CPP via Monte Carlo method

MC is widely used for numerical integration and has been employed in several works [12,13,15,17]. Within this framework, one can estimate $\text{Tr}(F(A)\rho)$ by sampling random $\mathbf{t}$ according to the function $f(\mathbf{t})$. More concretely, we first draw K samples $\mathbf{t}_k^{\text{mc}}$ from the distribution proportional to $f(\mathbf{t})$. On each $\mathbf{t}_k^{\text{mc}}$, we perform the Hadamard test M times to obtain estimators v_k^{mc} that satisfy $|v_k^{\text{mc}} - \text{Re}[\text{Tr}(G(A, \mathbf{t}_k^{\text{mc}})\rho)]| \leq O(1/\sqrt{M})$ with high probability. Finally, we form the

estimator $v^{\text{mc}} = \frac{1}{K} \sum_{k=1}^K v_k^{\text{mc}}$, which serves as an approximation to $\text{Tr}(F(A)\rho)$. In Appendix B, we prove that

$$\text{Ex}[v^{\text{mc}}] = \text{Tr}(F(A)\rho), \quad (12)$$

$$\text{Var}[v^{\text{mc}}] = \frac{C^{\text{mc,ht}}}{MK} + \frac{C^{\text{mc,ni}}}{K}, \quad (13)$$

where

$$C^{\text{mc,ht}} = 1 - \int_V f(t) \text{Re}[\text{Tr}(G(A, t)\rho)]^2 dt, \quad (14)$$

$$C^{\text{mc,ni}} = \int_V f(t) \text{Re}[\text{Tr}(G(A, t)\rho)]^2 dt - \text{Tr}(F(A)\rho)^2. \quad (15)$$

Here, the term $C^{\text{mc,ht}}/MK$ represents the variance contribution due to the Hadamard test and $C^{\text{mc,ni}}/K$ represents the variance contribution due to numerical integration. Consequently, the former scales as $\mathcal{O}(1/(MK))$, and the corresponding statistical error scales as $\mathcal{O}(1/M^{1/2}K^{1/2})$, while the latter scales as $\mathcal{O}(1/K)$, and the corresponding numerical integration error scales as $\mathcal{O}(1/K^{1/2})$.

C. LCU-CPP via the trapezoid rule

The trapezoid rule is another common approach to numerical integration and has been employed in several studies [14,16,18–20]. According to this method, one uses a set of deterministic grid points $\{t_k^{\text{tr}}\}$ over the domain V . Concretely, we take $K = \prod_{j=1}^d K_j$ grid points, where K_j is the number of integration points along the j th dimension. For each integration point t_k^{tr} , we perform the Hadamard test M times to obtain estimators v_k^{tr} satisfying $|v_k^{\text{tr}} - \text{Re}[\text{Tr}(G(A, t_k^{\text{tr}})\rho)]| < \mathcal{O}(1/\sqrt{M})$. Let ω_k be the trapezoid weight for each t_k^{tr} . Then, our final estimator is $v^{\text{tr}} = \prod_{j=1}^d (b_j - a_j)/(K_j - 1) \sum_{k=1}^K \omega_k f(t_k^{\text{tr}}) v_k^{\text{tr}}$, which approximates $\text{Tr}(F(A)\rho)$. In Appendix C, we show

$$\text{Ex}[v^{\text{tr}}] = \text{Tr}(F(A)\rho) + \text{Bias}^{\text{tr}}(K), \quad (16)$$

where $\text{Bias}^{\text{tr}}(K)$ arises from the numerical integration. Hence, unlike the Monte Carlo estimator, this method is biased, and we must consider the mean square error (MSE) rather than just the variance:

$$\begin{aligned} \text{MSE}^{\text{tr}} &= \text{Ex}[(v^{\text{tr}} - \text{Tr}(F(A)\rho))^2] \\ &= \text{Var}[v^{\text{tr}}] + \text{Bias}^{\text{tr}}(K)^2, \end{aligned} \quad (17)$$

where

$$\text{Var}[v^{\text{tr}}] \leq \frac{C^{\text{tr,ht}}}{MK} + \mathcal{O}\left(\frac{1}{MK^{\frac{2}{d}+1}}\right), \quad (18)$$

$$C^{\text{tr,ht}} = 2^d |V| \int_V f^2(t) (1 - \text{Re}[\text{Tr}(G(A, t)\rho)]^2) dt, \quad (19)$$

and

$$\begin{aligned} |\text{Bias}^{\text{tr}}(K)| &\leq \frac{C^{\text{tr,ni}}}{K^{\frac{2}{d}}} + \mathcal{O}\left(\frac{1}{K^{\frac{3}{d}}}\right), \quad (20) \\ C^{\text{tr,ni}} &= \sum_{j,j'=1}^d \frac{|V|}{3} \max_t \left| \frac{\partial^2}{\partial t_j \partial t_{j'}} f(t) \text{Re}[\text{Tr}(G(A, t)\rho)] \right| \\ &\quad \times (b_j - a_j)(b_{j'} - a_{j'}). \end{aligned} \quad (21)$$

Here, $\text{Var}[v^{\text{tr}}]$ represents the variance due to the Hadamard test and $\text{Bias}^{\text{tr}}(K)$ represents the numerical integration error. Consequently, the variance scales as $\mathcal{O}(1/MK)$, and the corresponding statistical error scales as $\mathcal{O}(1/M^{1/2}K^{1/2})$, while the numerical integration error scales as $\mathcal{O}(1/K^{2/d})$.

III. LCU-CPP VIA QUASI MONTE CARLO METHOD

In this section, we propose the LCU-CPP using QMC. We begin with an overview of QMC in classical numerical integration and then analyze the mean squared error of LCU-CPP when employing QMC. Finally, we highlight that QMC can possibly achieve faster convergence in integration error compared to MC, as well as a smaller constant coefficient than that associated with the trapezoid rule.

A. Quasi Monte Carlo method

First, we introduce QMC in classical numerical integration theory [22]. This method improves the asymptotic error behavior of MC by using low-discrepancy sequences, such as the Halton sequence, instead of random sampling. Let $h: \mathbb{R}^d \rightarrow \mathbb{R}$ be an integrand, and $p(\mathbf{x})$ a probability density function over a d -dimensional hyperrectangular domain V . We aim to approximate the integral

$$I = \int_V p(\mathbf{x}) h(\mathbf{x}) d\mathbf{x} \quad (22)$$

by the QMC estimator

$$Q(K) = \frac{1}{K} \sum_{k=1}^K h(\mathbf{u}_k), \quad (23)$$

where each $\mathbf{u}_k = \text{CDF}_p^{-1}(\mathbf{x}_k)$ is generated by applying the inverse cumulative distribution function (CDF) of $p(\mathbf{x})$ to a low-discrepancy sequence $\{\mathbf{x}_k\}$ such as the Halton sequence on $[0, 1]^d$. Here, CDF_p^{-1} denotes the inverse of the multivariate CDF of p , which can be constructed componentwise if p is a product distribution (i.e., if the variables are independent). This procedure is analogous to inversion sampling in MC. The Koksma-Hlawka inequality [22] gives the following bound:

