

Probabilistic quantum algorithm for Lyapunov equations and matrix inversion

Marcello Benedetti^{*,†}, Ansis Rosmanis[‡], and Matthias Rosenkranz[‡]
Quantinuum, Partnership House, Carlisle Place, London SW1P 1BX, United Kingdom

 (Received 29 August 2025; revised 11 February 2026; accepted 9 March 2026; published 16 April 2026)

We present a probabilistic quantum algorithm for preparing mixed states which, in expectation, are proportional to the solutions of Lyapunov equations—linear matrix equations ubiquitous in the analysis of classical and quantum dynamical systems. Building on previous results by Zhang *et al.*, [arXiv:2304.04526](#), at each step the algorithm can (i) return the current state, (ii) apply a trace nonincreasing completely positive map, or (iii) restart. We introduce a deterministic stopping rule, which leads to an efficient algorithm with a bounded expected number of calls to oracles representing the two input matrices of the Lyapunov equations. We also consider preparing a mixed state that approximates the normalized inverse of a positive definite matrix A . In its most general form, the algorithm generates mixed states, which approximate matrix-valued weighted sums and integrals. It can be shown that block encodings and states yield two incomparable computational resources even when they represent the same piece of data. While block encodings of functions have received much attention in the literature, our work takes a step toward the less explored problem of encoding functions into mixed states.

DOI: [10.1103/mprk-9vx3](#)

I. INTRODUCTION

We present a probabilistic quantum algorithm that solves several linear matrix equations under certain conditions in the following sense. Given an $N \times N$ normal matrix A and $N \times N$ positive semidefinite matrix B , in expectation the algorithm generates mixed states approximately proportional to solutions X of the *Lyapunov equations*

$$AXA^\dagger - X + B = 0 \quad (\text{discrete time}) \quad (1)$$

$$AX + XA^\dagger + B = 0 \quad (\text{continuous time}) \quad (2)$$

with bounded expected stopping time and guarantees on the approximation error. We also discuss an application of the algorithm to finding solutions of

$$AX = I \quad (\text{matrix inversion}) \quad (3)$$

with error ϵ in trace distance and expected stopping time $\lceil \kappa \ln(1/\epsilon) \rceil + 1$ for positive definite A with condition number κ and I the identity.

^{*}Contact author: marcello.benedetti@quantinuum.com

[†]Contact author: ansis.rosmanis@quantinuum.com

[‡]Contact author: matthias.rosenkranz@quantinuum.com

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

Lyapunov equations are foundational to control and systems theory, fields in engineering concerned with controlling and predicting dynamical systems (see, e.g., Refs. [1,2] and references therein). Consider a continuous-time, linear dynamical system $\dot{\mathbf{x}} = A\mathbf{x}$ with $\mathbf{x}$ the system state. More general dynamical systems include driving terms and outputs. If this system is stable [here, $\mathbf{x}(t) \rightarrow 0$ for $t \rightarrow \infty$], a solution to the Lyapunov equation exists for positive semidefinite B . Then functions of the form $\mathbf{x}^T X \mathbf{x}$ characterize energy transfer and the approach to equilibrium. Lyapunov equations also determine the time evolution of covariance matrices of linear open quantum systems [3,4]. Furthermore, their solutions appear in the calculation of the Fréchet derivatives of the matrix square root function [5], and in quantum information as the symmetric logarithmic derivative used in the quantum Fisher information matrix and Bures metric [6,7].

A conceptually simple way to solve either of the Lyapunov equations is to map it to a system of N^2 linear equations $\mathcal{A}\mathbf{x} = \mathbf{b}$ and find an approximate solution $\tilde{\mathbf{x}}$ with $\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 < \epsilon$. For general or sparse $\mathcal{A}$, this is impractical for large N owing to a poly(N) runtime of any classical solver. Specialized solvers, such as the Bartels-Stewart algorithm [8], or convex optimization [9] polynomially reduce the complexity but remain restricted to moderate N . Approximate methods for larger-scale systems include singular value decomposition-based methods and Krylov methods [1,10]. Alternatively, one can use a quantum linear system solver to obtain a state $|\tilde{\mathbf{x}}\rangle$ with $\|\tilde{\mathbf{x}} - |\mathbf{x}\rangle\|_2 < \epsilon$ for $|\mathbf{x}\rangle \propto \sum_k x_k |k\rangle$ [11]. Given an oracle that block encodes $\mathcal{A}$ with $\|\mathcal{A}\| = 1$ and condition

number κ , and an oracle preparing $|\mathbf{b}\rangle$, the best currently known quantum linear system solvers achieve query complexity $\mathcal{O}[\kappa \log(1/\epsilon)]$ [12,13]. Thus, if one can construct the block encoding and state-preparation circuits with $\text{polylog}(N)$ gates, preparing the state $|\tilde{\mathbf{x}}\rangle$ has runtime $\mathcal{O}[\kappa \log(1/\epsilon) \text{polylog}(N)]$. This approach has the potential for a large scaling advantage in N in applications that only require estimating few expectation values rather than reading out the full state $\tilde{\mathbf{x}}$. For a recent review of quantum linear system solvers, see Ref. [14]. A quantum algorithm for the continuous-time Lyapunov equation was proposed in Ref. [15]. This equation is also a special case of the algebraic Riccati equation, for which a quantum algorithm is given in Ref. [16], and of the Sylvester equation, for which a quantum algorithm appeared very recently [17]. These approaches output a block encoding of the subnormalized solution matrix.

In contrast to previous work, here we want to produce quantum states proportional to the solution of the matrix equation, i.e., $\rho \propto X$. This is interesting because, in general, a state and a block encoding for the same piece of data provide incomparable computational resources. We prove this in Appendix A by adapting a result from Ref. [17]. Our algorithm builds on the work by Zhang *et al.* [18]. Their algorithm generates states via a stochastic process $\{\rho_t\}$, which depends on the outcomes of both a classical coin flip and a quantum measurement of an ancilla register. Coin flips determine whether the process stops, while measurements determine whether the desired completely positive (CP) map $\mathcal{E}$ acting on the main register is correctly implemented. The stopped process generates states proportional to $\sum_{k=0}^{\infty} c_k \mathcal{E}^k(I/N)$ in expectation, where c_k depends on the coin-flip probabilities $\{r_j\}_{j=0}^k$ for all k and $\mathcal{E}^k$ means applying $\mathcal{E}$ k times. For a suitable choice of $\mathcal{E}$ and $\{r_k\}$, this process returns an approximate quantum Gibbs state in expectation.

We modify the algorithm to generate states proportional to $\sum_{k=0}^T c_k \mathcal{E}^k(\rho_0)$ with an arbitrary initial state ρ_0 . The main difference to Ref. [18] is the newly introduced parameter T and different choices of $\mathcal{E}$, $\{c_k\}$, and ρ_0 . These modifications allow us to generate states satisfying $\frac{1}{2} \|\mathbb{E}[\rho] - (X/\text{tr}(X))\|_1 \leq \epsilon$ for solutions X of either the Lyapunov equations or matrix inversion. For Lyapunov equations we assume $\text{tr}(B) = 1$ and show that, in expectation, our algorithm requires at most $T + 1$ calls to a block encoding of either A or $e^{\Delta A}$ for small Δ and a state-preparation circuit for $\rho_0 = B$. The same result holds for matrix inversion where we use $\rho_0 = I/N$ instead. Once the state is prepared, it can be used for the estimation of expectations values $\text{tr}(XO)/\text{tr}(X)$, overlaps $\langle \psi | X | \phi \rangle / \text{tr}(X)$, or the normalization $\text{tr}(X)$. Table I summarizes the main results. Approximate implementations of the map $\mathcal{E}$ lead to an additive error discussed in the main text.

For matrix inversion our algorithm appears to match the optimal query complexity bound in κ, ϵ for quantum linear system solvers [19–21]. However, it is worthwhile to emphasize the subtle difference: for quantum linear systems we want to output a pure state $|\mathbf{x}\rangle \propto A^{-1} |\mathbf{b}\rangle$, for matrix inversion we want to output a mixed state $\rho \propto A^{-1}$. While the former problem is known to be BQP complete [22], we are not aware of the complexity class of the latter. Finally, we compare quantum algorithms for solving the continuous-time Lyapunov equation in Appendix F. Bearing in mind the distinction in output formats (block encodings versus states), the query complexity analysis indicates that our algorithm remains competitive while relying on potentially simpler primitives.

II. BACKGROUND

We begin with a description of the probabilistic quantum algorithm by Zhang *et al.* [18]. Throughout this paper

TABLE I. Overview of results. Algorithm 1 achieves trace distance ϵ or $\epsilon_1 + \epsilon_2$ between the expected output state and normalized solutions $X/\text{tr}(X)$ of the listed problems using the stated parameters. We set parameter $r_k = 1/(T + 1 - k)$ for all $k \in [T + 1]$ and assume $B \geq 0$, $\text{tr}(B) = 1$, $[A, A^\dagger] = 0$. On average the algorithm queries a block encoding of M and a state-preparation circuit for ρ_0 at most $T + 1$ times. $\lambda(A)$ is the set the eigenvalues of A and κ the condition number of A .

Problem	Assumption	Parameters		
		T	M [for $\mathcal{E}(\cdot) = M \cdot M^\dagger$]	ρ_0
Discrete-time Lyapunov	$\ A\ < 1$	$\left\lceil \frac{\ln(1/\epsilon)}{2 \ln(1/\ A\)} \right\rceil$	A	B
Continuous-time Lyapunov	$\text{Re}[\lambda(A)] < 0$	$\left\lceil \frac{\ln(1/\epsilon_2) \ A\ \min\{\text{Re}[\lambda(A)]\} }{2\epsilon_1 \max\{\text{Re}[\lambda(A)]\} ^2} \right\rceil$	$e^{\frac{\epsilon_1}{\ A\ } \max\{\text{Re}[\lambda(A)]\} \min\{\text{Re}[\lambda(A)]\}^A}$	B
Matrix inversion I	$A \succ 0$	$\left\lceil \kappa \ln\left(\frac{1}{\epsilon}\right) \right\rceil$	$\sqrt{I - \frac{A}{\ A\ }}$	$\frac{I}{N}$
Matrix inversion II	$A \succ 0$	$\left\lceil \frac{\kappa^2}{\epsilon} \ln\left(\frac{2}{\epsilon}\right) \right\rceil$	$e^{-\frac{\epsilon}{2\kappa\ A\ } A}$	$\frac{I}{N}$

$\|\cdot\|_1$ denotes the trace norm, $\|\cdot\|$ the operator norm, $\lambda(A)$ the set of eigenvalues of A , and for integer $i > 0$, $[i] := \{0, 1, \dots, i-1\}$.

The algorithm starts with an initial quantum state ρ_0 and iteration number $i = 0$. At each iteration i there are up to two probabilistic events. The first event is given by a Bernoulli random variable $b \sim \mathcal{B}(r_i)$ where $0 \leq r_i \leq 1$ is the probability of observing the $b = 1$ outcome. When $b = 1$ is observed, the algorithm stops and returns the current state ρ_i . When $b = 0$, the algorithm applies a channel to the current state:

$$\Phi(\rho_i) = |0\rangle\langle 0| \otimes \mathcal{F}(\rho_i) + |1\rangle\langle 1| \otimes \mathcal{E}(\rho_i), \quad (4)$$

where $\mathcal{E}, \mathcal{F}$ are two trace nonincreasing CP maps that form a quantum instrument. The channel Φ can be interpreted as either the production of a desired state $\mathcal{E}(\rho_i)/\text{tr}[\mathcal{E}(\rho_i)]$ with probability $p_{i+1} = \text{tr}[\mathcal{E}(\rho_i)]$, or the production of an undesired state $\mathcal{F}(\rho_i)/(1 - \text{tr}[\mathcal{E}(\rho_i)])$ with probability $1 - p_{i+1}$. Therefore, a second probabilistic event is triggered by measuring the ancilla qubit in the computational basis. This is another Bernoulli random variable $a \sim \mathcal{B}(p_{i+1})$. A measurement outcome $a = 1$ indicates that the desired state has been produced. In this case we increment the iteration number to $i + 1$ and proceed with a recursive call of the iteration just described. If $a = 0$, we introduce a fresh copy of the initial state ρ_0 and restart the algorithm at iteration $i = 0$.

