Are Randomized Quantum Linear Systems Solvers Practical?

Siddharth Hariprakash*†‡§‖, Roel Van Beeumen¶, Katherine Klymko*, Daan Camps*‖

Abstract

Randomized quantum algorithms have been proposed in the context of quantum simulation and quantum linear algebra with the goal of constructing shallower circuits than methods based on block encodings. While the algorithmic complexities of such randomized schemes have been shown to be suboptimal, it has been speculated that they may offer benefits in the early fault-tolerant era where circuit depth comes at a premium. In this work, we investigate the end-to-end resource requirements for a randomized quantum linear systems solver to estimate scalar properties of the matrix inverse by combining sampling from a Fourier series with Hamiltonian simulation. We derive explicit bounds on all relevant algorithmic parameters that control the total error. We analyze two algorithmic kernels for Hamiltonian simulation: a second order product formula approximation and a method called random Taylor expansion (RTE).

Finally, we provide numerical demonstrations that confirm the validity of our analytical results and question the actual practicality of the randomized Fourier series-based approach for linear systems problems as we show that the sampling complexity can grow exponentially. By providing explicit bounds, our work serves as a bridge between theoretical algorithmic proposals and efficient hardware implementations while also enabling fair comparisons to alternative algorithms that exhibit optimal asymptotic complexities at the cost of large resource overheads.

1 Introduction

Quantum computers hold significant promise for solving complex computational problems in scientific computing, particularly within the physical sciences [1–5]. This has inspired a broader search for quantum algorithms that can effectively solve *classical* computational problems arising in combinatorial optimization, machine learning, and linear algebra [6, 7]. In this work, we focus on the latter and study the topic of *quantum linear algebra*.

Most existing approaches to quantum linear algebra rely on a fully coherent, oracle-based data access model with Hamiltonian simulation [8, 9] and block-encoding access [10] as two of the most common assumptions. Hamiltonian simulation is a key quantum algorithmic primitive; it is known to be in the BQP complexity class, and is efficiently implementable on quantum computers for k-local and sparse Hamiltonians [11–16]. Consequently, Hamiltonian simulation has become a widespread quantum routine and a foundation for many subsequent quantum algorithms. As a subroutine, it provides query access to the time evolution operator $e^{-\mathrm{i}Ht}$ with controllable error ϵ. In particular, the first quantum algorithm to solve linear systems of equations proposed by Harrow, Hassidim, and Lloyd (HHL) makes use of Hamiltonian simulation [8]. On the other hand, a block encoding is an embedding of a non-unitary operator into a larger unitary, and is central to algorithms based on the quantum

*National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

†Leinweber Institute for Theoretical Physics and Department of Physics, University of California, Berkeley, California 94720, USA

‡Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

§BlueQubit Inc, San Francisco, CA 94105, USA

¶Applied Mathematics and Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

‖email: SiddharthHariprakash@lbl.gov; dcamps@lbl.gov

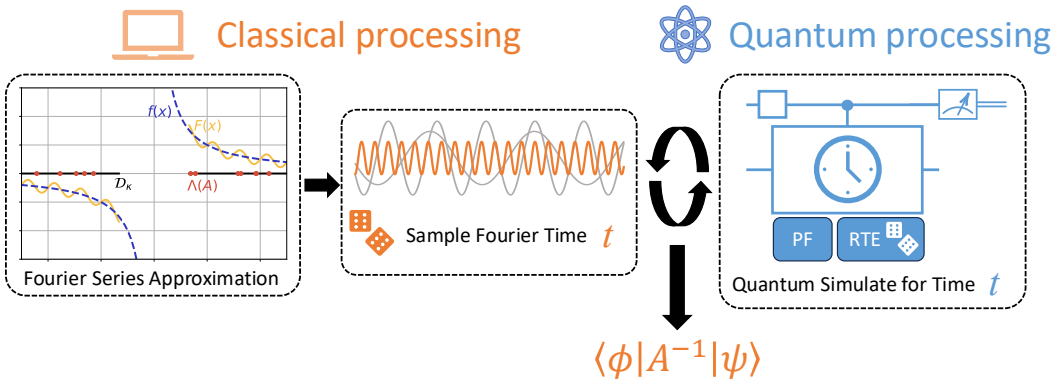

Figure 1: Randomized framework for sampling from solutions to the quantum linear systems (QLS) problem $\langle\phi| A^{-1} |\psi\rangle$ for a Hermitian matrix A, consisting of a classical processing step (**left**) which constructs a provably accurate Fourier series approximation to the inverse function, followed by repeatedly sampling a Fourier time t and constructing an approximate time evolution operator $U \approx e^{-\mathrm{i}At}$ using either a Product Formula (PF) or Random Taylor Expansion (RTE). Next the quantum processing step (**right**) uses $\log(N) + 1$ qubits to measure $\langle\phi| U |\psi\rangle$ for each sample. The statistical average over all samples converges to $\langle\phi| A^{-1} |\psi\rangle$.

singular value transformation (QSVT) [10, 17]. Block encodings provide query access to the operator itself or to polynomial transformations of the operator through QSVT and its variants. Quantum circuits for block encodings can be constructed explicitly for certain classes of highly structured, data-sparse problems [18–23] without relying on quantum random access memory (QRAM) [24], or from linear combinations of unitaries (LCU) [25, 26], which is efficient for operators that are sparse in the Pauli basis.

Coherent data access via block encoding models generally requires deep and complicated quantum circuits that will remain challenging to implement for near-term quantum computers, even if some degree of fault-tolerance is achieved. In an effort to bypass the challenge of constructing block encodings, a new class of randomized algorithms have been proposed in the context of quantum linear algebra. For this approach, the computational problem is reformulated such that a classical sampling procedure combined with a Hamiltonian simulation subroutine can be used to provide Monte Carlo estimates to the desired solution upon averaging over all estimates. It has been shown that a key advantage of this approach is that the circuit complexity of each individual sample can be reduced at the cost of increasing the total number of samples required to reach a certain precision. Prominent examples of randomized quantum algorithms include methods for Hamiltonian simulation [12, 27], for statistical phase estimation [28], and for computing more general matrix functions [29–32].

In this work, we build on [28, 29] and present an explicit randomized quantum algorithm for quantum linear algebra problems based on Fourier series approximations. We consider the task of estimating scalar properties of matrix functions $f(A)$ for an $N \times N$ Hermitian operator A, i.e., quantities of the form $\langle\phi| f(A) |\psi\rangle$[1]. We primarily focus on the quantum linear systems (QLS) problem, that is $f(A) = A^{-1}$, but provide a brief overview of how to extend the methods that we present to arbitrary functions. The proposed method is summarized in Fig. 1 and preserves the key advantages of [28, 29]: only a single ancilla qubit leading to a $\log(N) + 1$ qubit complexity for N-dimensional problems and tunable circuit and sampling complexities.

We note that all previous work in the space of Fourier series based randomized QLS solvers provide only asymptotic complexities. While useful, it is unclear what the actual end-to-end resource requirements are. The problem of constructing the Fourier series is often left as an open question and hence

[1]We do not consider the problem of computing the required normalization factor in this work, see [29] for a careful treatment of this topic

its contribution to the final algorithmic complexities have not been thoroughly studied. In this work, we accurately characterize the error made by approximating the inverse function by one possible choice for the Fourier series approximation proposed in recent work [33]. We also place upper bounds on the variance introduced as a result of sampling individual terms from this Fourier series. Thus, this allows us to compute end-to-end resource requirements for the full randomized algorithm including classical and quantum subroutines. We show that even for moderately sized problems, our analysis puts into question the practicality of such randomized schemes.

After defining the problem and giving an overview of the randomized procedure in Section 2, our main contributions include:

i A detailed and complete analysis of the truncation and discretization errors stemming from approximating the inverse function as a Fourier series presented in Section 3. This includes *non-asymptotic* choices for the parameters used to construct the Fourier series given some error tolerance, using a scheme that requires exponentially fewer terms compared to previous approaches. A comprehensive description of the normalization and sampling procedure is included in Section 4.

ii *Non-asymptotic* algorithmic resource estimates given an error budget. Specifically, we compute the number of samples required by the randomization procedure as well as the non-Clifford gate depth of each sample. In Section 5 we present results for two different strategies to implement Hamiltonian simulation. The two approaches are (1) product formulas (PF) [11, 34] and (2) an implementation based on *random Taylor expansion* (RTE) [28, 29][2].

iii End-to-end numerical demonstrations provided in Section 6 for a set of relevant small-scale examples that empirically confirm the validity of our results while also questioning the practicality of such randomized schemes.

We end with a discussion of our results in Section 7.

2 Proposed method

2.1 Problem statement

Let $\mathcal{H}$ be an $N = 2^n$-dimensional Hilbert space and $A \in \mathbb{C}^{N\times N}$: $\mathcal{H} \to \mathcal{H}$ be some Hermitian operator on this space. We assume that the Pauli decomposition of A is known and given by,

$$A = \sum_{\ell\in[L]} c_\ell P_\ell, \tag{1}$$

where $[L] := \{0, 1, \ldots, L-1\}$, each $P_\ell \in \{I, X, Y, Z\}^{\otimes n}$ (i.e., a length-n Pauli string), and each coefficient $c_\ell \in \mathbb{R}$. We assume efficient classical access to the coefficients and classical representation of the Paulis, $\{(c_\ell, P_\ell)\}_{\ell\in[L]}$[3]. We also assume access to the Pauli weight of A, defined as $\lambda := \sum_\ell |c_\ell|$. Furthermore, we assume that $\|A\| \geq 1$. From the unitarity of the Pauli matrices and the triangle inequality, this automatically implies that $\lambda \geq 1$. We also assume that the condition number of A, $\kappa := \kappa(A) = \|A^{-1}\|\|A\|$, is known or that we have an upper bound $\kappa^* \geq \kappa$. Finally, we also assume access to state preparation oracles for the pair of states $|\psi\rangle, |\phi\rangle \in \mathcal{H}$, which means that we have unitaries U_ψ, U_ϕ such that for a reference state $|g\rangle \in \mathcal{H}$,

$$|\psi\rangle = U_\psi |g\rangle\,, \ |\phi\rangle = U_\phi |g\rangle\,. \tag{2}$$

For the remainder of this paper, we assume that the cost associated with state preparation is constant and, for simplicity, we keep $|g\rangle = |0\rangle$. The problem statement that we study in this paper can be summarized as follows:

[2]Previous work [28, 29] referred to this result as the random compiler lemma. To avoid confusion with randomized compiling, also known as Pauli twirling, we stick to the terminology of random Taylor expansion introduced in [35].

[3]Each length-n Pauli string requires $2n$ classical bits of storage and each coefficient is assumed to be stored in some floating point representation requiring a constant amount of storage per coefficient.

Problem 1. *Given classical access to the Pauli decomposition (1) for an $N \times N$ Hermitian operator A with Pauli weight λ, condition number upper bounded by κ^*, state preparation oracles U_ψ, U_ϕ for $|\psi\rangle, |\phi\rangle$, and approximation error $\epsilon > 0$, construct an estimator $\hat{Z}$ where $\mathbb{E}[\hat{Z}] \approx \langle\phi| A^{-1} |\psi\rangle$. The number of samples N_S is to be chosen such that the statistical average $Z_{N_S} := N_S^{-1} \sum_{i\in[N_S]} \hat{Z}_i$ satisfies*

$$\left| Z_{N_S} - \langle\phi| A^{-1} |\psi\rangle \right| \leq \epsilon, \tag{3}$$

with probability greater than $1 - \delta$ for $\delta \in (0, 1)$.

2.2 Description of the randomized algorithm

The following provides a more detailed overview of the different steps of the algorithm. We highlight whether each individual step requires classical or quantum resources (or both).

The algorithm begins with a series of **classical pre-processing** steps, the first of which is to approximate the inverse function, $f(x) = x^{-1}$, over some domain $\mathcal{D} \subset \mathbb{R}_0$ by a Fourier series approximation $F(x)$,

$$f(x)\Big|_{\mathcal{D}} \approx \sum_{s\in[S]} \alpha_s e^{-\mathrm{i}xt_s} =: F(x), \tag{4}$$

that satisfies,

$$\sup_{x\in\mathcal{D}} \left| f(x) - F(x) \right| \leq \epsilon_F, \tag{5}$$

where S is the number of terms in the Fourier series approximation, $t_s \in \mathbb{R}$ is the Fourier time, $\alpha_s \in \mathbb{C}$ is the Fourier coefficient, and ϵ_F is the error on the Fourier series approximation. If the spectral range of the operator A falls within the domain $\mathcal{D}$, then Eq. (5) implies that

$$\sup_{x\in\mathcal{D}} \|A^{-1} - F(A)\| \leq \epsilon_F. \tag{6}$$

Given κ^* and λ, we use the methods in Section 3 to construct the domain $\mathcal{D}$ and hence $F(A)$ given an error tolerance ϵ_F. We show that the complexity of this step is independent of the Hilbert space dimension.

After this initial pre-processing, we use a combination of **classical and quantum processing**. In particular, we use an iterative and randomized routine such that at each step j we sample a Fourier time $t_j \in \{t_s\}_{s\in[S]}$ with probability $|\alpha_j| / \sum_{s\in[S]} |\alpha_s|$ and consider the exponential $e^{-\mathrm{i}At_j}$. We now seek a method to implement an approximate controlled version of the unitary to derive an estimate for $\langle\phi| e^{-\mathrm{i}At_j} |\psi\rangle$ using a Hadamard test. At this juncture, we consider two approaches for constructing a unitary that serves as either an approximation or estimator (respectively) for the controlled version of $e^{-\mathrm{i}At_j}$ that we use for implementing the Hadamard test circuit:

1. *Product Formula* (PF): We can approximate $e^{-\mathrm{i}At_j}$ with a product/Trotter formula by splitting the full exponential into products of "simpler" exponentials such that each exponential involves only a single n-qubit Pauli string. For example, one can use a 2^{nd} order PF approximation as follows (see Section 5.1 for more details):

$$e^{-\mathrm{i}At_j} \approx U(t_j) := e^{-\mathrm{i}c_0 P_0 t_j/2} e^{-\mathrm{i}c_1 P_1 t_j/2} \dots e^{-\mathrm{i}c_{L-1} P_{L-1} t_j} e^{-\mathrm{i}c_{L-2} P_{L-2} t_j/2} \dots e^{-\mathrm{i}c_0 P_0 t_j/2}. \tag{7}$$

   We note that to implement a controlled version of $U(t_j)$ on quantum hardware, we can control each of the "simpler" exponentials individually.

2. *Random Taylor Expansion* (RTE): Alternatively, we can use Lemma 2 from [28] to approximately express the time evolution operator $e^{-\mathrm{i}At_j}$ as a linear combination of unitaries (LCU):

$$e^{-\mathrm{i}At_j} \approx \sum_{m\in\mathcal{M}} d_{mj} U_{mj}, \tag{8}$$

for some appropriately defined index set $\mathcal{M}$. From this LCU we sample an individual unitary $U(t_j) \in \{U_{mj}\}_{m\in\mathcal{M}}$ (by interpreting the coefficients in the LCU as being proportional to probabilities) such that $\mathbb{E}[U(t_j)] \approx e^{-\mathrm{i}At_j}$, thus introducing a second level of randomization. As we show in Section 5.2, for appropriately chosen algorithmic parameters the unitary $U(t_j)$ can be expressed as a series of Pauli rotations and an insignificant number of Clifford operations. Thus, the primary source of the gate complexity to construct a controlled call to $U(t_j)$ on quantum hardware comes from controlling a series of Pauli rotations.