$$|I - Q(K)| \leq V_{\text{HK}}(h) \times D^*(\{\mathbf{u}_k\}, K), \quad (24)$$

where $V_{\text{HK}}(h)$ is the variation of h in the sense of Hardy and Krause, and $D^*(\{\mathbf{u}_k\}, K)$ is the star discrepancy of the point set $\{\mathbf{u}_k\}$. More explicitly,

$$V_{\text{HK}}(h) = \sum_r \int_{V_r} \left| \frac{\partial^{|r|} h(\mathbf{x}_r, \mathbf{1})}{\partial \mathbf{x}_r^{|r|}} \right| d\mathbf{x}_r, \quad (25)$$

$$D^*(\{\mathbf{u}_k\}, K) = \sup_{J' \subset V} \left| \frac{1}{K} \#\{i \leq K : \mathbf{u}_i \in J'\} - \text{Vol}(J') \right|, \quad (26)$$

where r runs over all nonempty subsets of $\{1, \dots, d\}$, and $V_r = \{(\mathbf{x}_1, \dots, \mathbf{x}_d) \in V \mid x_j = 1 \text{ for all } j \notin r\}$. In $h(\mathbf{x}_r, \mathbf{1})$, the coordinates not in r are set to 1. The operator $\frac{\partial^{|r|}}{\partial \mathbf{x}_r^{|r|}}$ denotes the mixed partial derivative with respect to all variables in r . The supremum is over all axis-aligned subrectangles $J' \subset V$ of the form $J' = \text{CDF}_p^{-1}([0, \mathbf{t}])$ for $\mathbf{t} \in [0, 1]^d$, and $\text{Vol}(J')$ is the Lebesgue measure of J' . Theorem 2.35 in Ref. [22] shows that the star discrepancy of the Halton sequence satisfies $D^*(\{\mathbf{x}_k\}, K) = \mathcal{O}(\ln^d K/K)$. In this paper, we assume that

$D^*({\mathbf{u}}_k, K) = \mathcal{O}(\ln^d K/K)$ also holds for the transformed sequence $\{{\mathbf{u}}_k\}$. Although this does not hold for arbitrary p , this can be numerically verified for each case before performing integration for $h({\mathbf{x}})$. Since $V_{\text{HK}}(h)$ depends only on the integrand, the total integration error satisfies

$$|I - Q(K)| = \mathcal{O}\left(\frac{\ln^d K}{K}\right), \quad (27)$$

which is asymptotically faster than the MC rate. As a remark, more tighter error bounds could be obtained by adopting the Reproducing Kernel Hilbert Space framework or by considering quasiuniform sequences other than the Halton sequence. Nevertheless, we employ the simplest error estimate in this work, which does not affect the generality of the arguments that follow.

B. LCU-CPP via quasi Monte Carlo method

We now describe the LCU-CPP framework using QMC. For LCU-CPP via QMC, the mean square error to an estimator represents the total error. We show that this MSE consists of a variance arising from the statistical error and a bias term arising from the numerical integration error. At first, the estimator is defined as

$$v^{\text{qmc}} = \frac{1}{K} \sum_{k=1}^K v_k^{\text{qmc}}, \quad (28)$$

where

$$v_k^{\text{qmc}} = \frac{1}{M} \sum_{m=1}^M s_m({\mathbf{t}}_k^{\text{qmc}}). \quad (29)$$

Here, $s_m({\mathbf{t}}_k^{\text{qmc}}) \in \{+1, -1\}$ is the outcome of the m th Hadamard test, which is used to estimate $\text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)]$. A low-discrepancy sequence $\{{\mathbf{t}}_k^{\text{qmc}}\}$ is generated by ${\mathbf{t}}_k^{\text{qmc}} = \text{CDF}_p^{-1}({\mathbf{x}}_k)$, where $\text{CDF}_p({\mathbf{x}}) = \int_{-\infty}^{{\mathbf{x}}} p({\mathbf{x}}') d{\mathbf{x}}'$ and $\{{\mathbf{x}}_k\}$ is the Halton sequence in $[0, 1]^d$.

Next, we derive the mean square error of v^{qmc} with respect to $\text{Tr}[F(A)\rho]$. First, note that for each fixed ${\mathbf{t}}_k^{\text{qmc}}$, it holds that $\text{Ex}[s_m({\mathbf{t}}_k^{\text{qmc}})] = \text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)]$. Therefore,

$$\text{Ex}[v_k^{\text{qmc}}] = \frac{1}{M} \sum_{m=1}^M \text{Ex}[s_m({\mathbf{t}}_k^{\text{qmc}})] \quad (30)$$

$$= \text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)], \quad (31)$$

and so

$$\text{Ex}[v^{\text{qmc}}] = \frac{1}{K} \sum_{k=1}^K \text{Ex}[v_k^{\text{qmc}}] \quad (32)$$

$$= \frac{1}{K} \sum_{k=1}^K \text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)]. \quad (33)$$

Let us define the numerical integration error as $\text{Bias}^{\text{qmc}}(K) := \frac{1}{K} \sum_{k=1}^K \text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)] - \int_V f({\mathbf{t}}) \text{Re}[\text{Tr}(G(A, {\mathbf{t}})\rho)] d{\mathbf{t}} = \text{Ex}[v^{\text{qmc}}] - \text{Tr}[F(A)\rho]$, so that

$$\text{Ex}[v^{\text{qmc}}] = \text{Tr}[F(A)\rho] + \text{Bias}^{\text{qmc}}(K). \quad (34)$$

Thus, as with the trapezoid rule, this estimator is biased, and we consider the mean square error rather than just the variance. According to the Koksma-Hlawka inequality, we have $|\text{Bias}^{\text{qmc}}(K)| \leq C^{\text{qmc,ni}} \frac{(\ln K)^d}{K}$, where $C^{\text{qmc,ni}}$ is a constant involving $V_{\text{HK}}(\text{Re}[\text{Tr}(G(A, {\mathbf{t}})\rho)])$.

Now, let us consider the variance of v^{qmc} . Since $\text{Var}[s_m({\mathbf{t}}_k^{\text{qmc}})] = \text{Ex}[s_m({\mathbf{t}}_k^{\text{qmc}})^2] - (\text{Ex}[s_m({\mathbf{t}}_k^{\text{qmc}})])^2 = 1 - \text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)]^2$, and using $\int_V f({\mathbf{t}})(1 - \text{Re}[\text{Tr}(G(A, {\mathbf{t}})\rho)]^2) d{\mathbf{t}} = \frac{1}{K} \sum_{k=1}^K (1 - \text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)]^2) + \mathcal{O}((\ln K)^d/K)$, we find

$$\text{Var}[v^{\text{qmc}}] = \text{Var}\left[\frac{1}{K} \sum_{k=1}^K v_k^{\text{qmc}}\right] \quad (35)$$

$$= \frac{1}{K^2 M^2} \sum_{k=1}^K \sum_{m=1}^M \text{Var}[s_m({\mathbf{t}}_k^{\text{qmc}})] \quad (36)$$

$$= \frac{1}{K^2 M} \sum_{k=1}^K (1 - \text{Re}[\text{Tr}(G(A, {\mathbf{t}}_k^{\text{qmc}})\rho)]^2) \quad (37)$$

$$= \frac{1}{KM} \int_V f({\mathbf{t}})(1 - \text{Re}[\text{Tr}(G(A, {\mathbf{t}})\rho)]^2) d{\mathbf{t}} + \mathcal{O}\left(\frac{(\ln K)^d}{K^2 M}\right). \quad (38)$$