We briefly mention a model for an exact implementation of the CP maps and discuss its approximate implementation later. For our purposes it is sufficient to focus on the map $\mathcal{E}$ and assume a single, square Kraus operator M such that

$$\mathcal{E}(\cdot) = M \cdot M^\dagger. \quad (5)$$

We suppose that $\mathcal{E}$ is implemented by block encoding M . That is, we have an $(n + m)$ -qubit unitary U_M such that $M = \alpha(|1\rangle^{\otimes m} \otimes I^{\otimes n})U_M(|1\rangle^{\otimes m} \otimes I^{\otimes n})$ for $\alpha \geq \|M\|$ and n, m the number of system qubits and ancillas, respectively. Projecting on the $|1\rangle^{\otimes m}$ subspace by postselecting on the all-ones outcome of the ancillas implements the map $\mathcal{E}$. Block-encoding constructions are thoroughly discussed in Refs. [23,24] and references therein. To keep notation light, in Eq. (4) we assume postselection on a single ancilla. This is without loss of generality, since we can always add an extra qubit to the block encoding and set it to $|1\rangle$ if the state of the other ancillas is $|1\rangle^{\otimes m}$.

III. RESULTS

The proposed algorithm is summarized as pseudocode in Algorithm 1 and the resulting stochastic process is illustrated in Fig. 1. Here we focus on three applications: solving the discrete- and continuous-time Lyapunov equations and matrix inversion.

ALGORITHM 1. The probabilistic quantum algorithm. Executed with $\text{SAMPLESTATE}(\rho_0, 0)$. Returns the quantum state given in Lemma 1 with stopping time in Lemma 2.

Require: State preparation circuit for ρ_0 , trace non-increasing CP map $\mathcal{E}$, parameters $T, \{r_i\}_{i=0}^{T-1}$, and $r_T = 1$

function $\text{SAMPLESTATE}(\rho, i)$

sample Bernoulli random variable $b \sim \mathcal{B}(r_i)$

if $b = 1$ **then return** ρ

apply channel $\Phi(\rho) = |0\rangle\langle 0| \otimes \mathcal{F}(\rho) + |1\rangle\langle 1| \otimes \mathcal{E}(\rho)$.

measure ancilla $a \sim \mathcal{B}(\text{tr}[\mathcal{E}(\rho)])$

if $a = 1$ **then**

$\rho \leftarrow \frac{\mathcal{E}(\rho)}{\text{tr}[\mathcal{E}(\rho)]}$

$i \leftarrow i + 1$

else

$\rho \leftarrow \rho_0$

$i \leftarrow 0$

return $\text{SAMPLESTATE}(\rho, i)$

We introduce a deterministic stopping rule where the algorithm returns if it has successfully completed T iterations. This can be implemented simply by setting $r_T = 1$. The expected output state of the stopped process is captured in the following lemma, which is shown in Appendix B 1.

Lemma 1 (Expected output state). Let $\{\mathcal{E}, \mathcal{F}\}$ be a quantum instrument with $\mathcal{E}, \mathcal{F}$ CP maps between states on Hilbert space $\mathcal{H}$. Let $0 \leq r_k < 1$ for $k \in [T]$, $r_T = 1$, and ρ_0 an arbitrary initial state on $\mathcal{H}$. Then the expected state at stopping time τ output by Algorithm 1 is given by

$$\mathbb{E}[\rho] = \frac{\sum_{k=0}^T r_k R_k \mathcal{E}^k(\rho_0)}{\sum_{l=0}^T r_l R_l \text{tr}[\mathcal{E}^l(\rho_0)]} \quad (6)$$

with $R_k := \prod_{j=0}^{k-1} (1 - r_j)$ for $k > 0$ and $R_0 = 1$.

We can generate different expected output states by tuning the probabilities r_k and changing T . Suppose we want to produce expected states with positive coefficients

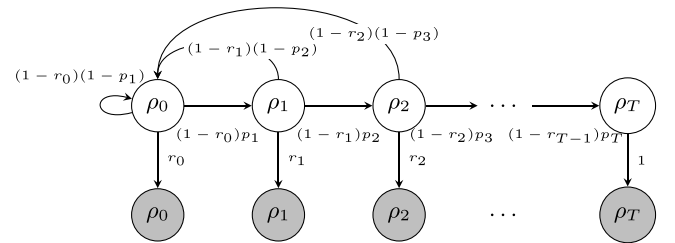


FIG. 1. Stochastic process of Algorithm 1. At each iteration i we have state $\rho_i = \mathcal{E}^i(\rho_0)/\text{tr}[\mathcal{E}^i(\rho_0)]$ (white circle), probability r_i of returning the current state (grey) and stopping or probability $1 - r_i$ of continuing. Conditioned on continuing we have probability $p_{i+1} = \text{tr}[\mathcal{E}(\rho_i)]$ of applying the CP map $\mathcal{E}$ and probability $1 - p_{i+1}$ of restarting with ρ_0 at $i = 0$.

$r_k R_k = c_k > 0$ for all k . This can be achieved by setting the Bernoulli probabilities to

$$r_k = \frac{c_k}{1 - \sum_{j=0}^{k-1} c_j} \quad \text{for } k \in [T]. \quad (7)$$

Since $r_T = 1$, we have the condition that $\sum_{k=0}^T c_k = 1$. Note that this setting is valid for $T \rightarrow \infty$ if the series of c_k converges to 1. In the following applications we work with finite T and set all coefficients to a constant $c_k = 1/(T+1)$. Then they cancel from Eq. (6) and we have probabilities $r_k = 1/(T+1-k)$ for all $k \in [T+1]$.

In principle, the algorithm could run forever, e.g., in the case where all iterations lead to a restart in Fig. 1. The following lemma shows that this does not happen on average because the expected stopping time is bounded.

Lemma 2 (Expected stopping time). The expected stopping time of Algorithm 1 is given by

$$\mathbb{E}[\tau] = \frac{\sum_{k=0}^T R_k \text{tr}[\mathcal{E}^k(\rho_0)]}{\sum_{l=0}^T r_l R_l \text{tr}[\mathcal{E}^l(\rho_0)]}. \quad (8)$$

As shown in Appendix B2, the following corollary follows immediately.

Corollary 1. Let $r_k R_k = c_k > 0$ as in Eq. (7). Then the expected stopping time is bounded by

$$\mathbb{E}[\tau] \leq \frac{1}{\min\{c_k\}}. \quad (9)$$

For the case $r_k = 1/(T+1-k)$ with $k \in [T+1]$ the expected stopping time is bounded by

$$\mathbb{E}[\tau] \leq T+1. \quad (10)$$

Due to Markov's inequality, $\mathbb{P}(\tau \geq a\mathbb{E}[\tau]) \leq 1/a$. For example, there is a 75% chance that the algorithm stops within 4 times the expected stopping time.

When the process stops, the resulting state can be used for downstream tasks. In Appendix E we show how to estimate expectation values, state overlaps, and normalization constants.

A. Discrete-time Lyapunov equation

Consider $N \times N$ matrices A, B and recall the discrete-time Lyapunov equation (1), $AXA^\dagger - X + B = 0$. We assume that $\|A\| < 1$, $[A, A^\dagger] = 0$, B is nonzero positive semidefinite, denoted with $B \geq 0$, and $\text{tr}(B) = 1$. For simplicity we assume $N = 2^n$ with n the number of qubits of the main register representing the desired state. Since

$\|A\| < 1$, a unique solution to Eq. (1) exists and can be written as [2]

$$X = \sum_{k=0}^{\infty} A^k B A^{\dagger k}. \quad (11)$$

Our algorithm can produce a normalized approximate solution as its expected state of the stopped process. To see this consider the CP map $\mathcal{E}(\cdot) = A \cdot A^\dagger$, probabilities $r_k = 1/(T+1-k)$, and initial state $\rho_0 = B$. We assume that we have a state-preparation circuit for B . The map $\mathcal{E}$ is implemented by block encoding its Kraus operator A . Inserting these parameters into Eq. (6) shows that our algorithm produces the expected output state

$$\mathbb{E}[\rho] = \frac{\sum_{k=0}^T A^k B A^{\dagger k}}{\sum_{k=0}^T \text{tr}(A^k B A^{\dagger k})}. \quad (12)$$

Next we need to bound the approximation error from finite T . This allows us to choose T such that the error becomes arbitrarily small as summarized by the following theorem.

Theorem 1 (Solution of discrete-time Lyapunov equation). Let $A, B \in \mathbb{C}^{2^n \times 2^n}$ with $\|A\| < 1$, $[A, A^\dagger] = 0$, $B \geq 0$, $\text{tr}(B) = 1$, and $0 < \epsilon < 1$. If $\|A\| \neq 0$, set

$$T^* := \left\lceil \frac{\ln(\frac{1}{\epsilon})}{2 \ln\left(\frac{1}{\|A\|}\right)} \right\rceil \quad (13)$$

and $T^* = 0$ otherwise. Instantiate Algorithm 1 with $T \geq T^*$, probabilities $r_k = 1/(T+1-k)$ for $k \in [T+1]$, the CP map $\mathcal{E}(\cdot) = A \cdot A^\dagger$, and initial state $\rho_0 = B$. Then the expected output state satisfies $\frac{1}{2} \|\mathbb{E}[\rho] - (X/\text{tr}(X))\|_1 \leq \epsilon$, where X is the solution of the discrete-time Lyapunov equation (1).

The proof is in Appendix C1. In Appendix D we prove that the choice of T^* in Theorem 1 is asymptotically optimal.

Choosing $T = T^*$ results in the shortest expected stopping time. Then Corollary 1 implies that the algorithm produces an approximation to the normalized solution of the discrete-time Lyapunov equation with (in expectation) at most $T^* + 1$ calls to the block encoding of A and the state-preparation circuit for B .

B. Continuous-time Lyapunov equation

Now recall the continuous-time Lyapunov equation (2), $AX + XA^\dagger + B = 0$. In this subsection, we assume that all eigenvalues of A have strictly negative real parts. Furthermore, $[A, A^\dagger] = 0$, $B \geq 0$, and $\text{tr}(B) = 1$ as before. Equation (2) has a unique solution if and only if the eigenvalues of A satisfy $\lambda_i + \lambda_j^* \neq 0$ for all $\lambda_i, \lambda_j \in \lambda(A)$. This

is satisfied because all eigenvalues have strictly negative real parts. Furthermore, the solution is given by [2]

$$X = \int_0^\infty e^{tA} B e^{tA^\dagger} dt. \quad (14)$$

To approximate this solution with our algorithm we truncate the integral in Eq. (14) to the domain $[0, L]$ and discretize using the left Riemann sum. We choose an interval size Δ and, for integer T , a cutoff $L = (T + 1)\Delta$. Then

$$X \approx \Delta \sum_{k=0}^{T/\Delta-1} e^{k\Delta A} B e^{k\Delta A^\dagger}. \quad (15)$$

Let us choose as the CP map $\mathcal{E}(\cdot) = e^{\Delta A} \cdot e^{\Delta A^\dagger}$, probabilities $r_k = 1/(T + 1 - k)$, and initial state $\rho_0 = B$. For the moment we assume that $\mathcal{E}$ is implemented exactly by block encoding its Kraus operator $e^{\Delta A}$. Inserting the parameters into Eq. (6) shows that our algorithm produces the expected output state

$$\mathbb{E}[\rho] = \frac{\sum_{k=0}^T e^{k\Delta A} B e^{k\Delta A^\dagger}}{\sum_{k=0}^T \text{tr}(e^{k\Delta A} B e^{k\Delta A^\dagger})}. \quad (16)$$

It remains to bound the approximation error from the discretization (denoted ϵ_1) and truncation (ϵ_2) of the solution. This allows us to choose parameters T and Δ such that the error becomes arbitrarily small as summarized by the following theorem proven in Appendix C 2.