For each Fourier sample, let $U(t_j)$ be the unitary obtained from one of the two subroutines outlined above. We define N_{CP} as the number of controlled Pauli rotations required to implement a controlled call to $U(t_j)$ and will discuss in Section 5 how to choose this parameter given a certain error budget. We set $\tilde{U}(t_j) := U_\phi^\dagger U(t_j) U_\psi$. We now use *two shots*[4] (one each for the real part and imaginary part of the overlap) of the Hadamard test to obtain estimates for the overlap $\langle 0|\tilde{U}(t_j)|0\rangle = \langle\phi|U(t_j)|\psi\rangle$. We denote the result as $Z(t_j) \in \mathbb{C}$, where we note that $\mathbb{E}[Z(t_j)] = \langle\phi|U(t_j)|\psi\rangle$. This completes a single iteration j of this hybrid workflow. We denote the total number of iterations, and thus the total number of samples of the form $\hat{Z}(t_j)$, by N_{S}. Each such $\hat{Z}(t_j)$ is then stored on the classical device. We note that all such iterations are independent of each other and can be conveniently parallelized.

The final step involves **classical post-processing** to compute the statistical average of all the samples that are collected:

$$Z_{N_{\mathrm{S}}} = N_{\mathrm{S}}^{-1} \sum_{j\in[N_{\mathrm{S}}]} \hat{Z}(t_j). \tag{9}$$

The quantities N_{S} and N_{CP} are chosen such that

$$\left|Z_{N_{\mathrm{S}}} - \langle\phi|A^{-1}|\psi\rangle\right| \leq \left|Z_{N_{\mathrm{S}}} - \langle\phi|F(A)|\psi\rangle\right| + \left|\langle\phi|F(A)|\psi\rangle - \langle\phi|A^{-1}|\psi\rangle\right| \leq \epsilon_S + \epsilon_F \leq \epsilon, \tag{10}$$

with probability greater than $1-\delta$ thus solving Problem 1.

Eq. (10) shows that the total approximation error can be characterized by considering two sources of error: (1) the sampling error ϵ_S when measuring $\langle\phi|F(A)|\psi\rangle$, and (2) the error on the Fourier series approximation ϵ_F. Each of these can be further attributed to more detailed sources of error. A simple strategy to distribute the total error budget is the balanced case of $\epsilon_S, \epsilon_F \leq \epsilon/2$. We stick to this convention and choose algorithmic parameters to control each source of error independently. Throughout our analysis the key *figures of merit* (or cost model) we consider consist of the sample complexity (N_{S}) and the number of controlled Pauli rotations per sample (N_{CP}). In this work, N_{CP} does not include the cost of preparing the states $|\psi\rangle, |\phi\rangle$. In Section 5 we provide two lower bounds for these figures of merit, one for each of the two Hamiltonian simulation subroutines that we consider. In particular, we show that there exists a tradeoff between these two figures of merit: in general, increasing N_{S} allows us to decrease N_{CP} and vice-versa. We also numerically compare the RTE and PF methods in Section 6 by performing end-to-end classical simulations for a selection of small-scale examples.

We provide a high-level flow chart of the full randomized quantum algorithm in Fig. 2 that summarizes the content of this section. We note that these methods are not specific to the inverse problem and can be used to estimate scalar properties of any matrix function $f(A)$ for which a provably accurate Fourier series approximation can be constructed.

3 Fourier series approximation to the inverse function with explicit error bounds

We start with the following integral representation for the inverse function, which was also used in [36],

$$x^{-1} = \frac{\mathrm{i}}{\sqrt{2\pi}} \int_0^\infty dy \int_{-\infty}^\infty dz \left(z e^{-z^2/2} e^{-\mathrm{i}xyz}\right) =: I(x). \tag{11}$$

[4] See Appendix A for why we choose this method over performing multiple shots for each Fourier sample

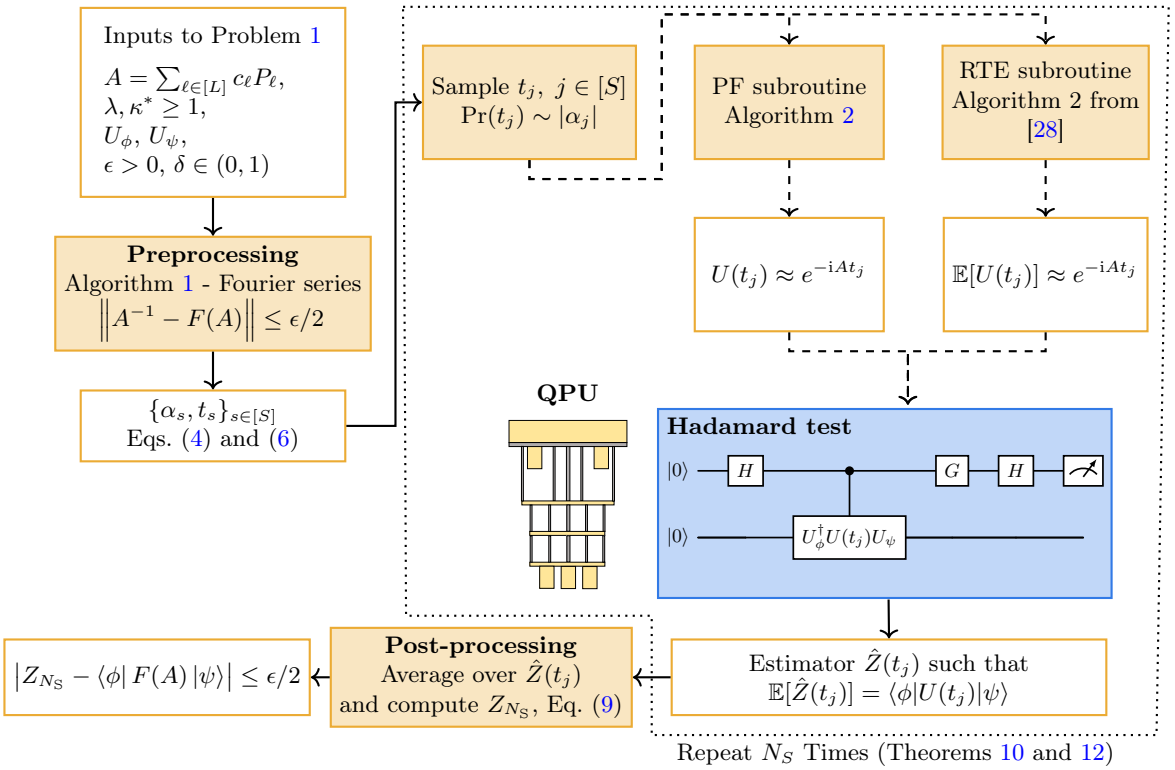

Figure 2: Flowchart outlining the randomized quantum algorithm for estimating $\langle\phi| A^{-1} |\psi\rangle$. Orange boxes are classical data processing steps, with boxes having a white background describing inputs, intermediate results, and outputs, and those filled with an orange background describing a classical algorithm. The blue box shows the Hadamard test circuit which is the only step that requires a QPU. The dashed lines indicate the iterative portion of the algorithm that uses a combination of classical and quantum processing. The dashed arrows indicate a choice in the method used to perform Hamiltonian simulation.

Next, we define the domain $\mathcal{D}_\kappa = \left[-1, -1/\kappa\right] \cup \left[1/\kappa, 1\right]$, with $\kappa \geq 1$, that excludes the singularity at zero and provide a method to construct a finite LCU (or Fourier series) approximation to the double integral in Eq. (11) over $\mathcal{D}_\kappa$ by discretizing the integral. This requires two steps: (1) truncating the infinite integral appearing in Eq. (11) and (2) discretizing this truncated integral. It is in these two steps that we improve upon [36] and derive a more efficient discetization of Eq. (11) by following a similar approach as in [33]. To this end, we introduce two definitions for the truncation and discretization steps.

Definition 2 (ϵ-close integral truncation)**.** *The truncated integral $I_T^{(\epsilon)} : \mathcal{D}_\kappa \to \mathbb{C}$ which approximates $I(x)$, the integral representation of the inverse function given in Eq. (11), is defined as,*

$$I_T^{(\epsilon)}(x) := \frac{\mathrm{i}}{\sqrt{2\pi}} \int_0^{y_{\max}} dy \int_{-z_{\max}}^{z_{\max}} dz \left(z e^{-z^2/2} e^{-\mathrm{i}xyz} \right), \tag{12}$$

where the pair of positive real truncation parameters $(y_{\max}, z_{\max}) \in \mathbb{R}^+ \times \mathbb{R}^+$ *determine the integration bounds and are chosen to satisfy,*

$$\sup_{x\in\mathcal{D}_\kappa} \left| I_T^{(\epsilon)}(x) - I(x) \right| \leq \epsilon. \tag{13}$$

Next, we consider the discretization of the truncated integrals. This requires us to introduce two

positive *weight functions*, $w_y : [0, y_{\max}] \to \mathbb{R}^+$ for the y-integral and $w_z : [-z_{\max}, z_{\max}] \to \mathbb{R}^+$ for the z-integral.

Definition 3 (ϵ-close integral discretization)**.** *Let the* Fourier parameters $(J, K) \in \mathbb{N} \times \mathbb{N}$. *The discretization* $I_D^{(\epsilon,\epsilon')} : \mathcal{D}_\kappa \to \mathbb{C}$ *of* $I_T^{(\epsilon)}$ *is defined as follows,*

$$I_D^{(\epsilon,\epsilon')}(x) := \frac{\mathrm{i}}{\sqrt{2\pi}} \sum_{j\in[J]} \sum_{k\in[K]} \left(w_y^{(j)} w_z^{(k)} z_k e^{-z_k^2/2} e^{-\mathrm{i}x y_j z_k} \right), \tag{14}$$

where $w_y^{(j)} := w_y(y_j)$, $w_z^{(k)} := w_z(z_k)$ *and the quadrature nodes* $y_j \in [0, y_{\max}], z_k \in [-z_{\max}, z_{\max}]$ *for* $(j, k) \in [J] \times [K]$. *The Fourier parameters are chosen to satisfy,*

$$\sup_{x\in\mathcal{D}_\kappa} \left| I_D^{(\epsilon,\epsilon')}(x) - I_T^{(\epsilon')}(x) \right| \leq \epsilon. \tag{15}$$

We remark that the discretization in Eq. (14) can be interpreted as a Fourier series approximation to the inverse function with JK terms, where each product of the form $y_j z_k$ is interpreted as a *Fourier time* (or frequency) and each product of the form $\mathrm{i} w_y^{(j)} w_z^{(k)} z_k e^{-z_k^2/2}/\sqrt{2\pi}$ as a *Fourier coefficient* (or amplitude).

It follows from Definitions 2 and 3 that the total error accrued while constructing approximations to the inverse function, which we denote by ϵ_F in accordance with Eq. (10), can be attributed to two sources which we denote by ϵ_T and ϵ_D for the truncation and discretization errors respectively. In particular, we have that,

$$\begin{aligned} \epsilon_F := \sup_{x\in\mathcal{D}_\kappa} \left| I_D^{(\epsilon_D,\epsilon_T)}(x) - I(x) \right| &\leq \sup_{x\in\mathcal{D}_\kappa} \left| I_D^{(\epsilon_D,\epsilon_T)}(x) - I_T^{(\epsilon_T)}(x) \right| + \sup_{x\in\mathcal{D}_\kappa} \left| I_T^{(\epsilon_T)}(x) - I(x) \right|, \\ &\leq \epsilon_D + \epsilon_T. \end{aligned}$$

Our goal now is to construct a provably accurate Fourier series approximation to the inverse function that requires as few as possible total terms JK and as short as possible maximum simulation time $t_{\max} := y_{\max} z_{\max}$ given a target approximation error ϵ. This is precisely the topic of the remainder of this section and our results are summarized in Algorithm 1. To satisfy the balanced choice of $\epsilon_F \leq \epsilon/2$ made in Section 2.2, we can use Algorithm 1 with $\epsilon_D + \epsilon_T \leq \epsilon/2$ to construct the appropriate Fourier series approximation to the inverse function.

3.1 Bounds on the truncation and discretization error

The following theorem characterizes the truncation error ϵ_T and describes how to choose the truncation parameters $(y_{\max}, z_{\max})$ for each of the integrals appearing in Eq. (12).

Theorem 4. *Given a truncation error* $\epsilon_T > 0$, *the integration bounds* $(y_{\max}, z_{\max}) \in \mathbb{R}^+ \times \mathbb{R}^+$ *given by*

$$y_{\max} = \kappa \sqrt{2 \ln\left(\frac{3\kappa}{\epsilon_T}\right)}, \quad z_{\max} = \sqrt{2 \ln\left(\frac{3\kappa}{\epsilon_T}\right)} \tag{16}$$

provide a truncated integral approximation $I_T^{(\epsilon_T)}$, *as defined in Definition 2, to* $I(x)$ *over the domain* $\mathcal{D}_\kappa$.

Proof. The proof is given in Appendix C.1 □

We note that Theorem 4 allows us to compute the maximal Fourier time parameter $t_{\max}$ for the Fourier series $I_T^{(\epsilon_T)}$ as follows,

$$t_{\max} = 2\kappa \ln\left(\frac{3\kappa}{\epsilon_T}\right). \tag{17}$$

Next, we present lower bounds on the Fourier parameters (J, K) that appear in each of the sums in Eq. (14) such that the discretization error to approximate an integral of the form $I_T^{(\epsilon_T)}$ is bounded from above by a given error tolerance ϵ_D. We note that this analysis requires specific choices for the quadrature rules for both the z- and y-integral appearing in Eq. (12). We follow the proposal introduced in [33] and combine discretization by the trapezoidal rule for the z-integral and the Gauss-Legendre rule for the y-integral to select the quadrature nodes and weights.

For the z-integral appearing in Eq. (12), we observe that the integrand is quasi-periodic, rapidly decaying (Gaussian), and analytic. It has been established that under these conditions the trapezoidal rule convergences exponentially [37]. In the notation of Eq. (14), for an appropriately chosen value of the Fourier parameter K (given some error tolerance), we can discretize the z-integral using the trapezoidal rule with the constant weight function,

$$w_z^{(k)} = \frac{2z_{\max}}{(K-1)} =: \Delta z, \tag{18}$$

with quadrature nodes $z_k = \Delta z(k - (K-1)/2)$ for $k \in [K]$. For the y-integral appearing in Eq. (12), the integrand is a purely oscillatory function that does not a decay and hence the trapezoidal rule does not converge exponentially fast. As proposed in [33], we instead make use of the Gauss-Legendre quadrature rule for the y-integral. We use the change of variable $y \mapsto y_{\max}(1+y)/2$ to rescale and write Eq. (12) as,

$$I_T^{(\epsilon)}(x) = \frac{\mathrm{i}y_{\max}}{2\sqrt{2\pi}} \int_{-1}^{1} dy \int_{-z_{\max}}^{z_{\max}} dz \left(z e^{-z^2/2} e^{-\mathrm{i}x y_{\max}(1+y)z/2} \right). \tag{19}$$

In the notation of Eq. (14), we can thus discretize the y-integral in Eq. (19) by using J many Legendre nodes and setting $y_j = y_{\max}(1+c_j)/2$, where $c_j \in [-1, 1]$ is the j^{th} root of the J^{th} degree Legendre polynomial. We also set $w_y^{(j)}$ to be the usual Gauss-Legendre quadrature weight scaled by the factor $y_{\max}/2$, i.e.,

$$w_y^{(j)} = \frac{y_{\max}}{2} \frac{2}{(1-y_j^2)(P_J''(y_j))^2}, \tag{20}$$

where P_J denotes the degree-J Legendre polynomial.

Thus, we have effectively approximated the integral appearing in Eq. (12) using a discretization of the form appearing in Eq. (14). We use the error bounds presented in [33, 37] to arrive at the following theorem on how to choose the Fourier parameters (J, K) to control the total discretization error.