Combining this variance with the bias, the MSE as the total error in LCU-CPP via QMC is given by

$$\text{MSE}^{\text{qmc}} = \text{Ex}[(v^{\text{qmc}} - \text{Tr}[F(A)\rho])^2] \quad (39)$$

$$= \text{Var}[v^{\text{qmc}}] + \text{Bias}^{\text{qmc}}(K)^2. \quad (40)$$

Here, $\text{Var}[v^{\text{qmc}}]$ represents the variance due to the Hadamard test, while $\text{Bias}^{\text{qmc}}(K)$ represents the numerical integration error. Consequently, the variance scales as $\mathcal{O}(1/MK)$, and the corresponding statistical error scales as $\mathcal{O}(1/M^{1/2}K^{1/2})$, while the numerical integration error scales as $\mathcal{O}((\ln K)^d/K)$.

C. Advantages

Let us discuss the advantages of QMC. The convergence rates for each method are summarized as follows:

(1) Monte Carlo (see Sec. II B):

Statistical error: $\mathcal{O}(1/M^{1/2}K^{1/2})$.

Numerical integration error: $\mathcal{O}(1/K^{1/2})$.

(2) Trapezoid Rule (see Sec. II C):

Statistical error: $\mathcal{O}(1/M^{1/2}K^{1/2})$.

Numerical integration error: $\mathcal{O}(1/K^{2/d})$.

(3) Quasi Monte Carlo (see Sec. III B):

Statistical error: $\mathcal{O}(1/M^{1/2}K^{1/2})$.

Numerical integration error: $\mathcal{O}((\ln K)^d/K)$.

Compared to MC, QMC achieves a numerical integration error of $\mathcal{O}((\ln K)^d/K)$, which converges asymptotically faster than MC, although both methods have the same rate for the statistical error. Therefore, when the number of Hadamard test shots M is large and the statistical error is ignorable, the quasi Monte Carlo method can outperform the Monte Carlo method. Comparing with the trapezoid rule, the trapezoid rule achieves a numerical integration error of $\mathcal{O}(1/K^{2/d})$, which is almost faster than or the same as QMC, especially for $d = 2$ as in many LCU-CPP applications. However, while the

statistical errors for both QMC and the trapezoid rule converge as $\mathcal{O}(1/M^{1/2}K^{1/2})$, the constant factor for the trapezoid rule depends on the integration domain $|V|$, as shown in Eq. (19). This dependency can be particularly disadvantageous when integrating over large domains, which is often encountered in LCU-CPP applications. In contrast, QMC can avoid this disadvantage by using importance sampling, which makes it independent of the integration domain.

As a result, this leads to the following scenarios:

- (1) When $M = 1$: The statistical error dominates. MC and QMC achieve the lowest error.
- (2) When M is moderate (e.g., 10^2 or 10^3): The statistical error and numerical integration error are balanced. QMC achieves the lowest error.
- (3) When $M \rightarrow \infty$: The numerical integration error dominates. the trapezoidal rule achieves the lowest error.

In many previous studies, the number of Hadamard test shots is fixed to $M = 1$. In realistic experimental setups, however, M should be regarded as a tunable parameter. Equivalently, the relevant resource is the total number of shots MK . This viewpoint is natural when taking into account experimental constraints such as the latency associated with quantum circuit execution and information transmission [23,24], under which repeated sampling is often less costly than circuit preparation. Within this framework, QMC is expected to become advantageous when M takes a moderate value. At this point, the question then is what constitutes an appropriate choice of M . Since both the statistical error and the numerical integration error depend on the integrand, it is necessary to consider concrete applications. While the range of admissible values of M is determined by the hardware experimental setup, it is nonetheless useful to clarify the values of M for which QMC becomes practically advantageous. In the next section, we perform numerical experiments on two representative applications, which substantiate this scenario and provide explicit examples of values of M for which QMC is practically favorable within the framework.

It should also be noted that the numerical integration error of the trapezoid rule can be significantly reduced by using extended Newton-Cotes formulas, such as Simpson's rule [27]. These methods only require modifying the weights in the trapezoid rule and can also be applied within the LCU-CPP framework, with a similar error analysis. However, while Newton-Cotes rules can provide theoretical improvements in accuracy, they tend to become increasingly numerically unstable for higher-order formulas. In our experiments, we also include results for Simpson's rule as a reference, as it is the most well known of the Newton-Cotes formulas.

IV. NUMERICAL EXPERIMENT

In this section, we conduct two numerical experiments, estimating ground-state properties and the Green's function, to demonstrate that the quasi Monte Carlo method yields the lowest error in LCU-CPP applications. Our results show that the QMC achieves the smallest errors, owing to its lower coefficients for numerical integration error compared to the trapezoid and Simpson's rules, and its faster convergence relative to the MC in moderate values M . All source code and data are publicly available in the Zenodo repository [28].

A. Case 1: Ground-state properties

We first present a numerical experiment aimed at estimating properties of the ground state in a quantum many-body system [12,14,19]. The key idea is to apply a filter function $e^{-\tau^2 H^2}$ to a quantum state. Since this filter suppresses higher-energy state more rapidly than the ground state, the ground state is selectively amplified. Here, H denotes the positive-definite Hamiltonian of the system, and τ is a tunable parameter. More precisely, let ρ_0 be the initial quantum state and let ρ_g denote the ground state of H . The expectation value of an observable O with respect to the ground state, $\text{Tr}[O\rho_g]$, can be approximated as $\text{Tr}[e^{-\tau^2 H^2} O e^{-\tau^2 H^2} \rho_0] / \text{Tr}[e^{-2\tau^2 H^2} \rho_0]$. In order to estimate the numerator and denominator, respectively, using the LCU-CPP, we use a Fourier representation of the Gaussian filter: $\exp(-\tau^2 H^2) = \int_{-\infty}^{\infty} 1/\sqrt{4\pi\tau^2} \exp(-t^2/4\tau^2) \exp(-iHt) dt$. Then, the numerator that we estimate here is expressed in the format of Eq. (1) as

$$F(H) = e^{-\tau^2 H^2} O e^{-\tau^2 H^2}, \quad (41)$$

$$f(t) = \frac{1}{4\pi\tau^2} e^{-\frac{t^2 + t'^2}{4\tau^2}}, \quad (42)$$

$$G(H, t) = e^{-iHt} O e^{-iHt'}. \quad (43)$$

For our experiment, we adopt the one-dimensional Heisenberg model as the system Hamiltonian and employ a two-site correlator as the observable:

$$H = J \sum_{j=0}^{N-1} (X_j X_{j+1} + Y_j Y_{j+1} + Z_j Z_{j+1}) + E_0 I, \quad (44)$$

$$\rho = \rho_0 = \frac{1}{2^N} \sum_{x=0}^{2^N-1} |x\rangle \langle x|, \quad (45)$$

$$O = Z_0 Z_1, \quad (46)$$

where periodic boundary conditions are assumed: Site N is identified with site 0. The parameters used are $N = 6$, $J = 1$, and $\tau = 8$. The offset E_0 is set to the ground-state energy, which is computed in advance. For numerical integration using the trapezoid and Simpson's rules, we introduce a cutoff region $[-t_c, t_c] \times [-t'_c, t'_c]$, where the cutoff t_c is chosen such that the absolute error is set to the double-precision floating-point accuracy, and we use $t_c = t'_c = 100$. In the simulation of the Hadamard test, we do not consider the error from the Trotter decomposition or other hardware errors.