Theorem 2 (*Solution of continuous-time Lyapunov equation*). Let $A, B \in \mathbb{C}^{2^n \times 2^n}$ with $\text{Re}(\lambda_i) < 0$ for all $\lambda_i \in \lambda(A)$, $[A, A^\dagger] = 0$, $B \geq 0$, $\text{tr}(B) = 1$, and $0 < \epsilon_1, \epsilon_2 < 1$. Set

$$T^* := \left\lceil \frac{1}{2\epsilon_1} \ln \left(\frac{1}{\epsilon_2} \frac{\|A\| \max_k \{\text{Re}(\lambda_k)\}}{\max_j \{\text{Re}(\lambda_j)\}^2} \right) \right\rceil, \quad (17)$$

$$\Delta^* := \frac{\epsilon_1 \max_j \{\text{Re}(\lambda_j)\}}{\|A\| \min_k \{\text{Re}(\lambda_k)\}}. \quad (18)$$

Instantiate Algorithm 1 with $T \geq T^*$, $0 < \Delta \leq \Delta^*$, probabilities $r_k = 1/(T + 1 - k)$ for $k \in [T + 1]$, the CP map $\mathcal{E}(\cdot) = e^{\Delta A} \cdot e^{\Delta A^\dagger}$, and initial state $\rho_0 = B$. Then the expected output state satisfies $\frac{1}{2} \|\mathbb{E}[\rho] - (X/\text{tr}(X))\|_1 \leq \epsilon_1 + \epsilon_2$, where X is the solution of the continuous-time Lyapunov equation (2).

Corollary 1 implies that the algorithm requires in expectation at most $T^* + 1$ calls to the block encoding of $e^{\Delta A}$ and the state-preparation circuit for B . This is multiplied by the cost of implementing the block encoding, which, of note, shall not have subnormalization factors that can bias the transition probabilities r_k . In

the special case of Hermitian A , the condition number is $\kappa = \min_k \{\lambda_k\} / \max_j \{\lambda_j\}$, and thus we find $T^* = \lceil (\kappa^2/2\epsilon_1) \ln(1/\epsilon_2) \rceil$ and $\Delta^* = \epsilon_1/\kappa \|A\|$.

C. Matrix inversion

Consider a matrix $A \succ 0$ and the task of finding its normalized inverse. We use κ to indicate its condition number. We can reduce the problem to the discrete-time Lyapunov equation $A'X(A')^\dagger - X + B = 0$ with $A' = \sqrt{I - A/\|A\|}$ and $B = I/N$. To see this, note that $X = A^{-1}\|A\|/N$ commutes with A' and solves the equation. Since $\|A'\| = \sqrt{1 - 1/\kappa} < 1$, the solution is unique. Finding the solution with our algorithm requires the CP map $\mathcal{E}(\cdot) = \sqrt{I - A/\|A\|} \cdot \sqrt{I - A/\|A\|}$. Plugging this map in Theorem 1 we obtain a state that satisfies $\frac{1}{2} \|\mathbb{E}[\rho] - (A^{-1}/\text{tr}(A^{-1}))\|_1 \leq \epsilon$ and expected stopping time bounded by $\lceil \kappa \ln(1/\epsilon) \rceil + 1$ yielding query complexity $\mathcal{O}[\kappa \ln(1/\epsilon)]$ in expectation [25].

An alternative path to matrix inversion is to reduce it to the continuous-time Lyapunov equation $A'X + X(A')^\dagger + B = 0$ with $A' = -A/2$ and $B = I/N$. Note that $X = A^{-1}/N$ solves the equation. Since $\text{Re}(\lambda'_i) = -\text{Re}(\lambda_i)/2 < 0$ for any pair of eigenvalues λ'_i and λ_i corresponding to the same eigenvector of A' and A , respectively, the solution is unique. Using Theorem 2 for this case with the choice $\epsilon_1 = \epsilon_2 = \epsilon/2$, we obtain an expected output state that satisfies $\frac{1}{2} \|\mathbb{E}[\rho] - (A^{-1}/\text{tr}(A^{-1}))\|_1 \leq \epsilon$ using the CP map $\mathcal{E}(\cdot) = e^{-\epsilon A/2\kappa\|A\|} \cdot e^{-\epsilon A/2\kappa\|A\|}$. The expected stopping time is bounded by $\lceil (\kappa^2/\epsilon) \ln(2/\epsilon) \rceil + 1$.

In Table I we denote the discrete-time approach as matrix inversion I, and the continuous-time approach as matrix inversion II. Which of the two algorithms is preferable? The discrete-time approach has a favorable scaling in both ϵ and κ , but the answer really depends on the cost of the block encoding. If we can block encode $\sqrt{A/\|A\|}$, then the qubitization trick [23, Lemmas 8 and 10] yields the block encoding of $\sqrt{I - A/\|A\|}$ in an off-diagonal block, which could be used in our discrete-time algorithm. Alternatively, one can use the quantum singular value transformation [24] to block-encode approximations to functions of A , including $\sqrt{I - A/\|A\|}$ and $e^{-\epsilon A/2\kappa\|A\|}$ (see Appendix C 4 for an explicit construction of the latter).

D. Approximate implementation

So far we have assumed an exact implementation of the CP map $\mathcal{E}$ via a block encoding of M . Now suppose that we can only implement an approximate block encoding $\tilde{M}$ yielding the trace nonincreasing CP map $\tilde{\mathcal{E}}(\cdot) = \tilde{M} \cdot \tilde{M}^\dagger$. This leads to a further additive error $\epsilon_\sim$ in the trace distance $\frac{1}{2} \|\mathbb{E}[\rho] - (X/\text{tr}(X))\|_1$. As shown in Appendix C 3 this error can be achieved in practice by choosing a block encoding such that $\|M - \tilde{M}\| \leq \epsilon_\sim/2T(T + 1)$ with T from Theorem 1 or Theorem 2, respectively. For example, the error of the continuous-time algorithm with this

approximate implementation is $\frac{1}{2} \|\mathbb{E}[\rho] - (X/\text{tr}(X))\|_1 \leq \epsilon_1 + \epsilon_2 + \epsilon_\sim$. In Appendix C 4 we provide an explicit construction of the approximate block encoding of $M = e^{\Delta A}$, using $\mathcal{O}[\kappa \log(T/\epsilon_\sim)]$ calls to the block encoding of A .

IV. CONCLUSION

We have presented a probabilistic quantum algorithm for approximating solutions to Lyapunov equations and matrix inversion under certain conditions. Given block-encoding access to a function of the normal input matrix A and a state-preparation circuit for ρ_0 , in expectation the algorithm outputs mixed states $\propto \sum_{k=0}^T \mathcal{E}^k(\rho_0)$. With the choice of T , $\mathcal{E}$, and ρ_0 in Table I the expected output state is proportional to the respective solution up to the given error in trace distance. The expected stopping time of the algorithm is at most $T + 1$ yielding query complexity $\mathcal{O}(T)$. For positive definite A and access to a block encoding of $\sqrt{A/\|A\|}$ the query complexity of matrix inversion matches the lower bound $\Omega[\kappa \ln(1/\epsilon)]$ of quantum linear system solvers [19–21]. In contrast, however, we only get sample access to the expected output state representing the solution. Nevertheless, our algorithm can be used in applications that require estimating expectations, overlaps, or traces of the normalized solution.

The presented algorithm could be extended to higher-precision quadrature formulas for Eq. (14) and to approximating more general integrals than Eq. (14). First, admitting a different CP map at each iteration yields an expected output state $\propto \sum_{k=0}^T c_k(\mathcal{E}_k \circ \dots \circ \mathcal{E}_0)(\rho_0)$. For example, $\mathcal{E}_k(\cdot) = e^{(t_k - t_{k-1})A} \cdot e^{(t_{k-1} - t_{k-2})A^\dagger}$ with a suitable choice of t_k , c_k leads to a matrix equivalent of Gaussian quadrature [26]. For smooth scalar functions Gaussian quadrature rules typically converge much faster than the Riemann sum we use. It would be interesting to analyze whether this can improve the scaling of parameter T with ϵ_1 in the continuous-time Lyapunov case. Second, in Eq. (14) we could introduce a weighting function $f(t)$ or admit an anti-Hermitian A . The former leads to a generalization of the Laplace transform to matrices, the latter to the (one-sided) operator Fourier transform. It would be fruitful to analyze approximation errors and applications of those generalizations.

For the lower bound on the copies of ρ_C , we will employ Ref. [17, Theorem 2]. This tells us that, if we are given asymptotically fewer than $\sqrt{N}$ copies of the state $|\phi_P\rangle := (1/\sqrt{N}) \sum_{j=1}^N |j, P(j)\rangle$, then we cannot with high probability decide if $P(1) = 1$. Thus, it suffices to show that there is a random procedure, which, in expectation, uses a constant number of copies of $|\phi_P\rangle$ and prepares the state ρ_C . Let $F_P := \{j \mid P(j) = j\}$ be the set of fixed points of P and let $M_P := |F_P| = \text{tr}[P] = \text{tr}[P^\dagger]$. Note that

$$\rho_C = \frac{2I + P + P^\dagger}{2N + 2M_P} \quad (\text{A1})$$

$$= \frac{1}{2N + 2M_P} \left(\sum_{j \in F_P} 4 |j\rangle\langle j| + \sum_{j \notin F_P} (2 |j\rangle\langle j| + |P(j)\rangle\langle j| + |P^\dagger(j)\rangle\langle j|) \right) \quad (\text{A2})$$

ACKNOWLEDGMENTS

We thank Hitomi Mori, Maria Tudorovskaya, and Alexandre Krajenbrink for helpful feedback, and Frederic Sauvage for useful discussions.

DATA AVAILABILITY

No data were created or analyzed in this study.

APPENDIX A: INCOMPARABILITY OF BLOCK ENCODINGS AND STATES AS RESOURCES

Consider a positive semidefinite matrix $C \in \mathbb{C}^{N \times N}$ of spectral norm $\|C\| = 1$. This data can be encoded as a state $\rho_C = C/\text{tr}[C]$, or as a block encoding U_C such that $(I \otimes \langle G|) U_C (I \otimes |G\rangle) = C$ for some known state $|G\rangle$. The following two theorems, which are adapted from Ref. [17], show that the two encodings are incomparable computational resources.

Theorem A1. For a permutation matrix $P = \sum_{j=1}^N |P(j)\rangle\langle j| \in \mathbb{C}^{N \times N}$, let $C = (2I + P + P^\dagger)/4$. Consider the decision problem of determining whether $\langle 1|P|1\rangle$ is 1 or 0, that is, respectively, whether $\langle 1|C|1\rangle$ is 1 or 1/2. This problem can be solved with high probability using a constant number of queries to the block encoding U_C , but it requires $\Omega(\sqrt{N})$ copies of the mixed state ρ_C .

Note that both P and its inverse $P^\dagger$ yield the same C and, in the reverse direction, C uniquely determines the unordered pair $\{P, P^\dagger\}$ (which is a singleton if P is its own inverse). Hence the problem is well defined.

Proof. Consider an algorithm that ℓ times repeats the following: it applies U_C to $|1\rangle|G\rangle$ and then performs the two-outcome projective measurement corresponding to $|1\rangle|G\rangle$ and its orthogonal complement. If all ℓ measurements yield $|1\rangle|G\rangle$ the algorithm returns $\langle 1|P|1\rangle = 1$, otherwise it returns $\langle 1|P|1\rangle = 0$. Since the algorithm essentially samples from the probability distribution $(\langle 1|C|1\rangle^2, 1 - \langle 1|C|1\rangle^2)^\ell$, it fails with probability at most $1/4^\ell$. Hence, for any constant error $\epsilon > 0$ a constant number of queries to U_C suffices.

$$= \frac{1}{2N + 2M_P} \left(\sum_{j \in F_P} 4 |j\rangle\langle j| + \sum_{j \notin F_P} (|j\rangle\langle j| + |P(j)\rangle\langle P(j)| + |P(j)\rangle\langle j| + |j\rangle\langle P(j)|) \right) \quad (\text{A3})$$

$$= \frac{2N}{N + M_P} \left(\frac{1}{N} \sum_{j \in F_P} |j\rangle\langle j| + \frac{1}{2N} \sum_{j \notin F_P} \frac{|j\rangle + |P(j)\rangle}{\sqrt{2}} \frac{\langle j| + \langle P(j)|}{\sqrt{2}} \right). \quad (\text{A4})$$

Now we show that, given the expected number of $2/(1 + M_P/N) \in [1, 2]$ copies of the state $|\phi_P\rangle$, we can prepare one copy of ρ_C . First, we measure $|\phi_P\rangle$ in the computational basis, obtaining $|j, P(j)\rangle$ for a uniformly random j . If $P(j) = j$, we prepare and return the state $|j\rangle$. If $P(j) \neq j$, with probability $1/2$ we prepare and return the state $(|j\rangle + |P(j)\rangle)/\sqrt{2}$ and with probability $1/2$ we restart the procedure with a fresh copy of $|\phi_P\rangle$.