Theorem 5. *Given a discretization error $\epsilon_D > 0$, the Fourier parameters $(J, K) \in \mathbb{N} \times \mathbb{N}$,*

$$K = 1 + \frac{1}{\pi}\left(\ln(2) + \frac{z_{\max}^2}{2} + 2\kappa \ln\left(\frac{3\kappa}{\epsilon_T}\right) + \ln\left(\frac{2}{\epsilon_D}\right) + \ln\left(1 + \frac{2}{z_{\max}\sqrt{2\pi}}\right)\right), \tag{21}$$

$$J = \frac{1}{\ln(2)}\left(\ln\left(\frac{K}{K-1}\right) + \ln\left(\frac{32 y_{\max} z_{\max}^2}{\epsilon_D \sqrt{2\pi}}\right) + \frac{3\kappa \ln\left(\frac{3\kappa}{\epsilon_T}\right)}{2\sqrt{2}}\right), \tag{22}$$

provide an integral discretization $I_D^{(\epsilon_D, \epsilon_T)}$, as defined in Definition 3, of $I_T^{(\epsilon_T)}(x)$ over the domain $\mathcal{D}_\kappa$.

Proof. The proof is given in Appendix C.2 □

3.2 Matrix rescaling

We can apply the results presented in this section to the task of approximating the inverse of a Hermitian matrix. We remind the reader that we assume $\|A\| \geq 1$ and that we know an upper bound on the condition number $\kappa \leq \kappa^*$. We distinguish between two cases. If $\|A\| = 1$, then $\Lambda(A) \subseteq \mathcal{D}_{\kappa^*}$,

where $\Lambda(A)$ indicates the set of eigenvalues of A, and the previous results readily apply. Alternatively, if $\|A\| > 1$, then it does not necessarily hold that $\Lambda(A) \subseteq \mathcal{D}_{\kappa^*}$. In this case, we can rescale the problem as follows.

Lemma 6. *Let A be a Hermitian matrix with $\|A\| > 1$, upper bound on the condition number κ^*, and Pauli weight λ. Define the rescaled matrix $\tilde{A}$ and the rescaled upper bound on the condition number $\tilde{\kappa}^*$ as follows:*

$$\tilde{A} := \frac{A}{\lambda}, \qquad\qquad \tilde{\kappa}^* := \lambda\kappa^*.$$

Then $\Lambda(\tilde{A}) \subseteq \mathcal{D}_{\tilde{\kappa}^*}$.

Proof. The proof is given in Appendix C.3. □

Note that the Pauli weight of $\tilde{A} = 1$. Thus, we may apply Theorems 4 and 5 to $\tilde{A}$ on the domain $\mathcal{D}_{\tilde{\kappa}^*}$ to obtain a Fourier series approximation for $\tilde{A}^{-1}$. We can then multiply this Fourier series by λ^{-1} to obtain an approximation of A^{-1}. Note that the error on the approximation to A^{-1} is also scaled by λ^{-1}.

Algorithm 1: Fourier series approximation to the inverse function A^{-1}.

Input:
- κ^*: upper bound on the condition number of A,
- λ: Pauli weight of A,
- ϵ_T, ϵ_D : truncation and discretization error tolerances respectively such that $\epsilon_T + \epsilon_D \leq \epsilon/2$.

Output: Fourier coefficients and time parameters $\{\alpha_s, t_s\}_{s\in[S]}$ such that $\|A^{-1} - \lambda^{-1}\sum_{s\in[S]} \alpha_s e^{-\mathrm{i}\tilde{A}t_s}\| \leq \frac{(\epsilon_T+\epsilon_D)}{\lambda}$

$\tilde{\kappa}^* \leftarrow \lambda\kappa^*$;

$y_{\max} \leftarrow \tilde{\kappa}^*\sqrt{2\ln\left(\frac{3\tilde{\kappa}^*}{\epsilon_T}\right)}$, $z_{\max} \leftarrow \sqrt{2\ln\left(\frac{3\tilde{\kappa}^*}{\epsilon_T}\right)}$; // Eq. (16)

$K \leftarrow 1 + \frac{1}{\pi}\left(\ln(2) + \frac{z_{\max}^2}{2} + 2\tilde{\kappa}^*\ln\left(\frac{3\tilde{\kappa}^*}{\epsilon_T}\right) + \ln\left(\frac{2}{\epsilon_D}\right) + \ln\left(1 + \frac{2}{z_{\max}\sqrt{2\pi}}\right)\right)$; // Eq. (21)

$J \leftarrow \frac{1}{\ln(2)}\left(\ln\left(\frac{K}{K-1}\right) + \ln\left(\frac{32 y_{\max} z_{\max}^2}{\epsilon_D\sqrt{2\pi}}\right) + \frac{3\tilde{\kappa}^*\ln\left(\frac{3\tilde{\kappa}^*}{\epsilon_T}\right)}{2\sqrt{2}}\right)$; // Eq. (22)

$\Delta z \leftarrow \frac{2z_{\max}}{K-1}$;

foreach $(j,k) \in [J]\times[K]$ **do**

  $x_j \leftarrow j$-th root of the J-th degree Legendre polynomial;

  $y_j \leftarrow y_{\max}(1+x_j)/2$;

  $w_y^{(j)} \leftarrow \frac{y_{\max}}{2}\frac{2}{(1-y_j^2)(P_J'(y_j))^2}$; // Eq. (20)

  $z_k \leftarrow \Delta z\left(k - (K-1)/2\right)$;

  $\alpha_{jk} \leftarrow \left(\frac{\mathrm{i}}{\sqrt{2\pi}} w_y^{(j)}\Delta z\right) z_k e^{-z_k^2/2}$;

  $t_{jk} \leftarrow y_j z_k$;

end foreach

return $\{\alpha_{jk}, t_{jk}\}_{(j,k)\in[J]\times[K]}$;

In the rest of this work we present our results in terms of the rescaled upper bound on the condition number $\tilde{\kappa}^*$. Algorithm 1 summarizes the content of this section and combines it with the previous results from this section.

4 Sampling from the Fourier series approximation

Having characterized the Fourier series approximation error ϵ_F appearing in Eq. (10), we shift our attention over the next two sections to characterizing the sampling error ϵ_S. In the current section, we introduce the routine we use to sample from the Fourier series approximation. To set up sampling access and interpret the Fourier coefficients as probabilities, we have to normalize this set of coefficients. In particular, we define the following normalization parameters:

$$N_y := \frac{1}{\sqrt{2\pi}} \sum_{j\in[J]} \left| w_y^{(j)} \right|, \quad N_z := \sum_{k\in[K]} \left| w_z^{(k)} z_k e^{-z_k^2/2} \right|, \tag{23}$$

where $w_y^{(j)}$ is given by Eq. (20) and $w_z^{(k)}$ is given by Eq. (18). These normalization parameters now allow us to define the following probability distributions:

$$p_y^{(j)} := \frac{\left| w_y^{(j)} \right|}{N_y}, \quad p_z^{(k)} := \frac{\left| w_z^{(k)} \right|}{N_z}, \tag{24}$$

and use these to rewrite the Fourier series generated by Algorithm 1 as follows:

$$\begin{aligned} \sum_{s\in[S]} \alpha_s e^{-\mathrm{i}\tilde{A}t_s} &= \lambda^{-1} I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}), \\ &= \lambda^{-1} N_y N_z \sum_{j\in[J]} \sum_{k\in[K]} \omega(j,k) p_y^{(j)} p_z^{(k)} e^{-\mathrm{i}\tilde{A}t_{jk}}, \end{aligned} \tag{25}$$

where each $\omega(j,k)$ has unit norm and represents the appropriate phase factor, i,e.

$$\omega(j,k) := \frac{\mathrm{i} w_y^{(j)} w_z^{(k)}}{\left| w_y^{(j)} \right| \left| w_z^{(k)} \right|}. \tag{26}$$

We can sample indices $j' \in [J]$ with probability $p_y^{(j')}$ and indices $k' \in [K]$ with probability $p_z^{(k')}$, and thus effectively sample exponentials of the form $e^{-\mathrm{i}\tilde{A}y_{j'}z_{k'}} = e^{-\mathrm{i}\tilde{A}t_{j'k'}}$ from Eq. (25). The variance of sampling from this distribution is upper bounded by the product $\lambda^{-1}N_yN_z$. The following Lemma uses Theorem 4 to express N_y and an upper bound for N_z in terms of parameters introduced in Section 3:

Lemma 7. *The normalization parameters N_y, N_z defined in Eq. (23) satisfy:*

$$N_y = \frac{y_{\max}}{\sqrt{2\pi}} = \frac{\tilde{\kappa}^*}{\sqrt{2\pi}} \sqrt{2\ln\left(\frac{3\tilde{\kappa}^*}{\epsilon_T}\right)}, \quad N_z \leq \frac{2z_{\max}^2\sqrt{2\pi}}{K-1} = \frac{4\sqrt{2\pi}}{K-1} \ln\left(\frac{3\tilde{\kappa}^*}{\epsilon_T}\right), \tag{27}$$

where K is given by Theorem 5.

Proof. The proof is given in Appendix C.4 □

Lemma 7 can thus used to upper bound the variance of sampling individual exponentials from Eq. (25). In particular, we note that the upper bound on the variance scales poly-logarithmically with $\tilde{\kappa}^*$.

5 Algorithmic resource requirements

Let τ be a Fourier time of the form $t_{j'k'}$ that we sample using the protocol outlined in Section 4. We now consider the task of implementing controlled versions of operators of the form $e^{-\mathrm{i}\tilde{A}\tau}$. In particular,

we consider two subroutines to accomplish this task: (i) Section 5.1 outlines a product formula (or PF) approach to construct approximate time evolution operators, and (ii) Section 5.2 outlines the random Taylor expansion (or RTE) scheme to sample from the LCU decomposition for controlled time evolution operators, which introduces a second randomized procedure into the overall algorithm. We derive the number of quantum circuit samples and gate depth per sample required to solve Problem 1 for each subroutine.

5.1 Complexity of the product formula approach

We consider the rescaled matrix $\tilde{A}$ defined in Section 3.2 and write it as,

$$\tilde{A} = \lambda^{-1} \sum_{\ell \in [L]} c_\ell P_\ell := \sum_{\ell \in [L]} \tilde{P}_\ell, \tag{28}$$

where we have absorbed the coefficients c_ℓ/λ into the modified Paulis $\tilde{P}_\ell$. The PF method allows one to construct the exponential $e^{-\mathrm{i}\tilde{A}\tau}$ by decomposing the full exponential into products of exponentials each involving only a single $\tilde{P}_\ell$. The exponential of a single $\tilde{P}_\ell$ can then be implemented using the standard CNOT ladder construction [38]. The controlled version of the exponential of a single $\tilde{P}_\ell$ thus requires a controlled single-qubit Pauli rotation and $4\log(N)$ Cliffords [28]. The most common method to construct a PF is using the Lie/Suzuki Trotter construction. For explicit constructions we refer the reader to [11]. The number of terms in a PF grows exponentially with the order of the formula, and in practice only low order product formulas are used. In the rest of this work we will consider the 2^{nd} order formula[5] given by,

$$S(\tau) := \prod_{\ell \in [L]} e^{-\mathrm{i}\tilde{P}_\ell \tau/2} \prod_{\ell \in [L]} e^{-\mathrm{i}\tilde{P}_{L-\ell} \tau/2}. \tag{29}$$

5.1.1 Approximation error of the second order product formula

As shown in [11], Eq. (29) provides a good approximation to the true exponential when τ is small since the error terms scale as a polynomial in τ. For finite τ, we can split the full interval τ into r smaller steps τ/r and use a 2^{nd} order PF approximation for each step, i.e.,

$$e^{-\mathrm{i}\tilde{A}\tau} \approx \big(S(\tau/r)\big)^r := S_r(\tau). \tag{30}$$

The quantity $r \in \mathbb{N}$ is known as the Trotter number. The question we ask now is how to choose r to ensure the approximation error,

$$\epsilon_{\text{PF}} := \|e^{-\mathrm{i}\tilde{A}\tau} - S_r(\tau)\|, \tag{31}$$

is bounded. The following lemma establishes an upper bound on ϵ_{PF} for a given choice r:

Lemma 8. *For a given Trotter number r, the error in operator norm ϵ_{PF} defined in Eq. (31) for a 2^{nd} order PF satisfies the following upper bound:*

$$\epsilon_{\text{PF}} \leq \frac{f\tau^3}{r^2}, \tag{32}$$

where $f \in \mathbb{C}$ is defined as follows:

$$f := \frac{1}{12} \sum_{\ell_0=0}^{L-1} \left\| \left[\sum_{\ell_2=\ell_0}^{L-1} \tilde{P}_{\ell_2}, \left[\sum_{\ell_1=\ell_0+1}^{L-1} \tilde{P}_{\ell_1}, \tilde{P}_{\ell_0} \right] \right] \right\| + \frac{1}{24} \sum_{\ell_0=0}^{L-1} \left\| \left[\tilde{P}_{\ell_0}, \sum_{\ell_1=\ell_0+1}^{L-1} \tilde{P}_{\ell_1} \right] \right\|. \tag{33}$$

[5] We note that our results can be easily generalized to arbitrary orders.

Proof. The proof of this lemma follows from applying the triangle inequality r times to the result in Proposition 16 from [11]. □

We can use Lemma 8 to construct provably accurate PF approximations to exponentials of the form $e^{-\mathrm{i}\tilde{A}\tau}$. In particular, given an upper bound on ϵ_{PF}, which we call $\epsilon_{\mathrm{PF}}*$, choosing

$$r \geq \sqrt{\frac{f\tau^3}{\epsilon^*_{\mathrm{PF}}}}, \tag{34}$$

ensures that $\epsilon_{\mathrm{PF}} \leq \epsilon^*_{\mathrm{PF}}$. We point out that while this provides a valid choice for the Trotter number r, empirical evidence shows that in many cases of interest this bound is pessimistic and an error ϵ_{PF} can be reached at significantly smaller Trotter numbers than that implied by Eq. (34) (for example see [11, 34, 39]). However, tighter analytic bounds require knowledge of the specific matrix of interest, so we stick to Eq. (34) for our analysis. We also note that the gate complexity, in terms of the number of single-qubit Pauli rotations, of implementing a PF formula scales linearly with the total number of terms in the PF and hence with r. In particular, we require $2rL$-many single-qubit Pauli rotations for the 2^{nd} order PF $S_r(\tau)$. We summarize the PF construction routine in Algorithm 2.