First, Fig. 2 shows the results of numerical experiments focused solely on the numerical integration error. The error rate is defined as the absolute difference between the estimated value and the desired value, divided by the desired value. These results do not include statistical error, corresponding to the case where the number of Hadamard test measurements M is infinite. The vertical axis indicates the average error rate of the 100 estimation trials, and the horizontal axis represents the number of variable samples K , or in other words, the number of different Hadamard test circuits K . Both the trapezoid and Simpson's rules exhibit a sharp drop in the error rate as the sample number K increases, and eventually output highly accurate results. It has been pointed out that this exponential convergence occurs when the integrand is an oscillatory and

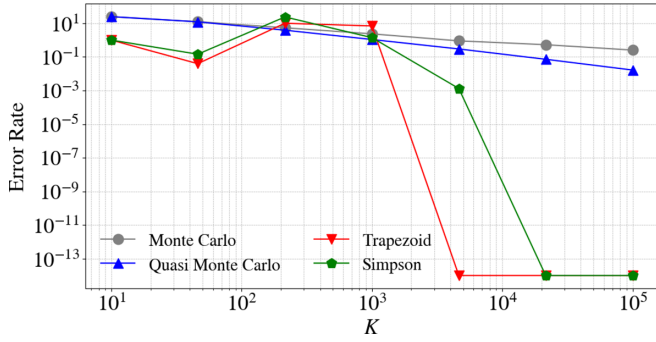


FIG. 2. Error rate vs the number of the different Hadamard tests K for each integration method without statistical error for estimating the ground-state properties. This corresponds to the limit case where $M \rightarrow \infty$.

decaying function. Since the Gaussian type discussed here has exactly such an integrand form, we can expect this phenomenon to be widely observed in applications involving the Fourier transform of Gaussian-type functions. Additionally, the starting point of the convergence on the trapezoid rule is mostly the case where the integration width becomes smaller than the oscillation period, which is called as Nyquist limit [29]. Both MC and QMC converge steadily as K increases, but QMC achieves a lower error rate more rapidly.

Next, we present numerical integration results that include statistical errors, as shown in Fig. 3. Panels (a) and (b) correspond to cases with $M = 1$ and $M = 1000$, respectively. The case of $M = 1$ is examined to observe the balance between

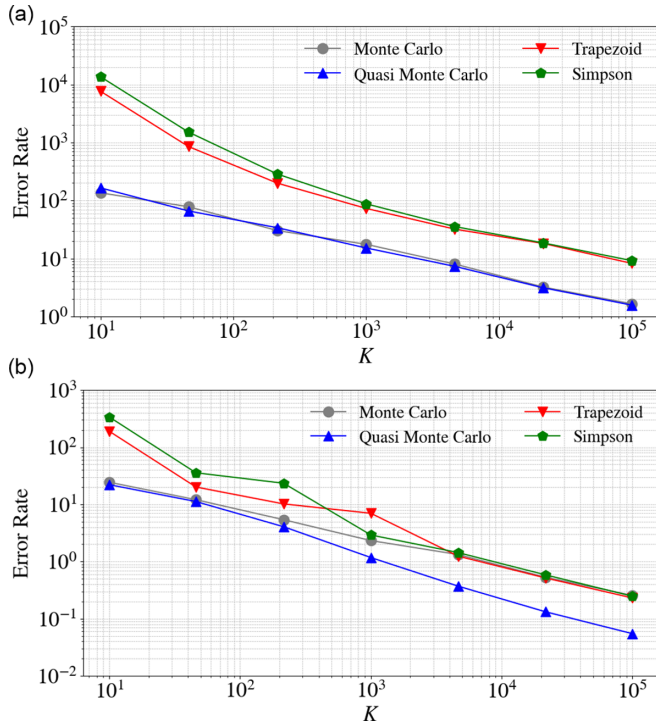


FIG. 3. Error rate vs the number of the different Hadamard tests K for each integration method including statistical error for estimating the ground-state properties: (a) $M = 1$ and (b) $M = 1000$. M is the number of shots of each Hadamard test.

statistical and numerical integration errors, while $M = 1000$ serves as a representative example of a natural hardware setup. In panel (a) ($M = 1$), all estimation points show higher error rates compared to those in Fig. 2, indicating that statistical errors are dominant. Here, MC and QMC exhibit lower errors, whereas the trapezoid and Simpson's rules display even larger errors. Panel (b) presents the results for $M = 1000$, where statistical errors become relatively small, allowing the numerical integration error to be observed. However, when K increases, the overall error eventually should become dominated by statistical error again except for MC, because the numerical errors converge faster and are asymptotically getting lower than the statistical errors. In the end, QMC achieves the lowest overall in this setup.

B. Case 2: Green's function

We next conduct a numerical experiment for estimating Green's functions [17,20]. Since the Green's function involves the inverse of the Hamiltonian, its evaluation benefits from employing a Fourier-type integral representation of the inverse matrix. This representation is not only applicable to Green's functions but also extendable to more general inverse matrix computations, making this numerical experiment broadly significant. More specifically, to approximate H^{-1} , we use the following truncated Fourier representation for an invertible Hermitian H :

$$H^{-1} = -\frac{i}{\sqrt{2\pi}} \int_0^{y_c} dy \int_{-z_c}^{z_c} dz z e^{-z^2/2} e^{-iyzH}. \quad (47)$$

This equation holds as $y_c, z_c \rightarrow \infty$. Here, the eigenvalues of H are assumed to lie within $[-a, -1/b] \cup [1/b, a]$, with $y_c = \Theta(b\sqrt{\ln(b/\varepsilon)})$ and $z_c = \Theta(\sqrt{\ln(b/\varepsilon)})$ ensuring an ε -accurate truncation. In our numerical experiments, we estimate the Green's function

$$G_{kl}^{(+)}(\omega) := \langle E_0 | \hat{a}_k(\omega - (H - E_0) + i\eta x)^{-1} \hat{a}_l^\dagger | E_0 \rangle, \quad (48)$$

as discussed in Ref. [17]. The Hamiltonian is the Heisenberg model of Eq. (44), and the ground state $|E_0\rangle$ is computed analytically in advance. We set the density matrix $\rho = |E_0\rangle\langle E_0|$. For spins, $\hat{a}_k = (X_k + iY_k)/2$ and $\hat{a}_l^\dagger = (X_l - iY_l)/2$, but here we replace $\hat{a}_k, \hat{a}_l^\dagger$ by X_k, X_l to isolate integration-scheme effects. This does not affect our comparative conclusions. As typical parameters, we take $k = l = 0, \omega = E_0$, and $\eta = 0.01$. Hence, we estimate $\text{Tr}[X_0 (2E_0 + 0.01i - H)^{-1} X_0 |E_0\rangle\langle E_0|]$. In the form of Eq. (2), with $V = [0, y_c] \times [-z_c, z_c]$ and $Z(z_c) := 1 - e^{-z_c^2/2}$, this is expressed as