To see that this procedure indeed gives the state ρ_C in expectation, consider Eq. (A4). The term $(1/N) \sum_{j \in F_P} |j\rangle\langle j|$ corresponds to the event when the measurement of $|\phi_P\rangle$ yields $|j, j\rangle$. The trace of this former term is M_P/N , which is the probability of this event. The other term, namely,

$$\frac{1}{2} \cdot \frac{1}{N} \sum_{j \notin F_P} \frac{|j\rangle + |P(j)\rangle}{\sqrt{2}} \frac{\langle j| + \langle P(j)|}{\sqrt{2}}, \quad (\text{A5})$$

corresponds to the event when *both* the measurement of $|\phi_P\rangle$ yields $|j, P(j)\rangle$ with $P(j) \neq j$ and, subsequently, we decide with probability $1/2$ not to restart the procedure. The trace of this latter term is $(N - M_P)/2N$, which is the probability of this event. Hence, the unnormalized mixed state in the parenthesis of Eq. (A4) is the output of the procedure conditioned on having concluded without a restart. The trace of this state is $(N + M_P)/2N$, which is the probability of the procedure concluding without a restart, therefore we need to multiply by its reciprocal to normalize the state, thus obtaining ρ_C . Note that $(N + M_P)/2N = 1 - ((N - M_P)/2N)$, where $(N - M_P)/2N$ is the probability that we restart the procedure. ■

Theorem A2. Let $C \in \mathbb{C}^{N \times N}$ be a binary diagonal matrix with exactly one nonzero entry, that is, $C = |j\rangle\langle j|$ for some unknown $j \in \{1, \dots, N\}$. Consider the search problem of finding the unique j such that $\langle j | C | j \rangle = 1$. This problem requires $\Omega(\sqrt{N})$ queries to the block encoding U_C , but can be solved while a single copy of the state ρ_C .

Proof. The lower bound $\Omega(\sqrt{N})$ on the number of queries to the block encoding is given by Ref. [17, Theorem 4]. On the other hand, because $\rho_C = C = |j\rangle\langle j|$, measuring the unique copy of ρ_C in the computational basis yields j with certainty. ■

Arguably, an even stronger evidence of incomparability of U_C and ρ_C as computational resources would be to give two computational problems for the same C , such that for one problem U_C is a more useful resource while ρ_C is more useful for the other. Whether such C exists is left as an open problem.

APPENDIX B: ANALYSIS OF THE PROBABILISTIC ALGORITHM

The algorithm is described in the main text and presented as pseudocode in Algorithm 1. Here we show one iteration of the algorithm as a transformation, while hiding all the intermediary steps such as coin flip, block encoding, and ancilla measurement:

$$\begin{aligned} \rho_i \longrightarrow & \underbrace{r_i \rho_i}_{\text{return}} + (1 - r_i) \text{tr}[\mathcal{E}(\rho_i)] \underbrace{\frac{\mathcal{E}(\rho_i)}{\text{tr}[\mathcal{E}(\rho_i)]}}_{\substack{=: \rho_{i+1} \\ \text{continue}}} \\ & + (1 - r_i)(1 - \text{tr}[\mathcal{E}(\rho_i)]) \underbrace{\rho_0}_{\text{restart}}. \end{aligned} \quad (\text{B1})$$

1. Expected output state (Lemma 1)

We calculate the expected output state $\mathbb{E}[\rho]$ of the algorithm following the derivation in Zhang *et al.* [18]. Here ρ denotes the quantum state of the stopped process. We begin by noting that whenever the algorithm restarts, it does so from the initial state ρ_0 and the iteration counter is reset to $i = 0$. In this case the whole probability subtree can be replaced by $\mathbb{E}[\rho]$. We have

$$\mathbb{E}[\rho] = r_0 \rho_0 + (1 - r_0) (1 - \text{tr}[\mathcal{E}(\rho_0)]) \mathbb{E}[\rho] \quad (\text{B2})$$

$$+ r_1 (1 - r_0) \mathcal{E}(\rho_0) + (1 - r_1) (1 - r_0) (\text{tr}[\mathcal{E}(\rho_0)] - \text{tr}[\mathcal{E}^2(\rho_0)]) \mathbb{E}[\rho] \quad (\text{B3})$$

$$+ \dots \quad (\text{B4})$$

$$\begin{aligned} & + \left[r_k \prod_{j=0}^{k-1} (1 - r_j) \right] \mathcal{E}^k(\rho_0) + \left[\prod_{i=0}^k (1 - r_i) \right] \\ & \times (\text{tr}[\mathcal{E}^k(\rho_0)] - \text{tr}[\mathcal{E}^{k+1}(\rho_0)]) \mathbb{E}[\rho] \end{aligned} \quad (\text{B5})$$

$$+ \dots \quad (\text{B6})$$

$$+ \left[\prod_{l=0}^{T-1} (1 - r_l) \right] \mathcal{E}^T(\rho_0), \quad (\text{B7})$$

where the last line is given by the deterministic stopping rule, $r_T = 1$. Grouping terms and introducing the notation $R_k = \prod_{l=0}^{k-1} (1 - r_l)$ and $R_0 = 1$ we have

$$\begin{aligned} \mathbb{E}[\rho] &= \sum_{k=0}^T r_k R_k \mathcal{E}^k(\rho_0) + \mathbb{E}[\rho] \sum_{l=0}^T R_{l+1} \{ \text{tr}[\mathcal{E}^l(\rho_0)] \\ &\quad - \text{tr}[\mathcal{E}^{l+1}(\rho_0)] \}. \end{aligned} \quad (\text{B8})$$

With some work one can show that the second summation is equivalent to $1 - \sum_{l=0}^T r_l R_l \text{tr}[\mathcal{E}^l(\rho_0)]$. Solving for the expected state we arrive at

$$\mathbb{E}[\rho] = \frac{\sum_{k=0}^T r_k R_k \mathcal{E}^k(\rho_0)}{\sum_{l=0}^T r_l R_l \text{tr}[\mathcal{E}^l(\rho_0)]}. \quad (\text{B9})$$

By tuning the probabilities r_k and the stopping parameter T it is now possible to implement different expected states. As a warm-up example, we consider the case where $r_k R_k = c$ for some constant $c > 0$ and for all k , thus canceling out from Eq. (B9). This can be achieved by setting the Bernoulli probabilities to $r_k = c/(1 - kc)$. Since $r_T = 1$ we find that $c = 1/(T + 1)$. Thus, we have

$$\begin{aligned} r_k &= \frac{1}{1 + T - k} \text{ for } k \in [T + 1] \\ \implies \mathbb{E}[\rho] &= \frac{\sum_{k=0}^T \mathcal{E}^k(\rho_0)}{\sum_{l=0}^T \text{tr}[\mathcal{E}^l(\rho_0)]}. \end{aligned} \quad (\text{B10})$$

We emphasize that this setting is valid only for finite T since, otherwise, all $r_k = 0$, i.e., the algorithm never enters the “return” branch in Eq. (B1). As a second example, we consider the more general case where $r_k R_k = c_k$ for coefficients $0 < c_k < 1$. This can be achieved by the setting

$$\begin{aligned} r_k &= \frac{c_k}{1 - \sum_{j=0}^{k-1} c_j} \text{ for } k \in [T + 1] \\ \implies \mathbb{E}[\rho] &= \frac{\sum_{k=0}^T c_k \mathcal{E}^k(\rho_0)}{\sum_{l=0}^T c_l \text{tr}[\mathcal{E}^l(\rho_0)]}. \end{aligned} \quad (\text{B11})$$

Since $r_T = 1$, we have the condition that $\sum_{k=0}^T c_k = 1$, i.e., the coefficients must form a probability vector. If $\{c_k\}$ also forms a convergent series then, unlike the previous example, T can be taken to infinity.

2. Expected stopping time (Lemma 2 and Corollary 1)

For the expected stopping time $\mathbb{E}[\tau]$ of the algorithm one simply follows the derivation in [18, Lemma 7] to get

$$\mathbb{E}[\tau] = \frac{\sum_{k=0}^T R_k \text{tr}[\mathcal{E}^k(\rho_0)]}{\sum_{l=0}^T r_l R_l \text{tr}[\mathcal{E}^l(\rho_0)]}. \quad (\text{B12})$$

When the Bernoulli probabilities r_k are given by Eq. (B11), we have $R_k = 1 - \sum_{j=0}^{k-1} c_j$. Then

$$\begin{aligned} \mathbb{E}[\tau] &= \frac{\sum_{k=0}^T \left(1 - \sum_{j=0}^{k-1} c_j\right) \text{tr}[\mathcal{E}^k(\rho_0)]}{\sum_{l=0}^T c_l \text{tr}[\mathcal{E}^l(\rho_0)]} \\ &\leq \frac{\sum_{k=0}^T \text{tr}[\mathcal{E}^k(\rho_0)]}{\sum_{l=0}^T c_l \text{tr}[\mathcal{E}^l(\rho_0)]} \leq \frac{1}{\min\{c_l\}}, \end{aligned} \quad (\text{B13})$$

where we use that $0 < c_k < 1$ and $0 < \text{tr}[\mathcal{E}^k(\rho_0)] \leq 1$ for all k by assumption, so we can ignore the term with the minus sign. It follows that for constant coefficients $c_k = 1/(T + 1)$ as in Eq. (B10), we have $\mathbb{E}[\tau] \leq T + 1$. Intuitively, this is because the Bernoulli probabilities of entering the “return” branch in Eq. (B1) are $r_k \geq 1/(T + 1)$.

APPENDIX C: APPROXIMATION ERROR FOR LYAPUNOV EQUATIONS

We start by the following claim, which we employ below in Appendices C 1–C 3. Let $\succeq$ denote the Loewner order.