Algorithm 2: Construct a 2^{nd} order PF approximation to $e^{-\mathrm{i}\tilde{A}\tau}$

Input:
- Pauli decomposition $\tilde{A} = \sum_{\ell\in[L]} \tilde{P}_\ell$
- $\tau \in \mathbb{R}$
- $\epsilon^*_{\mathrm{PF}} \in \mathbb{R}^+$

Output: Unitary $S_r(\tau)$ such that:
- the non-Clifford cost of controlling $S_r(\tau)$ is equal to that of controlling $2rL$ Pauli rotations
- $\epsilon_{\mathrm{PF}} \leq \epsilon^*_{\mathrm{PF}}$

$U \leftarrow \mathbb{1}$
$r \leftarrow \sqrt{f\tau^3/\epsilon^*_{\mathrm{PF}}}$; // Eq. (34)
$x \leftarrow \tau/r$
foreach $\ell \in [L]$ **do**
  $U \leftarrow U e^{-i\tilde{P}_\ell x/2}$
end foreach
foreach $\ell \in [L]$ **do**
  $U \leftarrow U e^{-i\tilde{P}_{L-\ell} x/2}$
end foreach
$S_{r_{\mathrm{PF}}}(t_{j'k'}) \leftarrow U^{r_{\mathrm{PF}}}$
return $S_r(\tau)$

5.1.2 Sampling error for the product formula approach

Having constructed $S_r(\tau)$, we can now estimate the overlap given by $\langle\phi| S_r(\tau) |\psi\rangle$ using the Hadamard test circuit shown in Fig. 2. We set $U = U^\dagger_\phi S_r(\tau) U_\psi$ and obtain two separate measurement outcomes (shots), each being either ± 1, that are estimates for the real and imaginary parts of the desired overlap respectively. Let $\hat{X}_\tau, \hat{Y}_\tau$ be the random variables associated (respectively) with these outcomes. These are unbiased estimators for the real and imaginary parts of $\langle\phi| S_r(\tau) |\psi\rangle$,

$$\mathbb{E}[\hat{X}_\tau] = \Re \langle 0| U^\dagger_\phi S_r(\tau) U_\psi |0\rangle , \quad \mathbb{E}[\hat{Y}_\tau] = \Im \langle 0| U^\dagger_\phi S_r(\tau) U_\psi |0\rangle . \tag{35}$$

We now include the phase and normalization factors shown in Eq. (25) and define the random variable $\hat{Z}_\tau$ as follows:

$$\hat{Z}_\tau := \omega_\tau N_y N_z \lambda^{-1} \left(X_\tau + \mathrm{i} Y_\tau \right), \tag{36}$$

and note that this is a biased estimator for the quantity $\lambda^{-1} \langle \phi | I_D^{(\epsilon_D, \epsilon_T)}(\tilde{A}) | \psi \rangle$,

$$\mathbb{E}[\hat{Z}_\tau] + B_{\mathrm{PF}} = \lambda^{-1} \langle \phi | I_D^{(\epsilon_D, \epsilon_T)}(\tilde{A}) | \psi \rangle, \tag{37}$$

where B_{PF} is the bias term that arises due to the fact that we replaced the exponential $e^{-\mathrm{i}\tilde{A}\tau}$ with a PF approximation. We present the following upper bound on the bias term:

Theorem 9. *The norm of the bias term B_{PF} appearing in Eq. (37) can be upper bounded as follows:*

$$|B_{\mathrm{PF}}| \leq \lambda^{-1} N_y N_z \frac{f t_{\max}^3}{r^2}, \tag{38}$$

where N_y and N_z are given by Lemma 7, and $t_{\max}$ is given by Eq. (17).

Proof. The proof is given in Appendix C.5. □

The following theorem now provides a way to choose the number of samples and controlled Pauli rotations required per sample when utilizing the PF subroutine:

Theorem 10. *Let $Z_{N_{\mathrm{S}}}$ be the random variable obtained by taking the average of N_{S} many samples of the biased estimator $\hat{Z}_\tau$ (defined in Eq. (36)) such that the norm of the bias defined in Eq. (37) is bounded by $B_{\mathrm{PF}} < \epsilon/2$ for some choice of Trotter number r. If $\epsilon_D + \epsilon_T \leq \epsilon/2$, then $Z_{N_{\mathrm{S}}}$ solves Problem 1 if:*

$$N_{\mathrm{S}} = 2 + \ln\left(\frac{2}{\delta}\right) \frac{16 (N_y N_z)^2}{\lambda^2 (\epsilon/2 - |B_{\mathrm{PF}}|))^2}, \quad N_{\mathrm{CP}} = 2rL, \tag{39}$$

where N_{CP} provides an upper bound on the number of controlled Pauli rotations required per sample (see Section 2.2), N_y and N_z are given by Lemma 7, and $t_{\max}$ is given by Eq. (17).

Proof. The proof is given in Appendix C.6. □

We observe that there exists a trade-off between N_{S} and N_{CP}. For instance, if we increase the value of the Trotter number r, we observe that this would increase N_{CP}. However, from Theorem 9, we see that this will also decrease the upper bound on B_{PF} and hence (in general) decrease the number of samples N_{S}. Similarly, the effect of decreasing r will be to increase N_{S}. Thus, using a more accurate PF decreases the overall sample complexity while increasing the non-Clifford gate complexity and vice versa. Using Theorem 9, we observe that the condition $B_{\mathrm{PF}} < \epsilon/2$ implies the a lower bound on r,

$$r > \sqrt{\frac{2 N_y N_z f t_{\max}^3}{\lambda \epsilon}}, \tag{40}$$

which yields the following lower bound on N_{CP} using Theorem 10,

$$N_{\mathrm{CP}} > 2L \sqrt{\frac{2 N_y N_z f t_{\max}^3}{\lambda \epsilon}}. \tag{41}$$

5.2 Complexity of the random Taylor expansion approach

In this section, we consider the sampling based approach presented in [28] to construct an estimator for the exponential $e^{-\mathrm{i}\tilde{A}\tau}$. In particular, we use a Taylor series expansion to approximately represent the exponential as a Linear Combination of Unitaries (LCU). We sample individual unitaries according to the coefficients of the LCU. The controlled version of each such unitary can then be used in the Hadamard test circuit step shown in Fig. 2 to estimate the overlap $\langle \phi | \tilde{A}^{-1} | \psi \rangle$. This approach was also used in [29] for randomized quantum linear algebra algorithms.

5.2.1 Sampling from the approximate Taylor expansion of the time evolution operator

Similar to the PF method (see Eq. (30)), we divide the Fourier time τ into r segments and write,

$$e^{-\mathrm{i}\tilde{A}\tau} = \left(e^{-\mathrm{i}\tilde{A}\tau/r}\right)^r. \tag{42}$$

As shown in Appendix B and [28], we can Taylor expand each exponential $e^{-\mathrm{i}\tilde{A}\tau/r}$ and truncate the resulting infinite series expansion to obtain an approximate expansion as follows,

$$e^{-\mathrm{i}\tilde{A}\tau/r} \approx \sum_{\substack{n=0 \\ n \text{ even}}}^{n_{\max}} \frac{1}{n!}\left(\frac{\mathrm{i}\tau}{r}\right)^n \sqrt{1+\left(\frac{\tau/r}{n+1}\right)^2} \left(\sum_{\ell\in[L]} \frac{c_\ell}{\lambda} P_\ell\right)^n \sum_{\ell\in[L]} \frac{c_\ell}{\lambda} e^{i\theta_n P_\ell}, \tag{43}$$

where $n_{\max} \in \mathbb{N} \cup \{0\}$ denotes the truncation parameter and θ_n in Eq. (43) is given by

$$\theta_n = \cos^{-1}\left(\left(1+\left(\frac{\tau/r}{n+1}\right)^2\right)^{-1/2}\right). \tag{44}$$

We can rewrite Eq. (43) as a (finite) LCU,

$$e^{-\mathrm{i}\tilde{A}\tau/r} \approx \sum_{m\in\mathcal{M}'} d'_{m\tau} U'_{m\tau}, \tag{45}$$

where $\mathcal{M}'$ represents the appropriately defined index set, each coefficient $d'_{m\tau} \in \mathbb{R}^+$ by absorbing their phase into their respective unitary $U'_{m\tau} \sim P_{\ell_1} P_{\ell_2} \dots P_{\ell_n} e^{i\theta_n P_{\ell_{n+1}}}$, where $\ell_1, \ell_2, \dots, \ell_n, \ell_{n+1} \in [L]$ and $n \in [n_{\max}]$. We note that each unitary $U'_{m\tau}$ requires a sequence of at most $n_{\max}$ Clifford operations and *one* multi-qubit Pauli rotation. The latter can be efficiently synthesized into a sequence of Clifford operations (CNOTs) and one single-qubit Pauli rotation using the standard CNOT ladder construction [38]. It follows that the non-Clifford cost of a controlled-$U'_{m\tau}$ is equivalent to that of controlling one single-qubit Pauli rotation.

The LCU approximation for $e^{\mathrm{i}\tilde{A}\tau}$ can then be obtained by raising Eq. (45) to the r^{th} power as follows,

$$e^{-\mathrm{i}\tilde{A}\tau} \approx \left(\sum_{m\in\mathcal{M}'} d'_{m\tau} U'_{m\tau}\right)^r := \sum_{m\in\mathcal{M}} d_{m\tau} U_{m\tau}, \tag{46}$$

for the appropriately defined index set $\mathcal{M} \cong \left(\mathcal{M}'\right)^r$. Thus, a controlled-$U_{m\tau}$ has a non-Clifford cost equivalent to controlling r single-qubit Pauli rotations.

We now consider the problem of sampling individual unitaries of the form $U_{m\tau}$ from the LCU shown in Eq. (46). We begin by rewriting Eq. (45) as follows:

$$e^{-\mathrm{i}\tilde{A}\tau/r} \approx \alpha \sum_{m\in\mathcal{M}'} \frac{d'_{m\tau}}{\alpha} U'_{m\tau} := \alpha \sum_{m\in\mathcal{M}'} p'_{m\tau} U'_{m\tau}, \tag{47}$$

with $\alpha = \sum_{m\in\mathcal{M}'} d'_{m\tau}$ the one-norm of the vector of LCU coefficients such that $p'_{m\tau}$ can be interpreted as the probability to sample the unitary $U'_{m\tau}$. To sample a unitary of the form $U_{m\tau}$ from Eq. (46), such that $\mathbb{E}[U_{m\tau}] = e^{-\mathrm{i}\tilde{A}\tau}$, we can sample r unitaries of the form $U'_{m\tau}$ according to the probabilities $p'_{m\tau}$ in Equation (47) and form their product. The overall sampling normalization constant, denoted by N_{RTE}, for this procedure is,

$$N_{\mathrm{RTE}} = \alpha^r \leq e^{\tau^2/r}, \tag{48}$$

where we note that the upper bound follows from Appendix C of [28]. We refer to Algorithm 2 in [28] which presents a classical approach to sample and return a description of r-many unitaries of the form $U'_{m\tau}$ according to the probabilities $p'_{m\tau}$ without having to construct the full set of probabilities $\{p'_{m\tau}\}_{m\in\mathcal{M}}$.

5.2.2 Sampling error for the random Taylor expansion approach

Up to this point, all computations are performed classically. Similar to the method outlined in Section 5.1.2, we can now construct an estimator for $\langle\phi| U_{m\tau} |\psi\rangle$ using the Hadamard test circuit for $U_\phi^\dagger U_{m\tau} U_\psi$. Let $X_{b\tau}, Y_{b\tau}$ be the random variables associated (respectively) with the 2 outcomes of the Hadamard test. We note that these are unbiased estimators for the real and imaginary parts of $\langle\phi| U_{m\tau} |\psi\rangle$:

$$\mathbb{E}[X_{b\tau}] = \Re \langle 0| U_\psi^\dagger U_{m\tau} U_\phi |0\rangle , \quad \mathbb{E}[Y_{b\tau'}] = \Im \langle 0| U_\psi^\dagger U_{m\tau} U_\phi |0\rangle . \tag{49}$$

Similar to Eq. (36), we define a random variable $Z_{b\tau}$ as follows:

$$Z_{b\tau} \equiv \omega_\tau N_{\mathrm{RTE}} N_y N_z \lambda^{-1} (X_{b\tau} + iY_{b\tau}) \tag{50}$$

and observe that it is a biased estimator for the desired quantity $\lambda^{-1} \langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}) |\psi\rangle$:

$$\mathbb{E}[Z_{b\tau}] + B_{\mathrm{RTE}} = \lambda^{-1} \langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}) |\psi\rangle , \tag{51}$$

where B_{RTE} represents the bias term that arises from the fact that Eq. (43) is an approximation to the true Taylor series expansion of the time evolution operator obtained by performing a truncation. If we assume $r \geq \tau$, we can present the following upper bound on this bias term.

Theorem 11. *Let $t_{\min}$ be defined as follows:*

$$t_{\min} := \min_{(j,k)\in[J]\times[K]} \left(t_{jk}\right) . \tag{52}$$

Then, for all choices of r satisfying $r \geq t_{\max}$, the following upper bound holds:

$$|B_{\mathrm{RTE}}| \leq \frac{r N_y N_z e^{(t_{\max}^2 + |t_{\max}| - |t_{\min}|)/r}}{2} \left(\frac{e t_{\max}}{r n_{\max}}\right)^{n_{\max}} , \tag{53}$$

where N_y and N_z are given by Lemma 7, and $t_{\max}$ is given by Eq. (17).

Proof. The proof is given in Appendix C.7. □

We observe that for fixed τ, r, N_y, N_z, and λ the upper bound on B_{RTE} falls off super exponentially as a function of $n_{\max}$ and can thus be made negligible for appropriately chosen values of $n_{\max}$. We now present efficiently computable choices for the sample and non-Clifford gate complexities required by the RTE approach given some error budget:

Theorem 12. *Let Z_{N_S} be the random variable obtained by taking the average of N_S many samples of the biased estimator $Z_{b\tau}$ defined in Eq. (50) such that the norm of the bias defined in Eq. (51) is bounded by $B_{\mathrm{RTE}} < \epsilon/2$. If $\epsilon_D + \epsilon_T \leq \epsilon/2$, then Z_{N_S} solves Problem 1 if:*

$$N_S = 1 + \ln\left(\frac{2}{\delta}\right) \frac{16(e^{t_{\max}^2/r} N_y N_z)^2}{\lambda^2(\epsilon/2 - |B_{\mathrm{RTE}}|)^2}, \quad N_{\mathrm{CP}} = r, \tag{54}$$

where N_{CP} provides an upper bound on the number of controlled Pauli rotations required per sample (see Section 2.2), N_y and N_z are given by Lemma 7, and $t_{\max}$ is given by Eq. (17) .

Proof. The proof is given in Appendix C.8. □

If we choose $r \geq t_{\max}$, then the condition $B_{\mathrm{RTE}} < \epsilon/2$ can be satisfied by setting the upper bound from Eq. (53) to be less than $\epsilon/2$:

$$\frac{rN_yN_ze^{(t_{\max}^2+|t_{\max}|-|t_{\min}|)/r}}{2}\left(\frac{et_{\max}}{rn_{\max}}\right)^{n_{\max}} < \frac{\epsilon}{2}, \tag{55}$$

from which an appropriate value of $n_{\max}$ can be determined. An important feature of the result shown in Theorem 12 is that the sufficient choice for N_{S} scales exponentially with the ratio $t_{\max}^2/r$. For instance, choosing $r = t_{\max}$ implies the need for exponentially many samples with respect to $t_{\max}$ while keeping the gate depth per sample constant. If we instead choose $r = \mathcal{O}(t_{\max}^2)$, then by using Theorem 11 to choose sufficiently large $n_{\max}$, we can essentially strip N_{S} of its dependence on $t_{\max}^2$ at the cost of N_{CP} picking up quadratic dependence on $t_{\max}^2$. Thus, there exists a tradeoff between the sample complexity and the gate complexity per sample[6]. We note that if one decides to move the dependence on $t_{\max}$ over to the gate complexity r, it might be more efficient to adaptively choose the value of r based on each sample Fourier time instead of setting it to be a constant function of $t_{\max}$. We also point out that if one decides to keep $r = \mathcal{O}(1)$ for instance (independent of $t_{\max}$), the task of generating N_{S}-many samples may both become computationally intractable for many problems due to the exponential dependence on $t_{\max}^2$, see Section 6.