$$F(H) = X_0 (2E_0 + 0.01i - H)^{-1} X_0, \quad (49)$$

$$f(t) = \frac{1}{2y_c Z(z_c)} |z| e^{-\frac{z^2}{2}}, \quad (50)$$

$$G(H, t) = -\frac{2iZ(z_c)}{\sqrt{2\pi}} \text{sign}(z) y_c X_0 \times e^{-iyz(2E_0 + 0.01i - H)} X_0, \quad (51)$$

for $t = (y, z)$. Since G is the imaginary unit, the Hadamard test uses a circuit that extracts the imaginary value. We set $y_c = z_c = 8$, which was sufficient to make the truncation error negligible at double precision in our experiments.

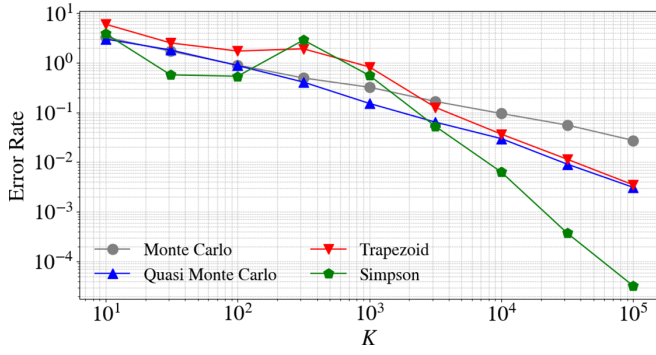


FIG. 4. Error rate vs the number of the different Hadamard tests K for each integration method without statistical error for estimating the Green's function. This corresponds to the limit case where $M \rightarrow \infty$.

First, Fig. 4 presents the results of numerical experiments focusing solely on numerical integration error. This situation corresponds to the case where M is infinite. In the trapezoid and Simpson's rules, the error rates decrease as K increases; however, unlike the previous section, no exponential decay is observed. This is because exponential decay occurs only when the integrand decays in each dimension [29], and whereas the function $f(t)$ in Eq. (50) does not decay with respect to y . Also, Simpson's rule becomes advantageous as K grows. The converging begins from $K \geq 10^3$, which means the Nyquist limit blocks the convergence until $K < 10^3$. Both MC and QMC steadily converge with increasing K , but QMC achieves a lower error rate more rapidly.

Next, we present the results that also include statistical error, as shown in Fig. 5. Panel (a) of Fig. 5 ($M = 1$) shows that the error rates at all estimation points are higher compared to Fig. 4. Although the difference in error rates is not as significant as in the previous section, statistical error remains the dominant contribution. Panel (b) of Fig. 3 displays the results for $M = 100$, where statistical error becomes relatively small and the impact of numerical integration error becomes apparent. However, as in the previous section, increasing K ultimately leads to overall error being dominated by the statistical error once again. Consequently, QMC achieves the lowest error rate for large values of K . As a result, QMC achieves the lowest overall error in this setup as well.

C. Practical considerations

So far, we have seen that in the case of QMC, the error remains the smallest for $M \leq 1000$ in case 1 and for $M \leq 100$ in case 2. These applications employ Gaussian and inverse matrix Fourier transforms, which are commonly used in previous studies. Therefore, similar discussions can be expected in other applications that utilize the same types of Fourier transforms. Then, based on the results of our numerical experiments, we would like to further discuss the practical advantages of QMC from a more applied perspective. In the case of the trapezoid rule, the Nyquist condition should be satisfied for the method to start converging. However, in practice, it is often difficult to know the period of the integrand in advance. In these applications, the shortest period of the integrand depends on the largest eigenvalue of the Hamiltonian

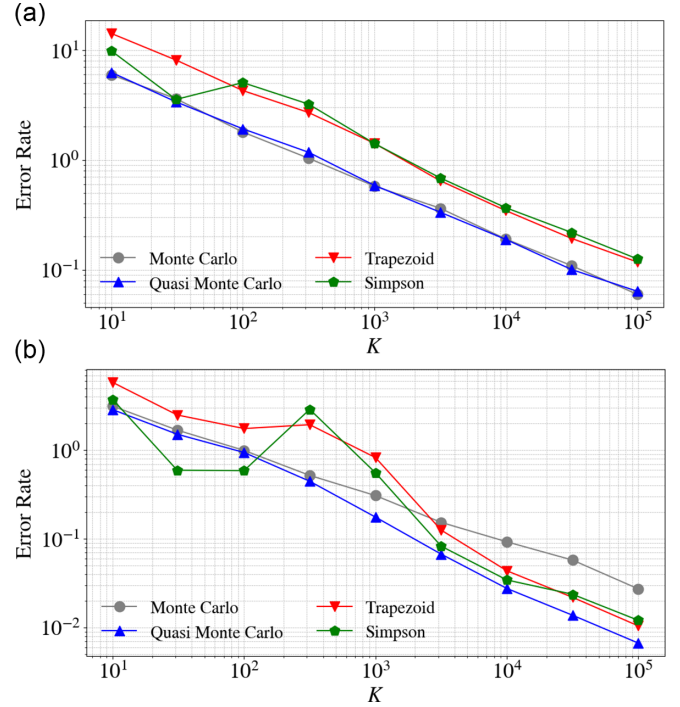


FIG. 5. Error rate vs the number of the different Hadamard tests K for each integration method including statistical error for estimating the Green's function: (a) $M = 1$ and (b) $M = 100$. M is the number of shots of each Hadamard test.

H . To satisfy the Nyquist condition, the integration width in each dimension should be as small as $|V|/K^{1/d}$ when dividing the domain evenly, which means that as the largest eigenvalue increases, it becomes more difficult for the trapezoid rule to achieve convergence. On the other hand, QMC converges at least as fast as MC, or even faster, and its convergence does not depend on K in the same way. This allows us to obtain the estimate even for small values of K , and it is easy to stop the calculation early if necessary. For these reasons, QMC can be used as a practical alternative to MC, offering greater flexibility and reliability in real setups.

We note that in Sec. II A we discussed previous studies on applications with dimensionality up to $d = 4$. The arguments presented there are expected to become even more relevant for larger values of d . However, applications requiring substantially higher dimensional integration remain unclear at present, and identifying such applications and assessing their suitability is left for future work.