Claim C1. For nonzero positive semidefinite Y, Z of the same dimensions, we have

$$\frac{1}{2} \left\| \frac{Y}{\text{tr}(Y)} - \frac{Z}{\text{tr}(Z)} \right\|_1 \leq \frac{\|Y - Z\|_1}{\max\{\text{tr}(Y), \text{tr}(Z)\}}. \quad (\text{C1})$$

Proof. Assume that $\text{tr}(Y) \geq \text{tr}(Z)$ (if not one can simply swap the roles of Y and Z in the following derivation). Because $Y, Z \succeq 0$, we have $\|Y\|_1 = \text{tr}(Y)$ and $\|Z\|_1 = \text{tr}(Z)$. By applying the triangle inequality to $\|(Y - Z) + Z\|_1$, we get

$$\text{tr}(Y) - \text{tr}(Z) \leq \|Y - Z\|_1, \quad (\text{C2})$$

which we will use in a moment. Applying the triangle inequality to the distance that we want to bound, we get

$$\begin{aligned} \frac{1}{2} \left\| \frac{Y}{\text{tr}(Y)} - \frac{Z}{\text{tr}(Z)} \right\|_1 &\leq \frac{1}{2} \left\| \frac{Y}{\text{tr}(Y)} - \frac{Z}{\text{tr}(Y)} \right\|_1 \\ &\quad + \frac{1}{2} \left\| \frac{Z}{\text{tr}(Y)} - \frac{Z}{\text{tr}(Z)} \right\|_1 \end{aligned} \quad (\text{C3})$$

$$\begin{aligned}
&\leq \frac{\|Y - Z\|_1}{2 \operatorname{tr}(Y)} + \frac{\|Z\|_1}{2} \left| \frac{1}{\operatorname{tr}(Y)} - \frac{1}{\operatorname{tr}(Z)} \right| \\
&\quad (C4) \\
&\leq \frac{\|Y - Z\|_1 + [\operatorname{tr}(Y) - \operatorname{tr}(Z)]}{2 \operatorname{tr}(Y)}, \\
&\quad (C5)
\end{aligned}$$

where we use $\operatorname{tr}(Y) \geq \operatorname{tr}(Z)$ and, once again, $\|Z\|_1 = \operatorname{tr}(Z)$. The proof is concluded by applying Eq. (C2). ■

1. Proof of Theorem 1 (solution of the discrete-time Lyapunov equation)

Let us recap the setting of the theorem. We want to solve the discrete-time Lyapunov equation

$$AXA^\dagger - X + B = 0, \quad (C6)$$

where B is positive semidefinite, $\operatorname{tr}(B) = 1$, $[A, A^\dagger] = 0$, and all eigenvalues of A have magnitude strictly less than 1 [the following are equivalent: (i) it is Schur stable, (ii) its spectral radius is less than 1]. The spectral radius of a normal matrix equals the operator norm. Let us introduce the following notation for this section:

$$\mathcal{S}(T) := \sum_{k=0}^T A^k B A^{k\dagger}. \quad (C7)$$

Note that $B = CC^\dagger$ for some C , since B is positive semidefinite. Then the summand in $\mathcal{S}(T)$ can be written in the form $(A^k C)(A^k C)^\dagger$ and so it is also positive semidefinite. It follows that $\mathcal{S}(T)$ itself is positive semidefinite and thus

$$\|\mathcal{S}(T)\|_1 = \operatorname{tr}[\mathcal{S}(T)] \quad (C8)$$

$$\mathcal{S}(N) \succeq \mathcal{S}(T) \text{ for } N \geq T \quad (C9)$$

$$\|\mathcal{S}(\infty) - \mathcal{S}(T)\|_1 = \operatorname{tr}[\mathcal{S}(\infty)] - \operatorname{tr}[\mathcal{S}(T)]. \quad (C10)$$

Equation (C6) has a unique solution X if and only if the eigenvalues of A satisfy $\lambda_i \lambda_j \neq 1$ for all i, j . This is satisfied because we assume $\|A\| < 1$. Furthermore, for $\|A\| < 1$ the solution to Eq. (C6) is $X = \mathcal{S}(\infty)$. We can produce a normalized version of Eq. (C6) by using the probabilistic quantum algorithm with the following setup: the block encoding of A is used to implement the map $\mathcal{E}(\cdot) = A \cdot A^\dagger$, and the Bernoulli probabilities given in Eq. (B10) are used to implement constant coefficients c_k . Then the expected output state is

$$\mathbb{E}[\rho] = \frac{\sum_{k=0}^T A^k B A^{k\dagger}}{\operatorname{tr}(\sum_{l=0}^T A^l B A^{l\dagger})} = \frac{\mathcal{S}(T)}{\operatorname{tr}[\mathcal{S}(T)]}. \quad (C11)$$

To prove the theorem we find the value of the parameter T such that the expected output state is ϵ -close to the normalized solution in trace distance.

Proof of Theorem 1. Using Claim 1 and, then, Eq. (C10), we get

$$\begin{aligned}
\frac{1}{2} \left\| \frac{X}{\operatorname{tr}[X]} - \mathbb{E}[\rho] \right\|_1 &= \frac{1}{2} \left\| \frac{\mathcal{S}(\infty)}{\operatorname{tr}[\mathcal{S}(\infty)]} - \frac{\mathcal{S}(T)}{\operatorname{tr}[\mathcal{S}(T)]} \right\|_1 \\
&\leq \frac{\|\mathcal{S}(\infty) - \mathcal{S}(T)\|_1}{\operatorname{tr}[\mathcal{S}(\infty)]} \\
&= \frac{\operatorname{tr}[\mathcal{S}(\infty)] - \operatorname{tr}[\mathcal{S}(T)]}{\operatorname{tr}[\mathcal{S}(\infty)]}. \quad (C12)
\end{aligned}$$

For the numerator we find

$$\begin{aligned}
\operatorname{tr}[\mathcal{S}(\infty)] - \operatorname{tr}[\mathcal{S}(T)] &= \sum_{k=T+1}^{\infty} \operatorname{tr}(A^k B A^{k\dagger}) \\
&= \sum_{j=0}^{\infty} \operatorname{tr}(A^{j+T+1} B A^{j+T+1\dagger}) \quad (C13) \\
&= \operatorname{tr} \left[(A^\dagger A)^{T+1} \sum_{j=0}^{\infty} (A^\dagger A)^j B \right] \\
&= \operatorname{tr}[(A^\dagger A)^{T+1} (I - A^\dagger A)^{-1} B], \quad (C14)
\end{aligned}$$

where we use the closed form of the Neumann series. Since $A^\dagger A$ is positive semidefinite, we can write its spectral decomposition as $A^\dagger A = \sum_i \alpha_i |i\rangle\langle i|$ where $0 \leq \alpha_i < 1$ and $\{|i\rangle\}_i$ is an eigenbasis of $A^\dagger A$. Since B is positive semidefinite, we have that $\beta_i = \langle i | B | i \rangle \geq 0$. Thus we can continue as

$$\begin{aligned}
\operatorname{tr}[\mathcal{S}(\infty)] - \operatorname{tr}[\mathcal{S}(T)] &= \sum_i \alpha_i^{T+1} \frac{\beta_i}{1 - \alpha_i} \leq \max_j \{\alpha_j^{T+1}\} \sum_i \frac{\beta_i}{1 - \alpha_i} \quad (C15) \\
&= \max\{|\lambda_i| \mid \lambda_i \in \lambda(A)\}^{2(T+1)} \operatorname{tr}[(I - A^\dagger A)^{-1} B], \quad (C16)
\end{aligned}$$

with equality when $T = -1$, for which $\mathcal{S}(-1) = 0$. Recall that A is normal so its spectral radius equals the operator norm. Then Eq. (C12) is bounded as

$$\begin{aligned}
\frac{1}{2} \left\| \frac{X}{\operatorname{tr}[X]} - \mathbb{E}[\rho] \right\|_1 &\leq \frac{\operatorname{tr}[\mathcal{S}(\infty)] - \operatorname{tr}[\mathcal{S}(T)]}{\operatorname{tr}[\mathcal{S}(\infty)] - \operatorname{tr}[\mathcal{S}(-1)]} \\
&\leq \|A\|^{2(T+1)} \leq \epsilon, \quad (C17)
\end{aligned}$$

where we introduced a parameter $0 < \epsilon < 1$. Assuming $\|A\| \neq 0$ and solving for T we conclude that

$$T \geq \left\lceil \frac{\ln(\frac{1}{\epsilon})}{2 \ln(\frac{1}{\|A\|})} \right\rceil \implies \frac{1}{2} \left\| \frac{X}{\operatorname{tr}[X]} - \mathbb{E}[\rho] \right\|_1 \leq \epsilon. \quad (C18)$$

The trivial case $\|A\| = 0$ has solution $X = B$, which is achieved by setting $T \geq 0$. ■

2. Proof of Theorem 2 (solution of the continuous-time Lyapunov equation)

We consider the continuous-time Lyapunov equation

$$AX + XA^\dagger + B = 0, \quad (\text{C19})$$

where B is positive semidefinite, $\text{tr}(B) = 1$, $[A, A^\dagger] = 0$, and all eigenvalues of A satisfy $\text{Re}[\lambda(A)] < 0$ (Hurwitz stable). Equation (C19) has a unique solution if and only if the eigenvalues of A satisfy $\lambda_i + \lambda_j^* \neq 0$ for all $\lambda_i, \lambda_j \in \lambda(A)$. This is satisfied because we assume $\text{Re}[\lambda(A)] < 0$ so the sum of the real parts of $\lambda_i + \lambda_j^*$ cannot be 0. Furthermore, the unique solution is given by

$$X = \int_0^\infty e^{tA} B e^{tA^\dagger} dt, \quad (\text{C20})$$

which can be verified by plugging it back in Eq. (C19). We approximate this integral in the domain $[0, L]$ using the left Riemann sum. We discretize the interval using L/Δ , which is assumed to be an integer, and intervals of size $\Delta > 0$:

$$X \approx \Delta \sum_{k=0}^{\frac{L}{\Delta}-1} e^{k\Delta A} B e^{k\Delta A^\dagger}. \quad (\text{C21})$$

Suppose that we have a quantum state-preparation circuit for B and a block-encoding circuit for $e^{\Delta A}$. We can implement the CP map $\mathcal{E}(\cdot) = e^{\Delta A} \cdot e^{\Delta A^\dagger}$ by executing the block-encoding circuit, measuring the ancilla, and postselecting. For the coefficients, we want to use Eq. (B10) and $T = (L/\Delta) - 1$ so that the expected output state is

$$\mathbb{E}[\rho] = \frac{\sum_{k=0}^{\frac{L}{\Delta}-1} e^{k\Delta A} B e^{k\Delta A^\dagger}}{\text{tr}\left(\sum_{l=0}^{\frac{L}{\Delta}-1} e^{l\Delta A} B e^{l\Delta A^\dagger}\right)}, \quad (\text{C22})$$

which is the normalized version of Eq. (C21). Let us introduce some notation for the calculations in this section. We define

$$\mathcal{I}(L) := \int_0^L e^{tA} B e^{tA^\dagger} dt \quad \text{and} \quad \mathcal{S}(L) := \Delta \sum_{k=0}^{\frac{L}{\Delta}-1} e^{k\Delta A} B e^{k\Delta A^\dagger}. \quad (\text{C23})$$

We are now ready to prove the theorem: we find the value of the parameters L and Δ such that the expected output state is ϵ -close to the normalized solution in trace distance. Since B is positive semidefinite, we have $B = CC^\dagger$ for some C . Then the integrand can be written in the form

$(e^{tA}C)(e^{tA}C)^\dagger$ and it is also positive semidefinite (similarly for the summation). It follows that both $\mathcal{I}(L)$ and $\mathcal{S}(L)$ are positive semidefinite, and, in turn, we have

$$\|\mathcal{I}(L)\|_1 = \text{tr}[\mathcal{I}(L)] \quad \text{and} \quad \|\mathcal{S}(L)\|_1 = \text{tr}[\mathcal{S}(L)]. \quad (\text{C24})$$

Furthermore, we have

$$\mathcal{I}(T) \succeq \mathcal{I}(L) \quad \text{for } T \geq L, \quad (\text{C25})$$

$$\mathcal{S}(T) \succeq \mathcal{S}(L) \quad \text{for } T \geq L. \quad (\text{C26})$$

Proof of Theorem 2. Let us denote $R := \max_k \{\text{Re}(\lambda_k)\} < 0$. Using the triangle inequality

$$\frac{1}{2} \left\| \frac{X}{\text{tr}(X)} - \mathbb{E}[\rho] \right\|_1 = \frac{1}{2} \left\| \frac{\mathcal{I}(\infty)}{\text{tr}[\mathcal{I}(\infty)]} - \frac{\mathcal{S}(L)}{\text{tr}[\mathcal{S}(L)]} \right\|_1 \quad (\text{C27})$$

$$\begin{aligned} &\leq \frac{1}{2} \underbrace{\left\| \frac{\mathcal{I}(\infty)}{\text{tr}[\mathcal{I}(\infty)]} - \frac{\mathcal{S}(\infty)}{\text{tr}[\mathcal{S}(\infty)]} \right\|_1}_{\leq \epsilon_1} \\ &\quad + \frac{1}{2} \underbrace{\left\| \frac{\mathcal{S}(\infty)}{\text{tr}[\mathcal{S}(\infty)]} - \frac{\mathcal{S}(L)}{\text{tr}[\mathcal{S}(L)]} \right\|_1}_{\leq \epsilon_2}. \end{aligned} \quad (\text{C28})$$