6 Numerical results

6.1 Error on the Fourier series approximation to the inverse function

In this section, we compare the theoretical bounds on the Fourier series approximation presented in Theorems 4 and 5 with empirical results for randomly generated 4×4 matrices with given condition number κ. We construct these random matrices by first generating diagonal matrices of the form $D = \mathrm{diag}\left(\pm 1/\kappa, \ldots, \pm 1\right)$, where the elements on the diagonal not explicitly shown are chosen randomly from the uniform distribution on $[-1, -1/\kappa] \cup [1/\kappa, 1]$ such that $\mathrm{spec}(D) \in \mathcal{D}_\kappa$. For each instance of D, we generate a 4×4 unitary matrix U randomly sampled from the group $U(4)$ according to the Haar measure. The matrix $A \equiv UDU^\dagger$ is then a randomly generated 4×4 matrix with condition number κ. We generate 100 such random matrices for each pair of condition number and error threshold values,

$$(\kappa, \epsilon_F) \in \{10, 10^2, 10^3\} \times \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}.$$

For each of the random matrices, we compute the Pauli weight λ and found that $\lambda \in [1.5, 2.1]$ and thus the values of $\tilde{\kappa}^* = \lambda\kappa$ lie within the range $[15, 2.1 \times 10^3]$. We set $\epsilon_T = \epsilon_D = \epsilon_F/2$ and use Algorithm 1 to construct the Fourier series approximation to the inverse of each random matrix,

$$A^{-1} \approx F(A) = \lambda^{-1} I_D^{(\epsilon_F/2, \epsilon_F/2)}(\tilde{A}), \tag{56}$$

such that the approximation error ϵ_F satisfies,

$$\lambda \cdot \|A^{-1} - \lambda^{-1} I_D^{(\epsilon_F/2, \epsilon_F/2)}(\tilde{A})\| \leq \epsilon_F. \tag{57}$$

For each trial matrix, we reconstruct the Fourier series approximation to the inverse and compute the error in the spectral norm with respect to each of the exact inverses scaled by the Pauli weight, i.e. the quantity shown on the LHS of Eq. (57). Our numerical results show that inequality (57) is satisfied for every trial matrix. In Table 1 we show our results in terms of the maximum errors observed across each of the 100 trials, denoted by $\epsilon_{\max}(\kappa, \epsilon_F)$, for each pair (κ, ϵ_F). We observe that $\epsilon_{\max}$ lies around 20% of the bound ϵ_F in all cases. Furthermore, we observe that the number of terms in the Fourier

[6] As noted in Appendix D of [28] one can in principle solve a one-dimensional problem to compute the elements $\{r_{j,k}\}_{(j,k)\in[J]\times[K]}$, where we have assigned a unique value of r for each Fourier time, that minimize either the total complexity $N_{\mathrm{S}}N_{\mathrm{CP}}$ or N_{S} (resp. N_{CP}) given an upper bound on N_{CP} (resp. N_{S}).

expansion JK increases with increasing condition number κ and decreasing approximation error ϵ_F. We note that we used the C++ implementation available in `FASTGL` [40] to efficiently compute the Gauss-Legendre quadrature up to degrees of $\mathcal{O}(10^6)$ required for the discretization $I_D^{(\epsilon_D,\epsilon_T)}$ according to Definition 3.

κ	ϵ_F	J	K	$\epsilon_{\max}(\kappa,\epsilon_F)$
10	10^{-2}	154	62	1.81×10^{-3}
10^2		1709	708	2.08×10^{-3}
10^3		20388	8478	2.03×10^{-3}
10	10^{-3}	194	78	2.10×10^{-4}
10^2		2065	856	2.023×10^{-4}
10^3		23915	9945	2.03×10^{-4}
10	10^{-4}	232	94	2.03×10^{-5}
10^2		2421	1004	1.76×10^{-5}
10^3		27442	11413	1.81×10^{-5}
10	10^{-5}	271	110	1.76×10^{-6}
10^2		2777	1152	1.81×10^{-6}
10^3		30969	12880	2.02×10^{-6}

Table 1: Numerical verification of the bounds on the Fourier series approximation to the matrix inverse derived in Theorems 4 and 5 for randomly generated 4×4 matrices with condition numbers $\kappa\in\{10,10^2,10^3\}$ and approximation errors $\epsilon_F\in\{10^{-2},10^{-3},10^{-4},10^{-5}\}$. The table reports the Fourier parameters (J,K) obtained from the theoretical bounds and the maximum numerical error over 100 trials.

6.2 End-to-end numerical simulations of the randomized quantum linear systems solver

In this section, our aim is to perform an end-to-end numerical simulation of the randomized quantum linear systems solver as described in Problem 1. In particular, we use the method described in Section 6.1 to generate a 4×4 matrix A with $\kappa=100$, $\lambda=2.09$, and thus $\tilde{\kappa}^*=209$. We set $\epsilon_D=\epsilon_T=\epsilon_F/2=10^{-3}$ and thus $\epsilon_F=2\times10^{-3}$. We then use Algorithm 1 to construct the Fourier series approximation $I_D^{(10^{-3},10^{-3})}$. We note that Eq. (17) implies that $t_{\max}$ for this Fourier series approximation is given by,

$$t_{\max}=2\cdot209\ln\left(\frac{3\cdot209}{10^{-3}}\right)\approx5580. \tag{58}$$

Instead of an exact numerical reconstruction of the Fourier series approximation as in Section 6.1, we simulate the randomized quantum algorithm described in Fig. 2. We set $U_\psi=U_\phi=I$ so that the randomized algorithm estimates $\langle0|A^{-1}|0\rangle$. For the real-time evolutions sampled from the Fourier series approximation, we tested both the product formula and the random Taylor expansion approaches. To simulate the statistical noise coming from a single shot of the Hadamard test, we add noise sampled from a Gaussian of width 1 to the numerical results [41].

A product formula implementation. We start by using the product formula subroutine outlined in Section 5.1. We perform classical simulations of Algorithm 2 to obtain a PF approximation to each sampled Fourier exponential of the form $e^{-iA\tau}$ for 3 different choices of Trotter number r: a fixed value at $r=500$, and two adaptive choices $\lceil0.05\tau^2\rceil$, and $\lceil0.1\tau^2\rceil$. We note that Eq. (58) implies

that for the latter two choices, we have $r \lesssim 1.6 \times 10^6$ and 3.2×10^6 respectively. To compute each PF approximation, we compute matrices of the form $e^{-iA\tau/r}$ and exponentiate each to the power r. We generate $N_s = 10^8$ random samples from the simulated Hadamard test, compute the average, and take the real part to obtain the final approximation to $\langle 0| A^{-1} |0\rangle$. Fig. 3 shows the convergence of the root mean squared error (RMSE) averaged over 20 trials as a function of the number of simulated shots. We note that by choosing a constant value for the Trotter number $r = 500$ instead of choosing it adaptively based on the sampled Fourier time τ, we reach an error plateau determined by the region where the error in the product formula dominates the sampling error. In other words, the condition $B_{\mathrm{PF}} < \epsilon/2$ from Theorem 10 is no longer satisfied, and therefore taking additional samples has little to no effect on the overall approximation error. We observe this effect in the orange curve in Fig. 3 around $N_S \approx 10^6$.

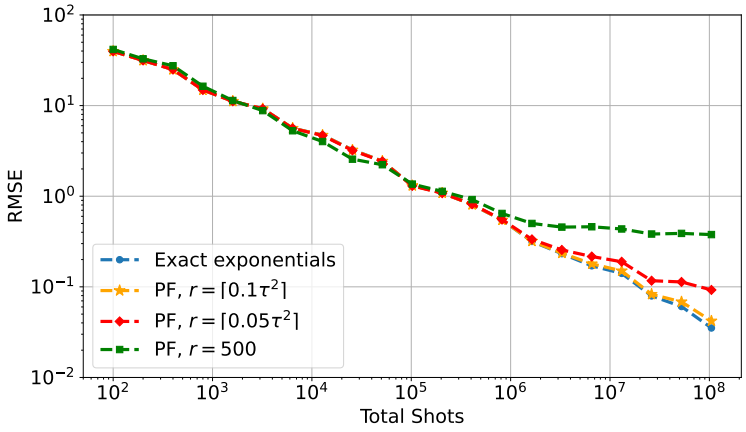

Figure 3: Convergence in the root mean squared error (RMSE) as a function of the total number of simulated shots, where the exponential $e^{-iA\tau}$ corresponding to each Fourier sample is approximated by a 2nd order product formula. Results are shown for three different choices of Trotter number r and for exact numerical evaluation of the real-time evolutions. To each such approximation, we add noise sampled from a Gaussian centered at 0 with width 1 to simulate the noise coming from a single shot of the Hadamard test circuit. We note that not choosing the Trotter number r to depend on the Fourier time τ leads to a plateau in the RMSE as a result of the bias dominating over the sampling error.

A random Taylor expansion implementation. The situation with RTE is different. We see from Theorem 11 that for each sampled Fourier time, we can in theory choose a constant value for the time step r as long as $r \geq t_{\max}$ and $n_{\max}$ is large enough to ensure that Eq. (55) is satisfied. If for instance we choose $r = t_{\max}$, we can technically use Theorem 11 to find a good choice for $n_{\max}$. The prefactor in Eq. (55) is approximately $\propto e^{t_{\max}^2/r} = e^{t_{\max}}$(which corresponds to the upper bound on N_{RTE} from Eq. (48)) and, even for our small-scale example, this value is astronomical, roughly e^{5580}. For this reason we were unable to compute and use the necessary value of $n_{\max}$ from the bound. Instead, we choose $n_{\max} = 150$ which is the largest value before underflow occurs in the computation of $1/n_{\max}!$ in the coefficients of the RTE. Furthermore, for the parameter choice $r = t_{\max}$, Theorem 12 implies that N_S also has an exponential dependence on $t_{\max}$ to converge to the desired error with the same prefactor of roughly e^{5580}. For these reasons, we were unable to simulate the RTE subroutine with $r = t_{\max}$ as part of the end-to-end randomized algorithm to compute $\langle 0|A^{-1}|0\rangle$ for our trial matrix.

Instead, we consider the simpler task of estimating the quantity $\Re \langle 0| e^{-\mathrm{i}A\tau} |0\rangle$ for modest values of τ such that the relevant prefactors are smaller and a classical simulation is possible. In particular we assume a maximal simulation time $t_{\max} = 100$ and keep $r = t_{\max}$. We consider values of τ between 1 and 80 and denote $U_{\mathrm{RTE}}(\tau)$ as a unitary sampled by the RTE method such that $\mathbb{E}[\langle 0| U_{\mathrm{RTE}}(\tau) |0\rangle] \approx$

$\Re \langle 0| e^{iA\tau} |0\rangle$. In Fig. 4 we plot the convergence of the RMSE of this estimator, averaged over 20 trials, as a function of the number of RTE samples collected. To isolate the convergence properties of the RTE method, we do not simulate noise coming from the Hadamard test and compute each overlap $\Re \langle 0| U_{\mathrm{RTE}}(\tau) |0\rangle$ exactly. We observe that gradual increases in τ and the ratio τ^2/r lead to very large increases in the RMSE. Thus, we see that even for modest choices for τ the RTE method is unable to accurately approximate even a single exponential in this parameter regime.

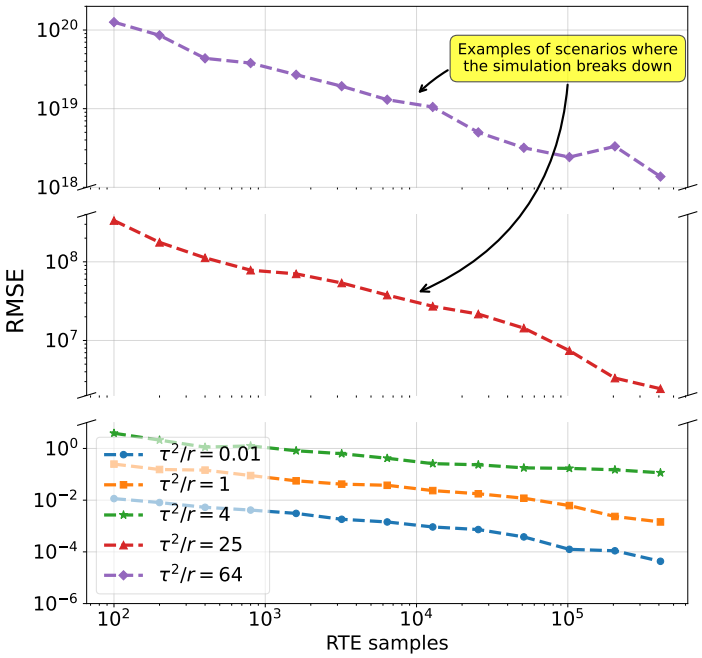

Figure 4: Convergence in the root mean squared error (RMSE) as a function of the total number of RTE samples for the task of approximating $\Re \langle 0| e^{-iA\tau} |0\rangle$. The RTE procedure uses $r = t_{\max} = 100$ and simulation time τ is varied from 1 to 80 and hence the ratio τ^2/r varies from 0.01 to 64. We observe that as the ratio τ^2/r increases past 1, the prefactor of the RTE method is severely impacted as illustrated by the large RMSE for the red and purple curves.

As discussed in Section 5.2, we can attempt to mitigate the impact of this exponential prefactor by alernatively choosing larger $r \sim t_s^2$ as a different parameter regime. However, this implies an upper bound $r \lesssim t_{\max}^2 \approx 31.14 \times 10^6$ and hence for *each* sampled Fourier time, Algorithm 2 from [28] has to sample the descriptions of and compute the product of up to 31 million unitaries $U'_{m\tau}$ (see Section 5.2). This would require multiplying up to 31 million (possibly unique) unitary matrices, rather than computing the rth power of a single matrix as was done for the PF simulations. This long matrix multiplication leads to impractically long computations.

For all reasons listed above, we were unable to perform a classical simulation using the RTE method to sample an approximation to $\langle 0| A^{-1} |0\rangle$ for our choice of A in a reasonable amount of time and resources in either parameter regime. Thus, an important consideration when using the RTE method is the trade-off between exponentially slow convergence, resource intensive sampling procedures, and deep quantum circuits.

7 Discussion

In this work, we analytically compute the resources required for a randomized hybrid quantum algorithm to estimate scalar properties of functions of Hermitian matrices, i.e., quantities of the form

$\langle\phi|f(A)|\psi\rangle$. We focus on the particular case of the inverse function and use a Fourier series approximation to evaluate the inverse of non-singular matrices A. Our results in Section 3 provide explicit bounds on the truncation and discretization parameters of the integral representation of the inverse function that determines the Fourier series approximation. Section 5 studies two subroutines, PFs and the RTE method, to implement the time evolutions $e^{-\mathrm{i}At}$ sampled from this Fourier series and presents upper bound estimates for the sample complexity N_{S} and the number of controlled single-qubit Pauli rotations required per sample N_{CP}. Since our analysis is exact and does not consider asymptotic scaling (as in previous work [29]), one can use our results to explicitly compute all algorithmic parameters such that the result provably satisfies a desired error threshold as long as a problem description is given, i.e., the Pauli decomposition of a Hermitian matrix A, upper bound κ^* on the condition number of A, and the Pauli weight λ of A. Thus, our work in principle enables a full end-to-end implementation of the randomized routine shown in Fig. 2.

However, by providing such explicit algorithmic resource requirements, our work also questions the utility of randomized algorithmic schemes such as those presented in [28, 29] for the specific example of the linear systems problem. As we showed in Sections 5 and 6.2, even modest-sized problems with moderate condition number require enormous classical and quantum resources to satisfy a reasonable error threshold. For both the PF and RTE approaches, this complexity is caused in part by the large variance of the probability distribution of the Fourier coefficients from which samples are drawn. For the RTE approach, there is an additional very large contribution to the variance as a result of the secondary sampling step from the approximate Taylor expansion of the time evolution operator. As we showed in Section 5.2, we can mitigate the latter contribution at the expense of choosing the gate depth to scale polynomially with the Fourier time. This also increases the number of unitaries one needs to sample classically (Algorithm 2 from [28]) to generate an estimator for each Fourier time. As shown in Section 6.2, when we combine these sources for the total variance, RTE requires immense resources. In particular, we need to sample on the order of tens of millions of gates for each individual Fourier time, such that even for a small-scale problem with condition number $\kappa = 100$ the numerical simulation of the algorithm becomes impractical. As presented in Section 6.2, this meant that we could only perform an end-to-end numerical simulation of the PF approach and we could not detect the onset of convergence of the RTE approach in a reasonable number of samples.