V. CONCLUSION

In this study, we proposed QMC for LCU-CPP. The LCU-CPP framework can realize a wide range of nonunitary operators, such as Gibbs states and inverse matrix, through integral representations, while requiring relatively few quantum hardware resources. We first analyzed the errors of the trapezoid rule, MC, and QMC. We showed that the errors consist of numerical integration error and statistical error, and we derived upper bounds for these two contributions. From this analysis, we argued that when the number of Hadamard test shots per unitary M is in a moderate range, QMC has the

potential to achieve the lowest error. Since this conclusion was based only on upper bounds, and since the definition of the moderate range of M was unclear, we performed numerical experiments. Specifically, we studied ground-state property estimation and Green's function estimation. We found that QMC indeed achieves the lowest error around $M = 10^3$ for the former and around $M = 10^2$ for the latter. These values of M are realistic for actual quantum devices, especially when circuit execution and communication latencies are taken into account. Overall, our results demonstrate that QMC can serve as the most effective integration strategy within the LCU-CPP framework in practical experimental settings.

ACKNOWLEDGMENTS

This work was supported by the MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant No. JP-MXS0120319794 and the JST COI-NEXT Program Grant No. JPMJPF2014.

DATA AVAILABILITY

The data that support the findings of this article are openly available [28].

APPENDIX A: HADAMARD TEST

We define a probability distribution $F_x(s)$ parametrized by x , where $-1 \leq x \leq 1$. The probability distribution is given by

$$F_x(s) = \begin{cases} \frac{1-x}{2}, & \text{for } s = -1, \\ \frac{1+x}{2}, & \text{for } s = +1. \end{cases} \quad (\text{A1})$$

The expected value and variance of s under the distribution $F_x(s)$ are as follows:

$$\text{Ex}[s] = x, \quad (\text{A2})$$

$$\text{Var}[s] = (1+x)(1-x), \quad (\text{A3})$$

where $\text{Ex}[s]$ denotes the expected value of s and $\text{Var}[s]$ denotes the variance of s . Considering ρ as a quantum state and U as a unitary matrix, we estimate $\text{Re}[U\rho]$. An estimator ν is defined by

$$\nu = \frac{1}{K} \sum_{k=1}^K s_k, \quad (\text{A4})$$

where s_k is the k th outcome from the Hadamard test as depicted in Fig. 6, and it follows $F_x(s_k)$. The expected value and variance of ν are as follows:

$$\text{Ex}[\nu] = \frac{1}{K} \sum_{k=1}^K \text{Ex}[s_k] = \text{Re}[\text{Tr}(U\rho)], \quad (\text{A5})$$

$$\text{Var}[\nu] = \frac{1}{K} \sum_{k=1}^K \text{Var}[s_k] = \frac{1}{K} (1 - \text{Re}[\text{Tr}(U\rho)]^2). \quad (\text{A6})$$

The square root of this variance can be interpreted as a statistical error.

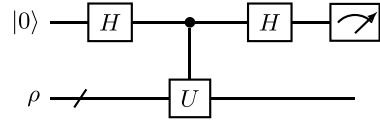


FIG. 6. Quantum circuit for the Hadamard test. The outcome $s \in \{+1, -1\}$ satisfies $\text{Pr}[s = +1] = (1+x)/2$ and $\text{Pr}[s = -1] = (1-x)/2$, where $x = \text{Re}[\text{Tr}(U\rho)]$.

APPENDIX B: ERROR ANALYSIS IN MONTE CARLO METHOD

The LCU-CPP approach relying on the Monte Carlo integration employs the following estimator:

$$\nu^{\text{mc}} = \frac{1}{K} \sum_{k=1}^K \nu_k^{\text{mc}}, \quad (\text{B1})$$

where

$$\nu_k^{\text{mc}} = \frac{1}{M} \sum_{m=1}^M s_m(t_k). \quad (\text{B2})$$

Here, $s_m(t_k) \in \{1, -1\}$ is the output of the m th execution of the Hadamard test for $\text{Re}[G(A, t_k)]$, where $\{t_k\}$ represents mutually independent samples drawn from $f(t)$. We seek to compute the expected value and the variance of ν^{mc} .

First, we consider the expected value. The expected value of $s_m(t_k)$ over the randomness of the measurement conditioned on the choice of t_k is

$$\text{Ex}[s_m(t_k)] = \text{Re}[\text{Tr}(G(A, t_k)\rho)]. \quad (\text{B3})$$

We can then take the expectation over the randomness of t_k for Eq. (B3) to conclude $\text{Ex}[s_m(t_k)] = \text{Tr}[F(A)\rho]$ and thus $\text{Ex}[\nu_k^{\text{mc}}] = \text{Ex}[\nu^{\text{mc}}] = \text{Tr}[F(A)\rho]$.

Next, we look at the variance. Since ν_k^{mc} are mutually independent and identically distributed, we have $\text{Var}[\nu^{\text{mc}}] = \text{Var}[\nu_k^{\text{mc}}]/K$. It therefore suffices to calculate the variance of single ν_k^{mc} to obtain $\text{Var}[\nu^{\text{mc}}]$. Let us define a random variable $\tilde{\nu}^{\text{mc}} := \frac{1}{M} \sum_{m=1}^M s_m(t)$, where t is sampled from $f(t)$. We make use of the law of total variance to compute $\tilde{\nu}^{\text{mc}}$, which in this case can be written as

$$\text{Var}[\tilde{\nu}^{\text{mc}}] = \text{Ex}_t[\text{Var}_{\text{meas}}(\tilde{\nu}^{\text{mc}}(t))] + \text{Var}_t[\text{Ex}_{\text{meas}}(\tilde{\nu}^{\text{mc}}(t))]. \quad (\text{B4})$$

$\text{Var}_{\text{meas}}[\tilde{\nu}^{\text{mc}}(t)]$ can be calculated as

$$\text{Var}_{\text{meas}}[\tilde{\nu}^{\text{mc}}(t)] = \frac{1}{M} \text{Var}_{\text{meas}}(s_m(t)), \quad (\text{B5})$$

since $s_m(t)$ represent independent execution of the Hadamard test and are thus independent random variables. $\text{Var}_{\text{meas}}[s_m(t)]$ can be calculated as

$$\begin{aligned} \text{Var}_{\text{meas}}[s_m(t)] &= \text{Ex}[s_m^2(t)] - \text{Ex}[s_m(t)]^2 = 1 \\ &\quad - \text{Re}[\text{Tr}(G(A, t)\rho)]^2. \end{aligned} \quad (\text{B6})$$

We have

$$\begin{aligned} \text{Ex}_t[\text{Var}_{\text{meas}}(\tilde{\nu}^{\text{mc}}(t))] &= \frac{1}{M} - \frac{1}{M} \int_V f(t) \\ &\quad \times \text{Re}[\text{Tr}(G(A, t)\rho)]^2 dt. \end{aligned} \quad (\text{B7})$$

Next, $\text{Ex}_{\text{meas}}(\tilde{v}^{\text{mc}}(\mathbf{t}))$ can be calculated as

$$\begin{aligned}\text{Ex}_{\text{meas}}(\tilde{v}^{\text{mc}}(\mathbf{t})) &= \frac{1}{M} \sum_{m=1}^M \text{Ex}_{\text{meas}}(s_m(\mathbf{t})) \\ &= \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)].\end{aligned}\quad (\text{B8})$$