The values $0 < \epsilon_1, \epsilon_2 < 1$ above are our desired bounds for these two distances. The latter distance is bounded as in the discrete case, namely, as in the proof of Theorem 1. In particular, analogously to Eq. (C17), we get

$$\frac{1}{2} \left\| \frac{\mathcal{S}(\infty)}{\text{tr}[\mathcal{S}(\infty)]} - \frac{\mathcal{S}(L)}{\text{tr}[\mathcal{S}(L)]} \right\|_1 \leq \|e^{\Delta A}\|^{2((L/\Delta)-1+1)} = e^{2RL}. \quad (\text{C29})$$

For this distance to be at most ϵ_2 , we can choose

$$L \geq \frac{\ln(1/\epsilon_2)}{-2 \max_j \{\text{Re}(\lambda_j)\}}. \quad (\text{C30})$$

For the former distance, we apply Claim 1, getting

$$\frac{1}{2} \left\| \frac{\mathcal{I}(\infty)}{\text{tr}[\mathcal{I}(\infty)]} - \frac{\mathcal{S}(\infty)}{\text{tr}[\mathcal{S}(\infty)]} \right\|_1 \leq \frac{\|\mathcal{S}(\infty) - \mathcal{I}(\infty)\|_1}{\text{tr}[\mathcal{S}(\infty)]}. \quad (\text{C31})$$

a. Bounding numerator

We define the function $F(t) := e^{tA} B e^{tA^\dagger}$ with derivative $F'(t) = e^{tA} A B e^{tA^\dagger} + e^{tA} B A^\dagger e^{tA^\dagger}$. Note that, for $t \geq \Delta l$, we

have $F(t) = F(\Delta l) + \int_{\Delta l}^t F'(s) ds$. We find

$$\|S(\infty) - \mathcal{I}(\infty)\|_1 = \left\| \Delta \sum_{l=0}^{\infty} e^{\Delta l A} B e^{\Delta l A^\dagger} - \int_0^\infty e^{tA} B e^{tA^\dagger} dt \right\|_1 \quad (\text{C32})$$

$$= \left\| \sum_{l=0}^{\infty} \int_{\Delta l}^{\Delta(l+1)} (F(\Delta l) - F(t)) dt \right\|_1 \quad (\text{C33})$$

$$\leq \sum_{l=0}^{\infty} \int_{\Delta l}^{\Delta(l+1)} \left(\int_{\Delta l}^t \|F'(s)\|_1 ds \right) dt. \quad (\text{C34})$$

We note that

$$\|F'(s)\|_1 \leq \|e^{sA}\| \|AB + BA^\dagger\|_1 \|e^{sA^\dagger}\| \quad (\text{C35})$$

$$= e^{2s \max_k \{\text{Re}(\lambda_k)\}} \|AB + BA^\dagger\|_1 \quad (\text{C36})$$

$$\leq 2\|A\| e^{2Rs}, \quad (\text{C37})$$

since $\|XYZ\|_1 \leq \|XY\|_1 \|Z\| \leq \|X\| \|Y\|_1 \|Z\|$ by two applications of a corollary of von Neumann's trace inequality (follows from Ref. [27, Theorem 7.4.1.3(d)]). In the last step we also use von Neumann's trace inequality and $\|B\|_1 = 1$. We therefore get

$$\|S(\infty) - \mathcal{I}(\infty)\|_1 \leq 2\|A\| \sum_{l=0}^{\infty} \int_{\Delta l}^{\Delta(l+1)} \left(\int_{\Delta l}^t e^{2Rs} ds \right) dt \quad (\text{C38})$$

$$\leq 2\|A\| \sum_{l=0}^{\infty} e^{2R\Delta l} \int_{\Delta l}^{\Delta(l+1)} (t - \Delta l) dt \quad (\text{C39})$$

$$= \|A\| \Delta^2 \sum_{l=0}^{\infty} e^{2R\Delta l} \quad (\text{C40})$$

$$= \frac{\|A\| \Delta^2}{1 - e^{2R\Delta}}. \quad (\text{C41})$$

b. Bounding denominator

We have

$$\text{tr}[S(\infty)] = \Delta \sum_j B_{jj} \sum_{l=0}^{\infty} e^{2\Delta l \text{Re}(\lambda_j)} \quad (\text{C42})$$

$$\begin{aligned} &= \sum_j \frac{B_{jj} \Delta}{1 - e^{2\Delta \text{Re}(\lambda_j)}} \geq \min_j \left\{ \frac{\Delta}{1 - e^{2\Delta \text{Re}(\lambda_j)}} \right\} \\ &= \frac{\Delta}{1 - \min_j \{e^{2\Delta \text{Re}(\lambda_j)}\}} = \frac{\Delta}{1 - e^{2r\Delta}}, \end{aligned} \quad (\text{C43})$$

where we use $\text{tr}(B) = 1$ and introduce the notation $r := \min_j \{\text{Re}(\lambda_j)\} \leq R < 0$. Note that $(1 - e^y)/(-y)$ is a positive strictly increasing function in y , which means that $(1 - e^{2r\Delta})/(-2r\Delta) \leq (1 - e^{2R\Delta})/(-2R\Delta)$. By combining the bound for the numerator and denominator, we get that

$$\begin{aligned} \frac{1}{2} \left\| \frac{\mathcal{I}(\infty)}{\text{tr}[\mathcal{I}(\infty)]} - \frac{S(\infty)}{\text{tr}[S(\infty)]} \right\|_1 &\leq \|A\| \Delta \frac{1 - e^{2r\Delta}}{1 - e^{2R\Delta}} \\ &\leq \|A\| \Delta \frac{r}{R}. \end{aligned} \quad (\text{C44})$$

For this distance to be at most ϵ_1 , we can choose

$$\Delta \leq \frac{\epsilon_1}{\|A\|} \frac{\max_j \{\text{Re}(\lambda_j)\}}{\min_k \{\text{Re}(\lambda_k)\}}. \quad (\text{C45})$$

Recall that $T = (L/\Delta) - 1$. Combining our choices for L , Eq. (C30), and Δ , Eq. (C45), we arrive at

$$T \geq \left\lceil \frac{1}{2\epsilon_1} \ln \left(\frac{1}{\epsilon_2} \right) \frac{\|A\| |\min_k \{\text{Re}(\lambda_k)\}|}{|\max_j \{\text{Re}(\lambda_j)\}|^2} \right\rceil. \quad (\text{C46})$$

Note that $|\min_k \{\text{Re}(\lambda_k)\}| \leq \|A\|$, with equality for Hermitian A . In our use context, the factor $\|A\|/|\max_j \{\text{Re}(\lambda_j)\}|$ is similar to the condition number κ . Indeed if A is Hermitian, and recalling that all its eigenvalues are negative, then $|\max_j \{\text{Re}(\lambda_j)\}| = \min_j \{|\lambda_j|\}$, so that $T \geq \lceil (\kappa^2/2\epsilon_1) \ln(1/\epsilon_2) \rceil$. ■

3. Robustness

In this section we discuss the effect of approximations when implementing the algorithm. Conceptually, it is straightforward to include such error in the analysis using the triangle inequality. Let $\sim$ mark approximate quantities. For the discrete Lyapunov equation, Eq. (C12) is replaced by

$$\begin{aligned} \frac{1}{2} \left\| \frac{X}{\text{tr}(X)} - \mathbb{E}[\rho] \right\|_1 &= \frac{1}{2} \left\| \frac{S(\infty)}{\text{tr}[S(\infty)]} - \frac{S(T)}{\text{tr}[S(T)]} \right\|_1 \\ &\leq \epsilon \\ &\quad + \frac{1}{2} \left\| \frac{S(T)}{\text{tr}[S(T)]} - \frac{\tilde{S}(T)}{\text{tr}[\tilde{S}(T)]} \right\|_1, \end{aligned} \quad (\text{C47})$$

where T is still given by Eq. (C18), and we introduce a parameter $\epsilon_{\sim} \geq 0$ to bound the implementation error. In the continuous case, Eq. (C28) is replaced by

$$\frac{1}{2} \left\| \frac{X}{\text{tr}(X)} - \mathbb{E}[\rho] \right\|_1 \leq \underbrace{\frac{1}{2} \left\| \frac{\mathcal{I}(\infty)}{\text{tr}[\mathcal{I}(\infty)]} - \frac{\mathcal{S}(\infty)}{\text{tr}[\mathcal{S}(\infty)]} \right\|_1}_{\leq \epsilon_1} + \underbrace{\frac{1}{2} \left\| \frac{\mathcal{S}(\infty)}{\text{tr}[\mathcal{S}(\infty)]} - \frac{\mathcal{S}(L)}{\text{tr}[\mathcal{S}(L)]} \right\|_1}_{\leq \epsilon_2} + \underbrace{\frac{1}{2} \left\| \frac{\mathcal{S}(L)}{\text{tr}[\mathcal{S}(L)]} - \frac{\tilde{\mathcal{S}}(L)}{\text{tr}[\tilde{\mathcal{S}}(L)]} \right\|_1}_{\leq \epsilon_{\sim}}, \quad (\text{C48})$$

and $L = (T + 1)\Delta$ is still given by the parameters in Theorem 2.

In the remainder of the section we inspect the newly added error term in order to obtain a more practical expression. We begin by relating the error term to the approximate implementation $\tilde{\mathcal{E}}$ of CP map $\mathcal{E}$. For the discrete-time Lyapunov equation we use Claim 1 to obtain

$$\frac{1}{2} \left\| \frac{\mathcal{S}(T)}{\text{tr}[\mathcal{S}(T)]} - \frac{\tilde{\mathcal{S}}(T)}{\text{tr}[\tilde{\mathcal{S}}(T)]} \right\|_1 \leq \frac{\|\mathcal{S}(T) - \tilde{\mathcal{S}}(T)\|_1}{\text{tr}[\mathcal{S}(T)]} = \frac{\left\| \sum_{k=0}^T \mathcal{E}^k(B) - \tilde{\mathcal{E}}^k(B) \right\|_1}{\sum_{l=0}^T \text{tr}[\mathcal{E}^l(B)]} \leq \sum_{k=0}^T \left\| \mathcal{E}^k(B) - \tilde{\mathcal{E}}^k(B) \right\|_1. \quad (\text{C49})$$

For the denominator, we use that $\sum_{l=0}^T \text{tr}[\mathcal{E}^l(B)] \geq \text{tr}[\mathcal{E}^0(B)] = \text{tr}(B) = 1$. In the continuous-time case the numerator and denominator of Eq. (C49) acquire a factor Δ , which cancels, and we can also plug $T = (L/\Delta) - 1$ to obtain an identical expression. Thus the following steps are valid for both continuous- and discrete-time cases.