The current work provides one possible construction of a Fourier series approximation for the inverse function using a combination of a Gauss-Legendre and trapezoidal rule discretization of the inverse function [33]. In order to improve the practicality of randomized schemes, one may be able to construct an approximation scheme with a more optimal dependence on the condition number κ. As noted in [29], one can in principle apply classical preconditioners to A, thus reducing the condition number κ and accelerating convergence. We also note that future work may be able to construct lower variance versions of the RTE subroutine enabling it as an viable estimator. Another interesting future direction would be to consider matrices with well-defined structure and develop a more refined sampling procedure that respects this structure to efficiently bias and improve the polynomial dependence on both κ^* and $1/\epsilon$.

Our work thus serves as a bridge between theoretical developments (provably accurate asymptotic resource estimates) and efficient implementations (provably accurate non-asymptotic resource estimates with efficient choices of algorithmic parameters). However, our work does not quite fit into the latter category given the demanding sample complexities we compute in Section 5. We further note that our analysis of the resource requirements for the linear systems problem opens the door for fair comparisons to block-encoding-based methods that promise more optimal dependence on both κ and the approximation error ϵ at the expense of a potentially larger overhead from constructing the block encoding itself. However, as mentioned in Section 1, highly-efficient block encodings can be constructed for classes of structured matrices. Given access to such block encodings, one could consider the Chebyshev approximation to the inverse function rather than a Fourier series approximation. Chebyshev series are known to converge exponentially fast when the function of interest is entire [42], which is the case for the inverse function on the domain $\mathcal{D}_{\kappa^*}$. This in turn implies that the query complexity (with respect to the block encoding) scales logarithmically with the inverse of the desired

approximation error. We leave it to future work to identify classes of matrices for which the superior asymptotic complexity of such block encoding methods provides practical advantages over Fourier series based randomized approaches such as those we consider in this work. Given the resource intensive nature of such randomized quantum algorithms for the QLS problem, it remains unclear which method is actually more practical and useful in the near term.

Acknowledgements

This research was supported by the U.S. Department of Energy (DOE) Office of Science (SC) under Contract No. DEAC02-05CH11231, through the National Energy Research Scientific Computing Center (NERSC), a User Facility located at Lawrence Berkeley National Laboratory, using NERSC award ASCR-ERCAP0033494. S.H. acknowledges funding through the Quantum Information Science Enabled Discovery (QuantISED) for High Energy Physics (KA2401032) program funded by DOE SC and the Quantum Systems Accelerator (QSA), a DOE SC National Quantum Information Science Research Center. R.V.B. acknowledges funding through the DOE SC Accelerated Research in Quantum Computing, Fundamental Algorithmic Research toward Quantum Utility (FAR-Qu) program. The authors thank Niel Van Buggenhout and Yizhi Shen for helpful discussions.

References

[1] Yudong Cao et al. "Quantum Chemistry in the Age of Quantum Computing". In: *Chemical Reviews* 119.19 (Aug. 2019), pp. 10856–10915. ISSN: 1520-6890. DOI: 10.1021/acs.chemrev.8b00803. URL: http://dx.doi.org/10.1021/acs.chemrev.8b00803.

[2] Alberto Di Meglio et al. "Quantum Computing for High-Energy Physics: State of the Art and Challenges". In: *PRX Quantum* 5 (3 Aug. 2024), p. 037001. DOI: 10.1103/PRXQuantum.5.037001. URL: https://link.aps.org/doi/10.1103/PRXQuantum.5.037001.

[3] Christian W. Bauer et al. "Quantum simulation of fundamental particles and forces". In: *Nature Rev. Phys.* 5.7 (2023), pp. 420–432. DOI: 10.1038/s42254-023-00599-8. arXiv: 2404.06298 [hep-ph].

[4] Christian W. Bauer et al. "Quantum Simulation for High-Energy Physics". In: *PRX Quantum* 4.2 (2023), p. 027001. DOI: 10.1103/PRXQuantum.4.027001. arXiv: 2204.03381 [quant-ph].

[5] Yuri Alexeev et al. "Quantum-centric supercomputing for materials science: A perspective on challenges and future directions". In: *Future Generation Computer Systems* 160 (2024), pp. 666–710. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2024.04.060. URL: https://www.sciencedirect.com/science/article/pii/S0167739X24002012.

[6] Ashley Montanaro. "Quantum algorithms: an overview". In: *npj Quantum Information* 2.1 (Jan. 2016). ISSN: 2056-6387. DOI: 10.1038/npjqi.2015.23. URL: http://dx.doi.org/10.1038/npjqi.2015.23.

[7] Mauro E. S. Morales et al. "Quantum Linear System Solvers: A Survey of Algorithms and Applications". In: (Nov. 2024). arXiv: 2411.02522 [quant-ph].

[8] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Phys. Rev. Lett.* 103 (15 Oct. 2009), p. 150502. DOI: 10.1103/PhysRevLett.103.150502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

[9] William Kirby et al. "Quantum Krylov Algorithm for Szegö Quadrature". In: (Sept. 2025). arXiv: 2509.19195 [quant-ph].

[10] András Gilyén et al. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2019. Phoenix, AZ, USA: Association for Computing Machinery, 2019, pp. 193–204. ISBN: 9781450367059. DOI: 10.1145/3313276.3316366. URL: https://doi.org/10.1145/3313276.3316366.

[11] Andrew M. Childs et al. "Theory of Trotter Error with Commutator Scaling". In: *Phys. Rev. X* 11 (1 Feb. 2021), p. 011020. DOI: 10.1103/PhysRevX.11.011020. URL: https://link.aps.org/doi/10.1103/PhysRevX.11.011020.

[12] Earl Campbell. "Random Compiler for Fast Hamiltonian Simulation". In: *Phys. Rev. Lett.* 123 (7 Aug. 2019), p. 070503. DOI: 10.1103/PhysRevLett.123.070503. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.070503.

[13] Dominic W. Berry et al. "Efficient Quantum Algorithms for Simulating Sparse Hamiltonians". In: *Communications in Mathematical Physics* 270.2 (Dec. 2006), pp. 359–371. ISSN: 1432-0916. DOI: 10.1007/s00220-006-0150-x. URL: http://dx.doi.org/10.1007/s00220-006-0150-x.

[14] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. "Hamiltonian Simulation with Nearly Optimal Dependence on all Parameters". In: *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*. FOCS '15. USA: IEEE Computer Society, 2015, pp. 792–809. ISBN: 9781467381918. DOI: 10.1109/FOCS.2015.54. URL: https://doi.org/10.1109/FOCS.2015.54.

[15] Guang Hao Low and Isaac L. Chuang. "Optimal Hamiltonian Simulation by Quantum Signal Processing". In: *Phys. Rev. Lett.* 118 (1 Jan. 2017), p. 010501. DOI: 10.1103/PhysRevLett.118.010501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.118.010501.

[16] Guang Hao Low and Isaac L. Chuang. "Hamiltonian Simulation by Qubitization". In: *Quantum* 3 (July 2019), p. 163. ISSN: 2521-327X. DOI: 10.22331/q-2019-07-12-163. URL: https://doi.org/10.22331/q-2019-07-12-163.

[17] John M. Martyn et al. "Grand Unification of Quantum Algorithms". In: *PRX Quantum* 2 (4 Dec. 2021), p. 040203. DOI: 10.1103/PRXQuantum.2.040203. URL: https://link.aps.org/doi/10.1103/PRXQuantum.2.040203.

[18] Daan Camps et al. "Explicit Quantum Circuits for Block Encodings of Certain Sparse Matrices". In: *SIAM Journal on Matrix Analysis and Applications* 45.1 (2024), pp. 801–827. DOI: 10.1137/22M1484298. eprint: https://doi.org/10.1137/22M1484298. URL: https://doi.org/10.1137/22M1484298.

[19] Christoph Sünderhauf, Earl Campbell, and Joan Camps. "Block-encoding structured matrices for data input in quantum computing". In: *Quantum* 8 (Jan. 2024), p. 1226. ISSN: 2521-327X. DOI: 10.22331/q-2024-01-11-1226. URL: https://doi.org/10.22331/q-2024-01-11-1226.

[20] Christopher F. Kane et al. "Block encoding bosons by signal processing". In: *Quantum* 9 (2025), p. 1747. DOI: 10.22331/q-2025-05-15-1747. arXiv: 2408.16824 [quant-ph].

[21] Siddharth Hariprakash et al. "Strategies for simulating the time evolution of Hamiltonian lattice field theories". In: *Phys. Rev. A* 111.2 (2025), p. 022419. DOI: 10.1103/PhysRevA.111.022419. arXiv: 2312.11637 [quant-ph].

[22] Mason L. Rhodes, Michael Kreshchuk, and Shivesh Pathak. "Exponential Improvements in the Simulation of Lattice Gauge Theories Using Near-Optimal Techniques". In: *PRX Quantum* 5.4 (2024), p. 040347. DOI: 10.1103/PRXQuantum.5.040347. arXiv: 2405.10416 [quant-ph].

[23] Diyi Liu et al. "An efficient quantum circuit for block encoding a pairing Hamiltonian". In: *Journal of Computational Science* 85 (2025), p. 102480. ISSN: 1877-7503. DOI: https://doi.org/10.1016/j.jocs.2024.102480. URL: https://www.sciencedirect.com/science/article/pii/S1877750324002734.

[24] B. David Clader et al. "Quantum Resources Required to Block-Encode a Matrix of Classical Data". In: *IEEE Transactions on Quantum Engineering* 3 (2022), pp. 1–23. DOI: 10.1109/TQE.2022.3231194.

[25] Andrew M. Childs and Nathan Wiebe. "Hamiltonian simulation using linear combinations of unitary operations". In: *Quantum Info. Comput.* 12.11–12 (Nov. 2012), pp. 901–924. ISSN: 1533-7146.

[26] Filippo Della Chiara et al. *Efficient LCU block encodings through Dicke states preparation*. 2025. arXiv: 2507.20887 [quant-ph]. URL: https://arxiv.org/abs/2507.20887.

[27] Etienne Granet and Henrik Dreyer. "Hamiltonian dynamics on digital quantum computers without discretization error". In: *npj Quantum Information* 10.1 (Sept. 2024). ISSN: 2056-6387. DOI: 10.1038/s41534-024-00877-y. URL: http://dx.doi.org/10.1038/s41534-024-00877-y.

[28] Kianna Wan, Mario Berta, and Earl T. Campbell. "Randomized Quantum Algorithm for Statistical Phase Estimation". In: *Phys. Rev. Lett.* 129.3 (2022), p. 030503. DOI: 10.1103/PhysRevLett.129.030503. arXiv: 2110.12071 [quant-ph].

[29] Samson Wang, Sam McArdle, and Mario Berta. "Qubit-Efficient Randomized Quantum Algorithms for Linear Algebra". In: *PRX Quantum* 5 (2 2024), p. 020324. DOI: 10.1103/PRXQuantum.5.020324. arXiv: 2302.01873 [quant-ph].

[30] Shantanav Chakraborty. "Implementing any Linear Combination of Unitaries on Intermediate-term Quantum Computers". In: *Quantum* 8 (2024), p. 1496. DOI: 10.22331/q-2024-10-10-1496. arXiv: 2302.13555 [quant-ph].

[31] Xinzhao Wang et al. "Randomized Quantum Singular Value Transformation". In: (Oct. 2025). arXiv: 2510.06851 [quant-ph].

[32] John M. Martyn and Patrick Rall. "Halving the cost of quantum algorithms with randomization". In: *npj Quantum Inf.* 11.1 (2025), p. 47. DOI: 10.1038/s41534-025-01003-2. arXiv: 2409.03744 [quant-ph].

[33] Yizhi Shen et al. *Quantum Rational Transformation Using Linear Combinations of Hamiltonian Simulations*. 2024. arXiv: 2408.07742 [quant-ph]. URL: https://arxiv.org/abs/2408.07742.

[34] Andrew M. Childs et al. "Toward the first quantum simulation with quantum speedup". In: *Proc. Nat. Acad. Sci.* 115.38 (2018), pp. 9456–9461. DOI: 10.1073/pnas.1801723115.

[35] Jakob Günther et al. *Phase estimation with partially randomized time evolution*. 2025. arXiv: 2503.05647 [quant-ph]. URL: https://arxiv.org/abs/2503.05647.

[36] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision". In: *SIAM J. Comput.* 46.6 (2017), pp. 1920–1950. DOI: 10.1137/16M1087072. arXiv: 1511.02306 [quant-ph].

[37] Lloyd N. Trefethen and J. A. C. Weideman. "The Exponentially Convergent Trapezoidal Rule". In: *SIAM Review* 56.3 (2014), pp. 385–458. DOI: 10.1137/130932132. eprint: https://doi.org/10.1137/130932132. URL: https://doi.org/10.1137/130932132.

[38] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[39] Christopher F. Kane, Siddharth Hariprakash, and Christian W. Bauer. "Obtaining continuum physics from dynamical simulations of Hamiltonian lattice gauge theories". In: (June 2025). arXiv: 2506.16559 [hep-lat].

[40] I. Bogaert. "Iteration-Free Computation of Gauss–Legendre Quadrature Nodes and Weights". In: *SIAM Journal on Scientific Computing* 36.3 (2014), A1008–A1026. DOI: 10.1137/140954969. eprint: https://doi.org/10.1137/140954969. URL: https://doi.org/10.1137/140954969.

[41] Yizhi Shen et al. "Efficient Measurement-Driven Eigenenergy Estimation with Classical Shadows". In: (Sept. 2024). arXiv: 2409.13691 [quant-ph].

[42] Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019. DOI: 10.1137/1.9781611975949. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611975949. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611975949.

[43] Wassily Hoeffding. "Probability Inequalities for Sums of Bounded Random Variables". In: *Journal of the American Statistical Association* 58.301 (1963), pp. 13–30. DOI: 10.1080/01621459.1963.10500830.

[44] Michael Short. "Improved Inequalities for the Poisson and Binomial Distribution and Upper Tail Quantile Functions". In: *International Scholarly Research Notices* 2013.1 (2013), p. 412958. DOI: https://doi.org/10.1155/2013/412958.

A Choosing the number of shots per Hadamard test

Suppose we wish to estimate the following quantity:

$$\langle 0|\, U_\psi^\dagger F(A) U_\phi \,|0\rangle = \sum_{s\in[S]} \alpha_s \langle 0|U_\psi^\dagger e^{-iAt_s} U_\phi|0\rangle. \tag{59}$$

We use the Monte Carlo estimator

$$\hat{Z} = \frac{1}{n}\sum_{i\in[n]} \frac{\alpha_{s_i}}{p_{s_i}} \hat{f}_{s_i}, \tag{60}$$

where each s_i is chosen independently at random with probability $p_s = |\alpha_s| / \sum_{s\in[S]} |\alpha_s|$, and $\hat{f}_{s_i}$ is an unbiased estimator of $f_{s_i} = \langle 0|U_\psi^\dagger e^{-iAt_s} U_\phi|0\rangle$, obtained by averaging $2S$ independent Hadamard test shots, where the factor of 2 comes from the fact that we perform S shots each for the real and imaginary part of each overlap $\langle 0|U_\psi^\dagger e^{-iAt_s} U_\phi|0\rangle$.