Therefore,

$$\begin{aligned}\text{Var}_t[\text{Ex}_{\text{meas}}(\tilde{v}^{\text{mc}}(\mathbf{t}))] &= \text{Ex}_t[\text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2] \\ &\quad - \text{Ex}_t[\text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]]^2\end{aligned}\quad (\text{B9})$$

$$= \int_V f(\mathbf{t}) \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2 d\mathbf{t} - \text{Tr}(F(A)\rho)^2. \quad (\text{B10})$$

From Eqs. (B4), (B7), and (B10), we obtain

$$\begin{aligned}\text{Var}[\tilde{v}^{\text{mc}}] &= \frac{1}{M} - \frac{1}{M} \int_V f(\mathbf{t}) \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2 d\mathbf{t} \\ &\quad + \int_V f(\mathbf{t}) \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2 d\mathbf{t} - \text{Tr}(F(A)\rho)^2.\end{aligned}\quad (\text{B11})$$

Finally, using that $\text{Var}[v^{\text{mc}}] = \text{Var}[\tilde{v}^{\text{mc}}]/K$,

$$\begin{aligned}\text{Var}[v^{\text{mc}}] &= \frac{1}{K} \left(\int_V f(\mathbf{t}) \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2 d\mathbf{t} - \text{Tr}(F(A)\rho)^2 \right) \\ &\quad + \frac{1}{KM} \left(1 - \int_V f(\mathbf{t}) \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2 d\mathbf{t} \right).\end{aligned}\quad (\text{B12})$$

APPENDIX C: ERROR ANALYSIS IN TRAPEZOID RULE

First, we briefly review classical numerical integration error in the trapezoid rule. Let $h: \mathbb{R}^d \rightarrow \mathbb{R}$ be a function with continuous second derivatives on a hyperrectangle $V = \prod_{j=1}^d [a_j, b_j]$. We divide each axis into N intervals of equal width $\Delta_j = (b_j - a_j)/(N - 1)$, yielding $K = N^d$ grid points $\mathbf{x}_i$. The trapezoid rule approximates the integral $I = \int_V h(\mathbf{x}) d\mathbf{x}$ as

$$T(K) = \left(\prod_{j=1}^d \Delta_j \right) \sum_{i=1}^K h(\mathbf{x}_i). \quad (\text{C1})$$

The error $E(K) := |I - T(K)|$ can be analyzed by Taylor expanding h to second order around each grid point and integrating over each cell. This yields a local error per cell of

$$E_{\text{cell}} = \sum_{j,j'=1}^d \frac{\partial^2 h}{\partial x_j \partial x_{j'}}(\mathbf{x}_i) \frac{\Delta_j \Delta_{j'}}{12} \prod_{m=1}^d \Delta_m + \mathcal{O}(\|\Delta\|^4). \quad (\text{C2})$$

Summing over all K cells gives

$$E(K) \leq \sum_{j,j'=1}^d \frac{|V|}{12} \max_{\mathbf{x} \in V} \left| \frac{\partial^2 h}{\partial x_j \partial x_{j'}} \right| \Delta_j \Delta_{j'} + \mathcal{O}(\|\Delta\|^3), \quad (\text{C3})$$

where $|V| = \prod_{j=1}^d (b_j - a_j)$. Using $\Delta_j \leq 2(b_j - a_j)/K^{1/d}$, we have

$$E(K) \leq \frac{C}{K^{2/d}} + \mathcal{O}\left(\frac{1}{K^{3/d}}\right), \quad (\text{C4})$$

where

$$C = \sum_{j,j'=1}^d \frac{|V|}{3} (b_j - a_j)(b_{j'} - a_{j'}) \max_{\mathbf{x} \in V} \left| \frac{\partial^2 h}{\partial x_j \partial x_{j'}} \right|, \quad (\text{C5})$$

which depends on the domain size and the maximum of the second derivatives of h . Thus, the trapezoid rule converges as $\mathcal{O}(K^{-2/d})$.

Next, the LCU-CPP approach relying on the trapezoid rule employs the following estimator:

$$v^{\text{tr}} = \prod_{j=1}^d \Delta_j \sum_{k=1}^K \omega_k f(\mathbf{t}_k^{\text{tr}}) v_k^{\text{tr}}, \quad (\text{C6})$$

where

$$v_k^{\text{tr}} = \frac{1}{M} \sum_{m=1}^M s_m(\mathbf{t}_k^{\text{tr}}). \quad (\text{C7})$$

Here, $\Delta_j = (b_j - a_j)/(K_j - 1)$, where K_j is the number of the integration points in the j th dimension. ω_k are the trapezoid rule weights for $\mathbf{t}_k^{\text{tr}}$, and $\mathbf{t}_k^{\text{tr}}$ is the k th integration point. $s_m(\mathbf{t}_k^{\text{tr}}) \in \{1, -1\}$ is the output of the m th execution of the Hadamard test for $G(A, \mathbf{t}_k^{\text{tr}})$, where $\{\mathbf{t}_k^{\text{tr}}\}$ represents evenly spaced grid points over V .

We then show the expected value. The expected value of $s_m(\mathbf{t}_k^{\text{tr}})$ over the deterministic integral point $\mathbf{t}_k$ is

$$\text{Ex}[s_m(\mathbf{t}_k^{\text{tr}})] = \text{Re}[\text{Tr}(G(A, \mathbf{t}_k)\rho)]. \quad (\text{C8})$$

We obtain $\text{Ex}[v_k^{\text{tr}}] = \frac{1}{M} \sum_{m=1}^M \text{Ex}[s_m(\mathbf{t}_k^{\text{tr}})] = \text{Re}[\text{Tr}(G(A, \mathbf{t}_k)\rho)]$. That leads to $\text{Ex}[v^{\text{tr}}] = \prod_{j=1}^d \Delta_j \omega_k f(\mathbf{t}_k^{\text{tr}}) \text{Ex}[v_k^{\text{tr}}] = \prod_{j=1}^d \Delta_j \omega_k f(\mathbf{t}_k^{\text{tr}}) \text{Re}[\text{Tr}(G(A, \mathbf{t}_k)\rho)]$, which is corresponding the trapezoid rule formula for estimating $\text{Re}[\text{Tr}(G(A, \mathbf{t}_k)\rho)]$. Therefore, the expected value of v_k^{tr} is biased by the numerical integration error of the trapezoid rule. Since the numerical integration error gives $\text{Bias}^{\text{tr}}(K) := \text{Tr}[F(A)\rho] - \prod_{j=1}^d \Delta_j \sum_{k=1}^K \omega_k f(\mathbf{t}_k^{\text{tr}}) \text{Re}[\text{Tr}(G(A, \mathbf{t}_k)\rho)] = \text{Tr}[F(A)\rho] - \text{Ex}[v^{\text{tr}}]$, the desired value $\text{Tr}[F(A)\rho]$ can be written as

$$\text{Tr}[F(A)\rho] = \text{Ex}[v^{\text{tr}}] + \text{Bias}^{\text{tr}}(K). \quad (\text{C9})$$