Let us now consider the diamond norm of a linear map $\mathcal{A} : \mathbb{C}^{d \times d} \rightarrow \mathbb{C}^{d \times d}$, which is defined as $\|\mathcal{A}\|_{\diamond} = \max\{\|(\mathcal{A} \otimes I_d)(X)\|_1 \mid X \in \mathbb{C}^{2d \times 2d}, \|X\|_1 \leq 1\}$. The diamond norm is submultiplicative under composition of maps, $\|\mathcal{B} \circ \mathcal{A}\|_{\diamond} \leq \|\mathcal{B}\|_{\diamond} \|\mathcal{A}\|_{\diamond}$, thus

$$\begin{aligned} \|\mathcal{B} \circ \mathcal{A} - \mathcal{D} \circ \mathcal{C}\|_{\diamond} \\ \leq \|\mathcal{B} \circ \mathcal{A} - \mathcal{B} \circ \mathcal{C}\|_{\diamond} + \|\mathcal{B} \circ \mathcal{C} - \mathcal{D} \circ \mathcal{C}\|_{\diamond} \end{aligned} \quad (\text{C50})$$

$$\leq \|\mathcal{B}\|_{\diamond} \|\mathcal{A} - \mathcal{C}\|_{\diamond} + \|\mathcal{B} - \mathcal{D}\|_{\diamond} \|\mathcal{C}\|_{\diamond}. \quad (\text{C51})$$

For trace nonincreasing maps, e.g., $\|(\mathcal{B} \otimes I_d)(X)\|_1 \leq \|X\|_1 \leq 1$, we have that $\|\mathcal{B}\|_{\diamond} \leq 1$. So Eq. (C51) indicates subadditivity with respect to composition of trace nonincreasing maps. Using this fact recursively in Eq. (C49) we have

$$\begin{aligned} \sum_{k=0}^T \left\| \mathcal{E}^k(B) - \tilde{\mathcal{E}}^k(B) \right\|_1 &\leq \sum_{k=0}^T \left\| \mathcal{E}^k - \tilde{\mathcal{E}}^k \right\|_{\diamond} \\ &\leq \sum_{k=0}^T k \left\| \mathcal{E} - \tilde{\mathcal{E}} \right\|_{\diamond} \leq \frac{T(T+1)}{2} \left\| \mathcal{E} - \tilde{\mathcal{E}} \right\|_{\diamond}. \end{aligned} \quad (\text{C52})$$

In the second inequality we use $\|\mathcal{F}(X)\|_1 \leq \|\mathcal{F}\|_{\diamond} \|X\|_1$. Recall that our algorithm utilizes an approximate block encoding of the Kraus operator M to produce an approximate implementation of the map $\mathcal{E}(\cdot) = M \cdot M^{\dagger}$. Let us recap how this works.

Suppose we have a block encoding $U_{\tilde{M}}$ such that $\tilde{M} = (\langle G| \otimes I^{\otimes n}) U_{\tilde{M}} (|G\rangle \otimes I^{\otimes n})$ for a known state $|G\rangle$. Note the absence of subnormalization constants as these could bias the transition probabilities of our algorithm (in Appendix C4 we show how to produce such block encoding in practice). We now attach the ancilla register $|G\rangle\langle G|$ to the system register ρ and apply $U_{\tilde{M}}$, producing $U_{\tilde{M}}(|G\rangle\langle G| \otimes \rho) U_{\tilde{M}}^{\dagger}$. We then perform a projective measurement on the ancilla register with elements $\{|G\rangle\langle G|, I - |G\rangle\langle G|\}$. If the measurement outcome is 1, we obtain $|G\rangle\langle G| \otimes \tilde{\mathcal{E}}(\rho) / \text{tr}[\tilde{\mathcal{E}}(\rho)]$. Discarding the ancilla register, we obtain the desired approximation and the algorithm enters the “continue” branch in Eq. (B1). If the measurement outcome is 0, the algorithm enters the “restart” branch. So how does the error in M relates to the error in $\mathcal{E}$? We have that

$$\|\mathcal{E} - \tilde{\mathcal{E}}\|_{\diamond} = \max_{\rho} \|(\mathcal{E} \otimes I_d)(\rho) - (\tilde{\mathcal{E}} \otimes I_d)(\rho)\|_1 \quad (\text{C53})$$

$$= \max_{\rho} \|(M \otimes I_d)\rho(M \otimes I_d)^{\dagger} - (\tilde{M} \otimes I_d)\rho(\tilde{M} \otimes I_d)^{\dagger}\|_1 \quad (\text{C54})$$

$$= \max_{\rho} \|(M - \tilde{M} \otimes I_d)\rho(M \otimes I_d)^{\dagger} + (M \otimes I_d)\rho(M - \tilde{M} \otimes I_d)^{\dagger} - (M - \tilde{M} \otimes I_d)\rho(M - \tilde{M} \otimes I_d)^{\dagger}\|_1 \quad (\text{C55})$$

$$\begin{aligned} &\leq \max_{\rho} \left\{ \|(M - \tilde{M} \otimes I_d) \rho (M \otimes I_d)^\dagger\|_1 + \|(M \otimes I_d) \rho (M - \tilde{M} \otimes I_d)^\dagger\|_1 \right. \\ &\quad \left. + \|(M - \tilde{M} \otimes I_d) \rho (M - \tilde{M} \otimes I_d)^\dagger\|_1 \right\} \end{aligned} \quad (\text{C56})$$

$$\leq \max_{\rho} \left\{ 2 \|M - \tilde{M} \otimes I_d\| \|\rho\|_1 \|M \otimes I_d\| + \|M - \tilde{M} \otimes I_d\|^2 \|\rho\|_1 \right\} \quad (\text{C57})$$

$$\leq 2 \|M - \tilde{M} \otimes I_d\| \|M \otimes I_d\| + \|M - \tilde{M} \otimes I_d\|^2 \quad (\text{C58})$$

$$\leq 2 \|M - \tilde{M}\| + \|M - \tilde{M}\|^2. \quad (\text{C59})$$

Here we use $\|XYZ\|_1 \leq \|X\| \|Y\| \|Z\|$ [see comment after Eq. (C37)], $\|X^\dagger\| = \|X\|$, $\|\rho\|_1 = 1$ and $\|X \otimes I_d\| = \|X\|$. In the final step we also use that $\|M\| \leq 1$ because M is the Kraus operator of a trace nonincreasing map.

Putting together Eqs. (C49), (C52), and (C59) we arrive at

$$\begin{aligned} &\frac{1}{2} \left\| \frac{\mathcal{S}(T)}{\text{tr}[\mathcal{S}(T)]} - \frac{\tilde{\mathcal{S}}(T)}{\text{tr}[\tilde{\mathcal{S}}(T)]} \right\|_1 \\ &\leq T(T+1) \left(\|M - \tilde{M}\| + \frac{1}{2} \|M - \tilde{M}\|^2 \right) \leq \epsilon_{\sim}. \end{aligned} \quad (\text{C60})$$

In the trivial case $T = 0$ the algorithm never uses the CP map so the block-encoding error is irrelevant. When $T \neq 0$ instead, solving for the block-encoding error we see that $\|M - \tilde{M}\| \leq \sqrt{1 + (2\epsilon_{\sim}/T(T+1))} - 1$ with T given in Theorem 1 for the discrete case and Theorem 2 for the continuous case. This expression is rather complicated, so we make one further approximation. Noting that $x/4 \leq \sqrt{1+x} - 1$ for $0 \leq x \leq 8$, we conclude that

$$\begin{aligned} \|M - \tilde{M}\| &\leq \frac{\epsilon_{\sim}}{2T(T+1)} \\ \Rightarrow \frac{1}{2} \left\| \frac{\mathcal{S}(T)}{\text{tr}[\mathcal{S}(T)]} - \frac{\tilde{\mathcal{S}}(T)}{\text{tr}[\tilde{\mathcal{S}}(T)]} \right\|_1 &\leq \epsilon_{\sim}. \end{aligned} \quad (\text{C61})$$

4. Block encoding the Kraus operator for the continuous-time Lyapunov equation

In the previous section we showed how the implementation error is related to the block-encoding error of the Kraus operator M . Here we provide an explicit construction for the continuous-time Lyapunov equation, where we specifically need $M = e^{\Delta A}$ (matrix inversion II is solved via a reduction to the continuous-time Lyapunov equation and thus the construction is very similar). For simplicity, we assume $A < 0$, $\Delta > 0$, and that we are given a block encoding of $-A/\|A\|$. That is, we have access to a circuit U_A such that $-A/\|A\| = (\langle G| \otimes I^{\otimes n}) U_A (|G\rangle \otimes I^{\otimes n})$

for some state $|G\rangle$. We now construct the block encoding of M using the quantum singular value transformation (QSVT) [24].

QSVT is a technique to apply a polynomial to a block-encoded matrix with query complexity proportional to the degree of the polynomial. In its basic version, the polynomial must have a definite parity (even or odd) and be bounded, $|P(x)| \leq 1$ for $x \in [-1, 1]$. Our application calls for the use of a polynomial approximation to $e^{-\beta x}$. This is discussed in Ref. [24] where, however, a subnormalization factor is used to ensure that the polynomial remains bounded. The subnormalization factor is related to the probability of success of the block encoding upon projective measurement of the ancilla register. The subnormalization factor can therefore bias the carefully designed transition probabilities in our algorithm. In the QSVT literature, the subnormalization factor is often removed using the amplitude amplification algorithm, but this requires quantum circuits that perform reflections with respect to the initial state. Due to the probabilistic nature of our algorithm, the system register is in an unknown mixed state at all times, and thus we cannot construct such reflections. In the following we present a solution that block encodes our operator M without any subnormalization.

Let us define $H := -A/\|A\| > 0$ and $\beta := \Delta\|A\|$, so that $e^{-\beta|H|} = e^{\Delta A} = M$. The condition number is $\kappa := \kappa(H) = \kappa(A)$. We now seek a polynomial approximation to the function $f(x) = e^{-\beta|x|}$, which is accurate in the interval $[1/\kappa, 1]$ where the spectrum of H lies. We resort to the following result.

Theorem C1 (Restatement of Theorem 68 in Ref. [24], bounded polynomial approximation based on multiple local Taylor series). Let $J \in \mathbb{N}$, $(x_j, r_j, \mu_j) \in [-1, 1]^J \times (0, 2]^J \times (0, 1]^J$, such that $x_j : j \in [J]$ is monotone increasing, and $\mu_j \leq r_j$ for all $j \in [J]$. Let $I := \bigcup_{j \in [J]} [x_j - r_j, x_j + r_j]$ be the union of the intervals $[x_j - r_j, x_j + r_j]$, and suppose that for all $i < j \in [J]$ such that $j - i \geq 2$ we have that $x_i + r_i < x_j - r_j$. Let $\mu = \min[\min_{j \in [J]} \mu_j, \min_{j \in [J-1]} |x_{j+1} - r_{j+1} - x_j - r_j|]$. Let $f : I + [-\mu/2, \mu/2] \rightarrow \mathbb{C}$, $B \in \mathbb{R}_+$ be such that for all $j \in [J]$ we have $f(x_j + x) = \sum_{k=0}^{\infty} a_k^{(j)} x^k$ for all

$x \in [-r_j - (\mu_j/2), r_j + (\mu_j/2)]$ and $\sum_{k=0}^{\infty} (r_j + \mu_j)^k |a_k^{(j)}| \leq B$. Let $\delta \in (0, 1/2BJ]$, then there is an efficiently computable polynomial $P \in \mathbb{C}[x]$ of degree $\mathcal{O}[(J/\mu) \log(BJ/\delta)]$ such that

$$\max_{x \in I} |f(x) - P(x)| \leq \delta, \quad (\text{C62})$$

$$\max_{x \in [-1, 1]} |P(x)| \leq \max_{x \in I + [-\mu/2, \mu/2]} |f(x)|, \quad (\text{C63})$$

$$\max_{x \in [-1, 1] \setminus (I + [-\mu/2, \mu/2])} |P(x)| \leq \delta. \quad (\text{C64})$$

Let us choose $J = 2$, and set $(x_1, r_1, \mu_1) = (-1, 1 - 1/\kappa, 1/2\kappa)$ and $(x_2, r_2, \mu_2) = (1, 1 - 1/\kappa, 1/2\kappa)$. First, we Taylor expand the functions $f_1: (-\infty, 0) \rightarrow \mathbb{R}, x \mapsto e^{\beta x}$ around $x_1 = -1$ and $f_2: (0, \infty) \rightarrow \mathbb{R}, x \mapsto e^{-\beta x}$ around $x_2 = 1$ (both functions are analytic on their respective open domains):

$$\begin{aligned} f_1(x) &= \sum_{\ell=0}^{\infty} \frac{f_1^{(\ell)}(-1)}{\ell!} (x+1)^\ell = \sum_{\ell=0}^{\infty} \frac{\beta^\ell e^{-\beta}}{\ell!} (x+1)^\ell \\ \Leftrightarrow f_1(-1+x) &= \sum_{\ell=0}^{\infty} \frac{\beta^\ell e^{-\beta}}{\ell!} x^\ell \quad \text{for } x \in (-\infty, 1), \end{aligned} \quad (\text{C65})$$

$$\begin{aligned} f_2(x) &= \sum_{\ell=0}^{\infty} \frac{f_2^{(\ell)}(1)}{\ell!} (x-1)^\ell = \sum_{\ell=0}^{\infty} \frac{(-\beta)^\ell e^{-\beta}}{\ell!} (x-1)^\ell \\ \Leftrightarrow f_2(1+x) &= \sum_{\ell=0}^{\infty} \frac{(-\beta)^\ell e^{-\beta}}{\ell!} x^\ell \quad \text{for } x \in (-1, \infty). \end{aligned} \quad (\text{C66})$$