The variance of $\hat{Z}$ is

$$\mathrm{Var}[\hat{Z}] = \frac{1}{n}\mathrm{Var}\left(\frac{\alpha_s}{p_s}\hat{f}_s\right) \tag{61}$$

$$= \frac{1}{n}\left(\mathbb{E}_s\left[\frac{|\alpha_s|}{C}\cdot\frac{\alpha_s^2}{p_s^2}\cdot\mathrm{Var}(\hat{f}_s)\right] + \mathrm{Var}_s\left(\frac{\alpha_s}{p_s}f_s\right)\right), \tag{62}$$

where $C = \sum_{s\in[S]} |\alpha_s|$ and $\mathrm{Var}(\hat{f}_s) = \sigma_s^2/S$, with σ_s^2 the variance of a single Hadamard shot for t_s. Since $p_s = |\alpha_s|/C$, we have

$$\frac{\alpha_s^2}{p_s} = C|\alpha_s|.$$

Thus,

$$\mathrm{Var}[\hat{Z}] = \frac{1}{n}\left(C\sum_{s=1}^{S}|\alpha_s|\frac{\sigma_s^2}{S} + C\sum_{s=1}^{S}|\alpha_s|f_s^2 - F(A)^2\right). \tag{63}$$

Suppose we have a fixed total budget $T = nS$ of Hadamard test shots. Solving for n gives $n = T/S$, so we can write the variance as a function of T:

$$\mathrm{Var}[\hat{Z}] = \frac{S}{T}\left[C\sum_{s=1}^{S}|\alpha_s|f_s^2 - F(A)^2\right] + \frac{C}{T}\sum_{s=1}^{S}|\alpha_s|\sigma_s^2 \tag{64}$$

$$= \frac{S}{T}\mathrm{Var}_s\left(\alpha_s f_s\right) + \frac{C}{T}\sum_{s=1}^{S}|\alpha_s|\sigma_s^2, \tag{65}$$

where we note that both terms decrease with T and the second term is independent of S. Since we must have $S \geq 1$, the variance is minimized by choosing $S = 1$ and $n = T$.

B Taylor expansion of the time evolution operator

In this section, we follow the approach of [28] to expand the time evolution operator as an infinite LCU. Given the exponential $e^{-\mathrm{i}\tilde{A}\tau/r}$, we can Taylor expand as follows:

$$e^{-\mathrm{i}\tilde{A}\tau/r} = \sum_{n=0}^{\infty} \frac{(\mathrm{i}\tilde{A}\tau/r)^n}{n!} = \sum_{\substack{n=0 \\ n \text{ even}}}^{\infty} \frac{(\mathrm{i}\tilde{A}\tau/r)^n}{n!} \left(I + \frac{\mathrm{i}\tilde{A}\tau/r}{n+1} \right), \tag{66}$$

where I is the identity operator. We then use the Pauli decomposition shown in Eq. (1) to write

$$e^{-\mathrm{i}\tilde{A}\tau/r} = \sum_{\substack{n=0 \\ n \text{ even}}}^{\infty} \frac{1}{n!} \left(\frac{\mathrm{i}\tau}{r} \right)^n \left(\sum_{\ell \in [L]} \frac{c_\ell}{\lambda} P_\ell \right)^n \left(I + \frac{\mathrm{i}\tau/r}{n+1} \sum_{\ell \in [L]} \frac{c_\ell}{\lambda} P_\ell \right) \tag{67}$$

$$= \sum_{\substack{n=0 \\ n \text{ even}}}^{\infty} \frac{1}{n!} \left(\frac{\mathrm{i}\tau}{r} \right)^n \left(\sum_{\ell \in L} \frac{c_\ell}{\lambda} P_\ell \right)^n \sum_{\ell \in L} \frac{c_\ell}{\lambda} \left(I + \frac{\mathrm{i}\tau/r}{n+1} P_\ell \right), \tag{68}$$

where we have used the fact that $\tilde{A}$ has unit Pauli weight. We now define the angle θ_n such that

$$\cos(\theta_n) = \frac{1}{\sqrt{1 + \left(\frac{\tau/r}{n+1} \right)^2}}, \quad \sin(\theta_n) = \frac{\frac{\tau/r}{n+1}}{\sqrt{1 + \left(\frac{\tau/r}{n+1} \right)^2}}, \tag{69}$$

and use this to rewrite Eq. (68) as follows:

$$e^{-\mathrm{i}\tilde{A}\tau/r} = \sum_{\substack{n=0 \\ n \text{ even}}}^{\infty} \frac{1}{n!} \left(\frac{\mathrm{i}\tau}{r} \right)^n \sqrt{1 + \left(\frac{\tau/r}{n+1} \right)^2} \left(\sum_{\ell \in [L]} \frac{c_\ell}{\lambda} P_\ell \right)^n \sum_{\ell \in [L]} \frac{c_\ell}{\lambda} \left(\cos(\theta_n) I + \mathrm{i} \sin(\theta_n) P_\ell \right) \tag{70}$$

$$= \sum_{\substack{n=0 \\ n \text{ even}}}^{\infty} \frac{1}{n!} \left(\frac{\mathrm{i}\tau}{r} \right)^n \sqrt{1 + \left(\frac{\tau/r}{n+1} \right)^2} \left(\sum_{\ell \in [L]} \frac{c_\ell}{\lambda} P_\ell \right)^n \sum_{\ell \in [L]} \frac{c_\ell}{\lambda} e^{i\theta_n P_\ell}, \tag{71}$$

C Proofs of Theorems

C.1 Proof of Theorem 4

Proof. We use a result found in the proof of Lemma 12 in [36] where the authors show that

$$\left| \frac{i}{\sqrt{2\pi}} \int_0^{y_{\max}} dy \int_{-z_{\max}}^{z_{\max}} dz \left(z e^{-z^2/2} e^{-ixyz} \right) - \frac{1}{x} \right| \leq \frac{e^{-(xy_{\max})^2/2}}{|x|} + \frac{2e^{-z^2/2}}{|x|}. \tag{72}$$

We bound each of the terms on the RHS separately by ϵ_1 and ϵ_2 respectively, such that these satisfy $\epsilon_1 + \epsilon_2 \geq \epsilon_T$. Since we have that $|x| \geq 1/\kappa$.

1. Bounding the first term, we see that:

$$\frac{e^{-(xy_{\max})^2/2}}{|x|} \leq \kappa e^{-y_{\max}^2/2\kappa^2} \leq \epsilon_1 \tag{73}$$

and thus

$$y_{\max} \geq \kappa \sqrt{2 \ln \left(\frac{\kappa}{\epsilon_1} \right)} \tag{74}$$

2. Bounding the second term, we see that:

$$\frac{2e^{-z^2/2}}{|x|} \leq 2\kappa e^{-z_{\max}^2/2} \leq \epsilon_2 \tag{75}$$

and thus

$$z_{\max} \geq \sqrt{2 \ln \left(\frac{2\kappa}{\epsilon_2} \right)} \tag{76}$$

Since any allowed ϵ_1, ϵ_2 will work, we set $\epsilon_1 = \epsilon_T/3$ and $\epsilon_2 = 2\epsilon_T/3$ to arrive at the desired result. □

C.2 Proof of Theorem 5

Proof. We begin the proof by first discussing the relevant error bounds we use to characterize the total discretization error ϵ_D. For the z-integral appearing in Eq. (19), we use the trapezoidal rule. The error of the trapezoidal rule with step Δz over the integration range $[-\infty, \infty]$ for an integrand $g(\cdot)$, which we denote by the functional $e_{\text{trap}}^{(\Delta z)}[g]$, can be upper bounded provided that g is analytic within and on the horizontal strip of width 2ρ in the complex plane, i.e., $\mathcal{H}_\rho = \{z : |\Im z| \leq \rho\}$. The bound is given by [37],

$$e_{\text{trap}}^{(\Delta z)}[g|\mathcal{H}_\rho] \leq \frac{2 \sup_{s \in [-\rho, \rho]} \left(\int_{-\infty}^{\infty} dx |g(x + \mathrm{i}s)| \right)}{e^{2\pi\rho/\Delta z} - 1}. \tag{77}$$

In our case, the z-integrand, $ze^{-z^2/2}e^{-\mathrm{i}xyz}$, is holomorphic on the entire complex plane and hence Eq. (77) applies for all $\rho > 0$.

We can discretize the y-integral appearing in Eq. (19) using the Gauss-Legendre scheme. We denote the error accrued over the integration range $[-1, 1]$ by the Gauss-Legendre discretization scheme with J quadrature nodes for a function $h(\cdot)$ by the functional $e_{GL}^{(J)}[h]$. We use the bound presented in [33] to upper bound this error as follows:

$$e_{\text{GL}}^{(J)}[h|\mathcal{E}_\sigma] \leq \frac{8\sigma^2 \sup_{t \in \mathcal{E}_\sigma} |h(t)|}{\sigma^{2J}(\sigma^2 - 1)}, \tag{78}$$

provided that h is analytic within and on the Bernstein σ-ellipse in the complex plane $\mathcal{E}_\sigma = \{\sigma e^{i\theta}/2 + e^{-i\theta}/2\sigma : \theta \in [-\pi, \pi]\}$. We note that the integrand for the y-integral, $e^{-\mathrm{i}xyz}$, appearing in Eq. (19) is holomorphic on the entire complex plane and hence Eq. (78) applies for all $\sigma > 1$.

Thus, the total discretization error, as a result of using the trapezoidal and Gauss-Legendre rules for the z, y integrals in Eq. (19) respectively, can be upper bounded using the triangle inequality as:

$$\begin{aligned} \left| f_{J,K,y_{\max},z_{\max}}^{(\epsilon_D,\epsilon_T)}(x) - \frac{1}{x} \right| \leq \epsilon_D \leq \frac{1}{\sqrt{2\pi}} \int_0^{y_{\max}} dy \left(\max_{0 \leq n \leq N-1} e_{\text{trap}}^{(\Delta z)} \left[g_n(y, z) | \mathcal{H}_\rho \right] \right) + \\ \frac{y_{\max} \Delta z}{2\sqrt{2\pi}} \sum_{k=0}^{K-1} z_k e^{-z_k^2/2} \left(\max_{0 \leq n \leq N-1} e_{GL}^{(J)} \left[h_{nk}(y) | \mathcal{E}_\sigma \right] \right), \end{aligned} \tag{79}$$

where N denotes the dimension of the matrix under consideration (which we will denote by A), $g_n(y, z) = ze^{-z^2/2}e^{-ix_n yz}$ with $x_n \in \text{spec}(A)$, $h_{nk}(y) = e^{-ix_n z_k y_{\max}(1+y)/2}$. We note that the maximum

error from each source occurs when $|-x_n|$ is maximum, i.e. when $|x_n| = 1$. Thus, Eq. (79) becomes

$$\begin{aligned}\epsilon_D \leq \frac{1}{\sqrt{2\pi}} \int_0^{y_{\max}} dy \left(e_{\text{trap}}^{(\Delta z)} \left[z e^{-z^2/2} e^{iyz} | S_\rho \right] \right) + \\ \frac{y_{\max} \Delta z}{2\sqrt{2\pi}} \sum_{k=0}^{K-1} z_k e^{-z_k^2/2} \left(e_{\text{GL}}^{(J)} \left[e^{i z_k y_{\max}(1+y)/2} | \mathcal{H}_\sigma \right] \right),\end{aligned} \tag{80}$$

We now compute each term separately (which we call $\epsilon_{(D,\text{trap})}$ and $\epsilon_{(D,\text{GL})}$ respectively). We first consider $\epsilon_{(D,\text{trap})}$ and compute $e_{\text{trap}}^{(\Delta z)}$ which, using Eq. (77), is bounded by:

$$e_{\text{trap}}^{(\Delta z)} \left[z e^{-z^2/2} e^{iyz} | S_\rho \right] \leq \frac{2 \sup_{s \in [-\rho,\rho]} \left(\int_{-\infty}^{\infty} dz \left| (z+is) e^{-(z+is)^2/2} e^{i(z+is)y} \right| \right)}{e^{2\pi\rho/\Delta z} - 1}. \tag{81}$$

We compute an upper bound on the supremum as follows:

$$\sup_{s \in [-\rho,\rho]} \left(\int_{-\infty}^{\infty} dz \left| (z+is) e^{-(z+is)^2/2} e^{i(z+is)y} \right| \right) \leq \sup_{s \in [-\rho,\rho]} \left(\int_{-\infty}^{\infty} dz \left| (z+is) e^{-z^2/2} e^{s^2/2} e^{-sy} \right| \right) \leq \tag{82}$$

$$\sup_{s \in [-\rho,\rho]} \left(e^{s^2/2} e^{-sy} \left(\int_{-\infty}^{\infty} dz \left| z e^{-z^2/2} \right| + |s| \int_{-\infty}^{\infty} dz \left| e^{-z^2/2} \right| \right) \right) = \tag{83}$$

$$\sup_{s \in [-\rho,\rho]} \left(e^{s^2/2} e^{-sy} \left(2 + |s| \sqrt{2\pi} \right) \right) \leq e^{\rho^2/2} e^{\rho y} \left(2 + \rho\sqrt{2\pi} \right) \tag{84}$$

Thus, we have that

$$e_{\text{trap}}^{(\Delta z)} \left[z e^{-z^2/2} e^{iyz} | \mathcal{H}_\rho \right] \leq \frac{2 e^{\rho^2/2} e^{\rho y} (2 + \rho\sqrt{2\pi})}{e^{\pi\rho(K-1)/z_{\max}} - 1}, \tag{85}$$

where we have used that $\Delta z = 2 z_{\max}/(K-1)$. We thus see that

$$\epsilon_{(D,\text{trap})} \leq \frac{1}{\sqrt{2\pi}} \frac{2 e^{\rho^2/2} (2 + \rho\sqrt{2\pi})}{e^{\pi\rho(K-1)/z_{\max}} - 1} \int_0^{y_{\max}} dy e^{\rho y} \tag{86}$$

$$\leq \frac{1}{\rho\sqrt{2\pi}} \frac{2 e^{\rho^2/2} e^{\rho y_{\max}} (2 + \rho\sqrt{2\pi})}{e^{\pi\rho(K-1)/z_{\max}}}, \tag{87}$$

where we have assumed that ρ, K, and $z_{\max}$ are independent of y. Next, we consider the term $\epsilon_{(D,\text{GL})}$ and compute $e_{\text{GL}}^{(J)}$ which, using Eq. (78), is bounded by:

$$e_{\text{GL}}^{(J)} \left[e^{i z_k y_{\max}(1+y)/2} | \mathcal{E}_\sigma \right] \leq \frac{8\sigma^2 \sup_{t \in \mathcal{E}_\sigma} \left| e^{i z_k y_{\max}(1+t)/2} \right|}{\sigma^{2J} (\sigma^2 - 1)}. \tag{88}$$

We can compute an upper bound on the supremum as follows:

$$\sup_{\theta \in [-\pi,\pi]} \left| \exp\left(\frac{i z_k y_{\max}}{2} \left(1 + \frac{\sigma e^{i\theta}}{2} + \frac{e^{-i\theta}}{2\sigma} \right) \right) \right| \leq \left| \exp\left(\frac{i z_k y_{\max}}{2} \left(1 - \frac{\sigma i}{2} - \frac{i}{2\sigma} \right) \right) \right| \tag{89}$$

$$\leq \exp\left(\frac{z_k y_{\max}}{4} \left(\sigma + \frac{1}{\sigma} \right) \right). \tag{90}$$

Thus

$$e_{\mathrm{GL}}^{(J)}\left[e^{iz_k y_{\max}(1+y)/2}|\mathcal{E}_\sigma\right] \leq \frac{8\sigma^2 e^{\frac{z_k y_{\max}}{4}\left(\sigma+\frac{1}{\sigma}\right)}}{\sigma^{2J}(\sigma^2-1)}. \tag{91}$$

The expression for $\epsilon_{(D,\mathrm{GL})}$ is now upper bounded as follows:

$$\epsilon_{(D,\mathrm{GL})} \leq \frac{y_{\max}\Delta z}{2\sqrt{2\pi}} \sum_{k=0}^{K-1} \frac{8 z_k e^{-z_k^2/2}\sigma^2 e^{\frac{z_k y_{\max}}{4}\left(\sigma+\frac{1}{\sigma}\right)}}{\sigma^{2J}(\sigma^2-1)} \leq \frac{8 y_{\max}\Delta z K z_{\max}\sigma^2 e^{\frac{z_{\max} y_{\max}}{4}\left(\sigma+\frac{1}{\sigma}\right)}}{2\sqrt{2\pi}\sigma^{2J}(\sigma^2-1)} \tag{92}$$