Therefore, we find that the estimator is biased by the numerical integration error. Next, we examine the variance of the

estimator v^{tr} . The variance is

$$\text{Var}[v^{\text{tr}}] = \left(\prod_{j=1}^d \Delta_j \right)^2 \sum_{k=1}^K \omega_k^2 f^2(\mathbf{t}_k^{\text{tr}}) \text{Var}[v_k^{\text{tr}}] = \left(\prod_{j=1}^d \Delta_j \right)^2 \sum_{k=1}^K \omega_k^2 f^2(\mathbf{t}_k^{\text{tr}}) \frac{1}{M^2} \sum_{m=1}^M \text{Var}[s_m(\mathbf{t}_k^{\text{tr}})]. \quad (\text{C10})$$

Since $\text{Var}[s_m(\mathbf{t}_k^{\text{tr}})] = \text{Ex}[s_m(\mathbf{t}_k^{\text{tr}})^2] - \text{Ex}[s_m(\mathbf{t}_k^{\text{tr}})]^2 = 1 - \text{Re}[\text{Tr}(G(A, \mathbf{t}_k^{\text{tr}})\rho)]^2$, we can write

$$\text{Var}[v^{\text{tr}}] = \left(\prod_{j=1}^d \Delta_j \right)^2 \sum_{k=1}^K \omega_k^2 f^2(\mathbf{t}_k^{\text{tr}}) \frac{1}{M} (1 - \text{Re}[\text{Tr}(G(A, \mathbf{t}_k^{\text{tr}})\rho)]^2). \quad (\text{C11})$$

Then, according to the numerical integration error,

$$\text{Var}[v^{\text{tr}}] = \prod_{j=1}^d \Delta_j \frac{1}{M} \int_V f^2(\mathbf{t}) (1 - \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2) d\mathbf{t} + \mathcal{O}\left(\prod_{j=1}^d \Delta_j \frac{1}{MK^{\frac{2}{d}}}\right). \quad (\text{C12})$$

When all K_j are assumed to be the same value, we can obtain $\prod_{j=1}^d \Delta_j = \prod_{j=1}^d (b_j - a_j/K_j - 1) \leq 2^d \prod_{j=1}^d (b_j - a_j/K_j) = 2^d |V|/K$, and then we have

$$\text{Var}[v^{\text{tr}}] \leq \frac{1}{MK} 2^d |V| \int_V f^2(\mathbf{t}) (1 - \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]^2) d\mathbf{t} + \mathcal{O}\left(\frac{1}{MK^{\frac{2}{d}+1}}\right). \quad (\text{C13})$$

When the first term is dominant, we can see $\text{Var}[v^{\text{tr}}] = \mathcal{O}(1/MK)$. Considering the variance and the bias over the estimator, the mean squared error to $\text{Tr}[F(A)\rho]$ is

$$\text{MSE}^{\text{tr}} = \text{Ex}[(v^{\text{tr}} - \text{Tr}[F(A)\rho])^2] = \text{Var}[v^{\text{tr}}] + \text{Bias}^{\text{tr}}(K)^2, \quad (\text{C14})$$

where

$$\text{Bias}^{\text{tr}}(K) \leq \frac{1}{K^{2/d}} \sum_{j,j'=1}^d \frac{|V|(b_j - a_j)(b_{j'} - a_{j'})}{3} \max_{\mathbf{t} \in V} \left| \frac{\partial^2 f(\mathbf{t}) \text{Re}[\text{Tr}(G(A, \mathbf{t})\rho)]}{\partial t_j \partial t_{j'}} \right| + \mathcal{O}\left(\frac{1}{K^{3/d}}\right). \quad (\text{C15})$$

-
- [1] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, *Rev. Mod. Phys.* **92**, 015003 (2020).
- [2] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, *et al.*, Quantum chemistry in the age of quantum computing, *Chem. Rev.* **119**, 10856 (2019).
- [3] R. Orús, S. Mugel, and E. Lizaso, Quantum computing for finance: Overview and prospects, *Rev. Phys.* **4**, 100028 (2019).
- [4] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, Quantum computing for finance: State-of-the-art and future prospects, *IEEE Trans. Quantum Eng.* **1**, 1 (2020).
- [5] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, *arXiv:1411.4028*.
- [6] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [7] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, *Rev. Mod. Phys.* **86**, 153 (2014).
- [8] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear T complexity, *Phys. Rev. X* **8**, 041015 (2018).
- [9] 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).
- [10] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoefler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vashchillo, Assessing requirements to scale to practical quantum advantage, *arXiv:2211.07629*.
- [11] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [12] P. Zeng, J. Sun, and X. Yuan, Universal quantum algorithmic cooling on a quantum computer, *arXiv:2109.15304*.
- [13] M. Huo and Y. Li, Error-resilient Monte Carlo quantum simulation of imaginary time, *Quantum* **7**, 916 (2023).
- [14] S. Wang, S. McArdle, and M. Berta, Qubit-efficient randomized quantum algorithms for linear algebra, *PRX Quantum* **5**, 020324 (2024).
- [15] S. Lu, M. C. Bañuls, and J. I. Cirac, Algorithms for quantum simulation at finite energies, *PRX Quantum* **2**, 020321 (2021).
- [16] K. Seki and S. Yunoki, Energy-filtered random-phase states as microcanonical thermal pure quantum states, *Phys. Rev. B* **106**, 155111 (2022).
- [17] Z. Zhang, A. Wang, X. Xu, and Y. Li, Measurement-efficient quantum Krylov subspace diagonalisation, *Quantum* **8**, 1438 (2023).

- [18] O. Kyriienko, Quantum inverse iteration algorithm for programmable quantum simulators, [npj Quantum Inf. 6, 7 \(2020\)](#).
- [19] M.-Q. He, D.-B. Zhang, and Z. Wang, Quantum Gaussian filter for exploring ground-state properties, [Phys. Rev. A 106, 032420 \(2022\)](#).
- [20] T. Keen, E. Dumitrescu, and Y. Wang, Quantum algorithms for ground-state preparation and Green's function calculation, [arXiv:2112.05731](#).
- [21] S. Chakraborty, Implementing any linear combination of unitaries on intermediate-term quantum computers, [Quantum 8, 1496 \(2024\)](#).
- [22] G. Leobacher and F. Pillichshammer, *Introduction to Quasi Monte Carlo Integration and Applications* (Birkhäuser, Cham, 2014).
- [23] M. Menickelly, Y. Ha, and M. Otten, Latency considerations for stochastic optimizers in variational quantum algorithms, [Quantum 7, 949 \(2023\)](#).
- [24] K. Ito, Latency-aware adaptive shot allocation for run-time efficient variational quantum algorithms, [arXiv:2302.04422](#).
- [25] H. D. Raedt, A. H. Hams, K. Michielsen, S. Miyashita, and K. Saito, Quantum statistical mechanics on a quantum computer, [Prog. Theor. Phys. Suppl. 138, 489 \(2000\)](#).
- [26] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, [SIAM J. Comput. 46, 1920 \(2017\)](#).
- [27] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd ed. (Dover Publications, Mineola, NY, 2007).
- [28] Y. Kawamata, Codes: Quasi Monte Carlo method for linear combination unitaries via classical postprocessing, Zenodo (2026), <https://doi.org/10.5281/zenodo.19659154>.
- [29] L. N. Trefethen and J. Weideman, The exponentially convergent trapezoidal rule, [SIAM Rev. 56, 385 \(2014\)](#).