Putting this all together, we have that

$$\epsilon_D \leq \frac{1}{\rho\sqrt{2\pi}} \frac{2e^{\rho^2/2}e^{\rho y_{\max}}(2+\rho\sqrt{2\pi})}{e^{\pi\rho(K-1)/z_{\max}}} + \left(\frac{K}{K-1}\right)\frac{8 y_{\max} z_{\max}^2 \sigma^2 e^{\frac{z_{\max} y_{\max}}{4}\left(\sigma+\frac{1}{\sigma}\right)}}{\sqrt{2\pi}\sigma^{2J}(\sigma^2-1)}. \tag{93}$$

for all valid choices of the quantities ρ, σ. To present efficiently computable lower bounds on the Fourier parameters J, K, we can separately control the errors due to each of these terms as follows: Let ϵ', ϵ'' denote upper bounds on each of terms (respectively) appearing on the RHS of Eq. (93). We thus see that

$$K \geq 1 + \frac{z_{\max}}{\pi\rho}\left(\ln(2) + \frac{\rho^2}{2} + \rho y_{\max} + \ln\left(\frac{1}{\epsilon'}\right) + \ln\left(1 + \frac{2}{\rho\sqrt{2\pi}}\right)\right) \tag{94}$$

$$J \geq \frac{1}{2\ln(\sigma)}\left(\ln\left(\frac{K}{K-1}\right) + \ln\left(\frac{8 y_{\max} z_{\max}^2 \sigma^2}{\epsilon''(\sigma^2-1)\sqrt{2\pi}}\right) + \frac{z_{\max} y_{\max}}{4}\left(\sigma + \frac{1}{\sigma}\right)\right). \tag{95}$$

We consider the balanced case of $\epsilon' = \epsilon'' = \epsilon_D/2$, set $(\rho, \sigma) = (z_{\max}, \sqrt{2})$, and use Eq. (17) to arrive at the desired result. □

C.3 Proof of Lemma 6

Let A be a Hermitian matrix with condition number κ. Let λ denote the Pauli weight of A. We now consider the case when $|\max(\mathrm{spec}(A))| \geq 1$. We note that this automatically implies that $|\min(\mathrm{spec}(A))| \geq 1/\kappa$ since the ratio of the two quantities determines the condition number κ. Since it is true that $1 \leq ||A|| \leq \lambda$, we see that:

$$|\max(\mathrm{spec}(\tilde{A}))| \leq 1, \tag{96}$$

$$|\min(\mathrm{spec}(\tilde{A}))| \geq \frac{1}{\lambda\kappa}. \tag{97}$$

Hence $\mathrm{spec}(\tilde{A}) \subseteq \mathcal{D}_{\lambda\kappa} = \mathcal{D}_{\tilde{\kappa}}$.

C.4 Proof of Lemma 7

From Eqs. (20) and (23), we have that:

$$N_y = \frac{y_{\max}}{2\sqrt{2\pi}} \sum_{j\in[J]} \left|\frac{2}{(1-y_j^2)(P_J'(y_j))^2}\right| \equiv \frac{y_{\max}}{2\sqrt{2\pi}} \sum_{j\in[J]} \mathrm{GL}_j, \tag{98}$$

where $\mathrm{GL_j}$ denotes the usual Gauss-Legendre quadrature weight. We note the these quadrature weights are always positive. To evaluate this sum, consider a function $f(x) : \mathbb{R} \to \mathbb{R}$ that is a polynomial of degree d such that $d \leq 2J-1$. Then it is true that

$$\int_{-1}^{1} f(x)dx = \sum_{j\in[J]} \mathrm{GL_j} f(x_j), \tag{99}$$

where as stated in the main text, x_j is the j^{th} root of the degree J Legendre polynomial. In particular, we let $f(x) = 1$, then we have that

$$\sum_{j\in[J]} \mathrm{GL_j} = 2, \tag{100}$$

and hence

$$N_y = \frac{y_{\max}}{\sqrt{2\pi}}. \tag{101}$$

We may then use Theorem 4 to arrive at the upper bound for N_y.

For the upper bound on N_z, from Eq. (23) we have that

$$N_z = \frac{2z_{\max}}{K-1} \sum_{k\in[K]} \left| z_k e^{-z_k^2/2} \right| \leq \frac{2z_{\max}^2}{K-1} \sum_{k\in[K]} e^{-z_k^2/2} \tag{102}$$

$$\approx \frac{2z_{\max}^2}{K-1} \int_{-z_{\max}}^{z_{\max}} e^{-z^2/2} dz \leq \frac{4z_{\max}^2}{K-1} \int_0^{\infty} e^{-z^2/2} dz \tag{103}$$

$$= \frac{2z_{\max}^2\sqrt{2\pi}}{K-1}. \tag{104}$$

We can then use Theorem 4 to arrive at the upper bound for N_z.

C.5 Proof of Theorem 9

From Eq. (37), we have that

$$|B_{j',k'}| = \left| \frac{\langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}) |\psi\rangle}{\lambda} - \mathbb{E}[Z_{j',k'}] \right| \tag{105}$$

$$= \left\| \frac{\mathrm{i}}{\lambda\sqrt{2\pi}} \sum_{j\in[J]} \sum_{k\in[K]} \left(w_y^{(j)} w_z^{(k)} z_k e^{-z_k^2/2} e^{-\mathrm{i}\tilde{A} y_j z_k} \right) - \frac{\mathrm{i}}{\lambda\sqrt{2\pi}} \sum_{j\in[J]} \sum_{k\in[K]} \left(w_y^{(j)} w_z^{(k)} z_k e^{-z_k^2/2} S_{r_{\mathrm{PF}}}(t_{j'k'}) \right) \right\| \tag{106}$$

$$\leq \left| \frac{\mathrm{i}}{\lambda\sqrt{2\pi}} \sum_{j\in[J]} \sum_{k\in[K]} \left(w_y^{(j)} w_z^{(k)} z_k e^{-z_k^2/2} \right) \right| \max_{(j,k)\in[J]\times[K]} \left(\left\| e^{-\mathrm{i}\tilde{A} y_j z_k} - S_{r_{\mathrm{PF}}}(t_{j'k'}) \right\| \right) \tag{107}$$

$$\leq \frac{N_y N_z \epsilon_{\mathrm{PF}}}{\lambda}, \tag{108}$$

where in the final inequality we have used Eqs. (23) and (31).

C.6 Proof of Theorem 10

We use the triangle inequality to write:

$$\left| Z_{N_{\mathrm{S}}} - \lambda^{-1} \langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}) |\psi\rangle \right| \leq \left| \Re\left(Z_{N_{\mathrm{S}}}\right) - \lambda^{-1} \Re\left(\langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}) |\psi\rangle \right) \right| + \tag{109}$$

$$\left| \Im\left(Z_{N_{\mathrm{S}}}\right) - \lambda^{-1} \Im\left(\langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}) |\psi\rangle \right) \right|. \tag{110}$$

Thus, we can separately bounding each term on the RHS of this inequality by $\epsilon_S/2$. In particular, by individually approximating the real and imaginary parts of the desired overlap $\lambda^{-1} \langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A}) |\psi\rangle$ to

error $\epsilon_S/2$ with probability at least $1-\delta$, we will have effectively constructed an ϵ_S-close approximation to $\lambda^{-1}\langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A})|\psi\rangle$ also with probability at least $1-\delta$. We first consider the task of constructing an approximation to the real part.

We note that

$$|Z_{jk}| = \frac{N_y N_z \sqrt{2}}{\lambda}, \tag{111}$$

and hence

$$\frac{-N_y N_z \sqrt{2}}{\lambda} \leq \Re\left(N_{jk}\right) \leq \frac{N_y N_z \sqrt{2}}{\lambda}. \tag{112}$$

We also have that:

$$\begin{aligned}
&|\Re\left(Z_{N_S}\right) - \lambda^{-1}\Re\left(\langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A})|\psi\rangle\right)| \\
&\quad\leq |\Re\left(Z_{N_S}\right) - \Re\left(\mathbb{E}\left[Z_{jk}\right]\right)| + |\Re\left(\mathbb{E}\left[Z_{jk}\right]\right)| - \lambda^{-1}\Re\left(\langle\phi| I_D^{(\epsilon_D,\epsilon_T)}(\tilde{A})|\psi\rangle\right)| \\
&\quad\leq |\Re\left(Z_{N_S}\right) - \Re\left(\mathbb{E}\left[Z_{jk}\right]\right)| + \frac{B_{\mathrm{PF}}}{2},
\end{aligned}$$

where we have split the contribution of the bias term presented in Eq. (37) equally between the real and imaginary parts of the overlap. By Hoeffding's inequality [43], we thus have that:

$$\mathrm{prob}\left(|\Re\left(Z_{N_S}\right) - \Re\left(\mathbb{E}\left[Z_{jk}\right]\right)| \geq \frac{\epsilon_S - B_{\mathrm{PF}}}{2}\right) = \delta \leq 2\exp\left(\frac{-N_S(\epsilon_S - B_{\mathrm{PF}})^2\lambda^2}{8N_y^2 N_z^2}\right) \tag{113}$$

$$\implies N_S \leq \frac{8N_y^2 N_z^2}{(\epsilon_S - B_{\mathrm{PF}})^2\lambda^2}\ln\left(\frac{2}{\delta}\right). \tag{114}$$

Thus to succeed with probability $1-\delta$ in approximating the real part of the desired overlap, we must take N_S to be at least

$$N_S = 1 + \frac{8N_y^2 N_z^2}{(\epsilon_S - B_{\mathrm{PF}})^2\lambda^2}\ln\left(\frac{2}{\delta}\right). \tag{115}$$

At this juncture, we note that the analysis for the imaginary part is identical and thus, to approximate both the real and imaginary parts of the desired overlap we must take N_S to be at least:

$$N_S = 2 + \frac{16N_y^2 N_z^2}{(\epsilon_S - B_{\mathrm{PF}})^2\lambda^2}\ln\left(\frac{2}{\delta}\right). \tag{116}$$

We then set $\epsilon_S = \epsilon/2$, and along with the assumption that $\epsilon_D + \epsilon_T \leq \epsilon/2$ (balanced choice made in Section 2.2), we arrive at the desired result for N_S. To arrive at the result for N_{CP}, we note that upon constructing a PF approximation of a single exponential, the controlled version of the unitary can be implemented by individually controlling each of the exponentials appearing in the decomposition. N_{CP} is then given by the total number of terms appearing in the PF decomposition and thus:

$$N_{\mathrm{CP}} = 2r_{\mathrm{PF}}L. \tag{117}$$

C.7 Proof of Theorem 11

From Appendix B, we have that the exponential $e^{-i\tilde{A}\tau/r}$ can be expressed as an infinite LCU:

$$e^{-i\tilde{A}\tau/r} = \sum_{m\in\mathcal{M}'_\infty} d'_{m\tau}U'_{m\tau}, \tag{118}$$

where $\mathcal{M}'_\infty$ denotes the appropriately defined infinite dimensional index set and each coefficient $d'_{m\tau} \in \mathbb{R}^+$ by absorbing the phases into the unitaries $U'_{m\tau}$. Eq. (45) represents an approximation to this infinite LCU by truncating the infinite dimensional index set $\mathcal{M}_\infty$ and replacing it with the finite dimensional set $\mathcal{M}'$. Using the notation from Eq. (25), the quantity $|B_{\text{RTE}}|$ can thus be upper bounded as

$$|B_{\text{RTE}}| \leq \sum_{s\in[S]} |\alpha_s| \left\| \left(\sum_{m\in\mathcal{M}'_\infty} d'_{mt_s} U'_{mt_s} \right)^r - \left(\sum_{m\in\mathcal{M}'} d'_{mt_s} U_{mt_s} \right)^r \right\|, \tag{119}$$

where have assumed a choice for r independent of the Fourier index s[7]. From [28] we have that

$$\begin{aligned} &\left\| \left(\sum_{m\in\mathcal{M}'_\infty} d'_{mt_s} U'_{mt_s} \right)^r - \left(\sum_{m\in\mathcal{M}'} d'_{mt_s} U_{mt_s} \right)^r \right\| \\ &\qquad \leq \sum_{i\in[r]} \left\| \sum_{m\in\mathcal{M}'_\infty} d'_{mt_s} U'_{mt_s} \right\|^k \left\| \sum_{m\in\mathcal{M}'_\infty/\mathcal{M}'} d'_{mt_s} U'_{mt_s} \right\| \left\| \sum_{m\in\mathcal{M}'} d'_{mt_s} U'_{mt_s} \right\|^{r-k} \qquad (120) \\ &\qquad \leq r \left\| \sum_{m\in\mathcal{M}'_\infty} d'_{mt_s} U'_{mt_s} \right\|^r \left\| \sum_{m\in\mathcal{M}'_\infty/\mathcal{M}'} d'_{mt_s} U'_{mt_s} \right\|, \qquad (121) \end{aligned}$$

and hence the LHS of Eq. (119) can be upper bounded as

$$|B_{\text{RTE}}| \leq \sum_{s\in[S]} |\alpha_s|\, r \left(\sum_{m\in\mathcal{M}'_\infty} d'_{mt_s} \right)^r \sum_{m\in\mathcal{M}'_\infty/\mathcal{M}'} d'_{mt_s}. \tag{122}$$

From Eq. (71) and the proof of Proposition 13 from [28], we have that

$$\begin{aligned} \sum_{m\in\mathcal{M}'_\infty/\mathcal{M}'} d'_{mt_s} &\leq \sum_{n=n_{\max}+1}^{\infty} \frac{1}{(2n)!} \left(\frac{t_s}{r}\right)^{2n} \sqrt{1 + \left(\frac{t_s}{r(2n+1)}\right)^2} \qquad (123) \\ &\leq e^{|t_{\max}|/r} \sum_{n=n_{\max}+1}^{\infty} \frac{1}{n!} \left(\frac{|t_s|}{r}\right)^n e^{-|t_s|/r} \qquad (124) \end{aligned}$$

We can then use the assumption that $r \geq |t_s| \;\; \forall s \in [S]$ thus allowing us to use Corollary 6 from [44] (restated in the proof of Proposition 4 from [28], specifically Equation A4) and write

$$e^{|t_{\max}|/r} \sum_{n=n_{\max}+1}^{\infty} \frac{1}{n!} \left(\frac{|t_s|}{r}\right)^n e^{-|t_s|/r} \leq \frac{e^{(|t_{\max}|-|t_{\min}|)/r}}{2} \left(\frac{e t_{\max}}{r n_{\max}}\right)^{n_{\max}}. \tag{125}$$

Thus we have that

$$\begin{aligned} |B_{\text{RTE}}| &\leq \frac{e^{(|t_{\max}|-|t_{\min}|)/r}}{2} \left(\frac{e t_{\max}}{r n_{\max}}\right)^{n_{\max}} \sum_{s\in[S]} |\alpha_s|\, r \left(\sum_{m\in\mathcal{M}'_\infty} d'_{mt_s} \right)^r \qquad (126) \\ &\leq \frac{r N_y N_z e^{(t_{\max}^2+|t_{\max}|-|t_{\min}|)/r}}{2} \left(\frac{e t_{\max}}{r n_{\max}}\right)^{n_{\max}} \qquad (127) \end{aligned}$$

[7] We note that our proof can be generalized to the case where r has an explicit dependence on the index variable s

C.8 Proof of Theorem 12

The proof for the result for N_{S} is identical to that shown in Appendix C.6 for the PF subroutine with the following simple changes (see Eq. (48)):

$$N_y N_z \to N_{\mathrm{RTE}} N_y N_z \leq e^{t_{\max}^2/r} N_y N_z, \tag{128}$$

$$B_{\mathrm{PF}} \to B_{\mathrm{RTE}}. \tag{129}$$