Now we consider the intervals $[x_j - r_j - \mu_j/2, x_j + r_j + \mu_j/2]$ for $j = 1, 2$:

$$I_1 = \left[-2 + \frac{3}{4\kappa}, -\frac{3}{4\kappa}\right], \quad (\text{C67})$$

$$I_2 = \left[\frac{3}{4\kappa}, 2 - \frac{3}{4\kappa}\right]. \quad (\text{C68})$$

Our target function $f(x) = e^{-\beta|x|}$ matches $f_1(x)$ for $x \in I_1 \subset (-\infty, 0)$ and $f_2(x)$ for $x \in I_2 \subset (0, \infty)$. Hence, we can express the target function using the expansions in Eqs. (C65) and (C66) on the intervals I_1 and I_2 , respectively, as required by the theorem:

$$\begin{aligned} f(-1+x) &= \sum_{\ell=0}^{\infty} \frac{\beta^\ell e^{-\beta}}{\ell!} x^\ell \\ \text{for } x \in \left[-1 + \frac{3}{4\kappa}, 1 - \frac{3}{4\kappa}\right] &\subset (-\infty, 1), \end{aligned} \quad (\text{C69})$$

$$\begin{aligned} f(1+x) &= \sum_{\ell=0}^{\infty} \frac{(-\beta)^\ell e^{-\beta}}{\ell!} x^\ell \\ \text{for } x \in \left[-1 + \frac{3}{4\kappa}, 1 - \frac{3}{4\kappa}\right] &\subset (-1, \infty). \end{aligned} \quad (\text{C70})$$

This defines the coefficients $a_\ell^{(1)} = \beta^\ell e^{-\beta}/\ell!$ and $a_\ell^{(2)} = (-\beta)^\ell e^{-\beta}/\ell!$ for all $\ell \geq 0$, and the other parameters

$$\mu = \frac{1}{2\kappa}, \quad (\text{C71})$$

$$I = \left[-2 + \frac{1}{\kappa}, -\frac{1}{\kappa}\right] \cup \left[\frac{1}{\kappa}, 2 - \frac{1}{\kappa}\right], \quad (\text{C72})$$

$$\begin{aligned} B &= \sum_{\ell=0}^{\infty} \left(1 - \frac{1}{\kappa} + \frac{1}{2\kappa}\right)^\ell \left| \frac{(\mp\beta)^\ell e^{-\beta}}{\ell!} \right| \\ &= e^{-\beta} \sum_{\ell=0}^{\infty} \frac{\beta^\ell (1 - \frac{1}{2\kappa})^\ell}{\ell!} = e^{-\frac{\beta}{2\kappa}}. \end{aligned} \quad (\text{C73})$$

By the theorem, for $\delta \in (0, 1/4B]$ there exists an efficiently computable polynomial such that

$$\max_{x \in I} |e^{-\beta|x|} - P(x)| \leq \delta, \quad \max_{x \in [-1, 1]} |P(x)| \leq e^{-\frac{3\beta}{4\kappa}} \leq 1. \quad (\text{C74})$$

Since P is complex and not guaranteed to have a definite parity, it cannot be directly implemented by QSVT. Let us then take the real, even part of the polynomial, $Q(x) = [P(x) + P(-x) + P(x)^* + P(-x)^*]/4$. Noting that the target function is real and even, we can also write $f(x) = [f(x) + f(-x) + f(x)^* + f(-x)^*]/4$. Simple applications of the triangle inequality show that $\max_{x \in I} |f(x) - Q(x)| \leq \delta$, and that $\max_{x \in [-1, 1]} |Q(x)| \leq 1$. This polynomial can be implemented by QSVT using Theorem 17 and Corollary 10 in Ref. [24]. More specifically, we can construct a block encoding $U_{\tilde{M}}$ such that $\tilde{M} = Q(-A/\|A\|) = (\langle G| \otimes I^{\otimes n}) U_{\tilde{M}} (|G\rangle \otimes I^{\otimes n})$ with the same state $|G\rangle$ used in the block encoding U_A . By Theorem C1 the query complexity, i.e., the number of uses of U_A , is proportional to the degree of the polynomial Q . That is, $\mathcal{O}\left[\frac{J}{\mu} \log\left(\frac{BJ}{\delta}\right)\right] = \mathcal{O}\left[\kappa \log\left(\frac{1}{\delta}\right)\right]$.

Specifically choosing $\delta = \epsilon_{\sim}/2T(T+1)$ we arrive at $\|M - \tilde{M}\| \leq \epsilon_{\sim}/2T(T+1)$ as required by Eq. (C61). The total number of queries to U_A is then $\mathcal{O}[\kappa \log(T/\epsilon_{\sim})]$. Note, however, that Theorem C1 requires $\delta \leq 1/4B$. Plugging our parameters in we find that the construction is valid for $\epsilon_{\sim} \leq (T(T+1)/2)e^{\Delta\|A\|/2\kappa}$.

APPENDIX D: LOWER BOUND FOR THE DISCRETE-TIME ALGORITHM

We can show that the choice of $T^* := \lceil \frac{1}{2} \ln(1/\epsilon) / \ln(1/\|A\|) \rceil$ in Theorem 1 is asymptotically optimal. In

particular, the following theorem implies that, if we had chosen $T^* < \frac{1}{4} \ln(1/\epsilon) / \ln(10/\|A\|)$ in Theorem 1, then it would have been false.

Theorem D1. For every $1/2 \leq \lambda < 1$, there exists a normal matrix A of operator norm λ and a state B such that the matrix $\mathcal{S}(T)$ defined in Eq. (C7) satisfies

$$\frac{1}{2} \left\| \frac{\mathcal{S}(\infty)}{\text{tr}[\mathcal{S}(\infty)]} - \frac{\mathcal{S}(T)}{\text{tr}[\mathcal{S}(T)]} \right\|_1 \geq \frac{\lambda^{4T}}{10}. \quad (\text{D1})$$

Note that Theorem D1 does not rule out the existence of an algorithm for solving the discrete-time Lyapunov equation that is more efficient than Algorithm 1. That is, an algorithm that produces a state ρ satisfying $\frac{1}{2} \|X/\text{tr}(X) - \rho\|_1 \leq \epsilon$ while consuming less copies of the initial state $\rho_0 = B$ and with less calls to the block encoding of A .

Proof of Theorem D1. We choose the maximally mixed state for a single qubit, $B = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$, and

$$A := \sqrt{\mu} |0\rangle\langle 0| + \sqrt{\nu} |1\rangle\langle 1|, \quad (\text{D2})$$

with $0 < \nu < \mu < 1$ and $\mu = \lambda^2$. Note that the operator norm of A is λ , as required. Eventually, after Eq. (D14), we will effectively choose $\nu := (3\lambda^2 - 1)/2$. For A and B chosen as above, from Eq. (C7) we see that

$$\begin{aligned} \mathcal{S}(T) &= \frac{1}{2} \sum_{k=0}^T (\mu^k |0\rangle\langle 0| + \nu^k |1\rangle\langle 1|) \\ &= \frac{1}{2} \left(\frac{1 - \mu^{T+1}}{1 - \mu} |0\rangle\langle 0| + \frac{1 - \nu^{T+1}}{1 - \nu} |1\rangle\langle 1| \right). \end{aligned} \quad (\text{D3})$$

For brevity, define $\eta(T) := \mathcal{S}(T)/\text{tr}[\mathcal{S}(T)]$, so we have

$$\begin{aligned} \eta(T) &= \frac{\frac{1-\mu^{T+1}}{1-\mu} |0\rangle\langle 0| + \frac{1-\nu^{T+1}}{1-\nu} |1\rangle\langle 1|}{\frac{1-\mu^{T+1}}{1-\mu} + \frac{1-\nu^{T+1}}{1-\nu}} \\ &= \frac{|0\rangle\langle 0|}{1 + \frac{1-\mu}{1-\nu} \frac{1-\nu^{T+1}}{1-\mu^{T+1}}} + \frac{|1\rangle\langle 1|}{1 + \frac{1-\nu}{1-\mu} \frac{1-\mu^{T+1}}{1-\nu^{T+1}}}. \end{aligned} \quad (\text{D4})$$

Because $\eta(T) - \eta(\infty)$ is a Hermitian matrix of trace 0 and it is diagonal in the basis $\{|0\rangle, |1\rangle\}$, its diagonal entries have the same absolute value yet opposite signs. Hence, by symmetry we have

$$\frac{\|\eta(T) - \eta(\infty)\|_1}{2} = |\langle 0 | [\eta(T) - \eta(\infty)] | 0 \rangle| \quad (\text{D5})$$

$$= \frac{1}{1 + \frac{1-\mu}{1-\nu}} - \frac{1}{1 + \frac{1-\mu}{1-\nu} \frac{1-\nu^{T+1}}{1-\mu^{T+1}}} \quad (\text{D6})$$

$$= \frac{\frac{1-\mu}{1-\nu} \left(\frac{1-\nu^{T+1}}{1-\mu^{T+1}} - 1 \right)}{\left(1 + \frac{1-\mu}{1-\nu} \right) \left(1 + \frac{1-\mu}{1-\nu} \frac{1-\nu^{T+1}}{1-\mu^{T+1}} \right)} \quad (\text{D7})$$

$$= \frac{(1-\mu)(1-\nu)(\mu^{T+1} - \nu^{T+1})}{(1-\nu + 1-\mu) \left[(1-\nu)(1-\mu^{T+1}) + (1-\mu)(1-\nu^{T+1}) \right]}. \quad (\text{D8})$$

Let $0 < \delta \leq 1/2$ be such that $\mu = 1 - \delta$ and, in turn, $c > 1$ be such that $\nu = 1 - c\delta$. Hence we have

$$\frac{\|\eta(T) - \eta(\infty)\|_1}{2} = \frac{c\delta^2 [(1-\delta)^{T+1} - (1-c\delta)^{T+1}]}{(c\delta + \delta) \{ c\delta [1 - (1-\delta)^{T+1}] + \delta [1 - (1-c\delta)^{T+1}] \}} \quad (\text{D9})$$

$$= \frac{c [(1-\delta)^{T+1} - (1-c\delta)^{T+1}]}{(c+1) [c + 1 - c(1-\delta)^{T+1} - (1-c\delta)^{T+1}]}. \quad (\text{D10})$$

For bounding the numerator of Eq. (D10), we will use

$$(1-\delta)^{T+1} - (1-c\delta)^{T+1} = [(1-\delta) - (1-c\delta)] \sum_{k=0}^T (1-\delta)^k (1-c\delta)^{T-k} \quad (\text{D11})$$

$$\geq (c-1)\delta \sum_{k=0}^T (1-c\delta)^k (1-c\delta)^{T-k} = (c-1)\delta(T+1)(1-c\delta)^T. \quad (\text{D12})$$

Above, the former equality is analogous to the equality $1 - x^{T+1} = (1 - x) \sum_{k=0}^T x^k$ concerning the finite geometric series where we have taken $x := (1 - c\delta)/(1 - \delta)$. For bounding the denominator of Eq. (D10), we will use

$$\begin{aligned} c(1 - \delta)^{T+1} + (1 - c\delta)^{T+1} \\ \geq c[1 - \delta(T+1)] + [1 - c\delta(T+1)] \\ = c + 1 - 2c\delta(T+1). \end{aligned} \quad (\text{D13})$$

By incorporating the two inequalities, we get

$$\begin{aligned} \frac{\|\eta(T) - \eta(\infty)\|_1}{2} &\geq \frac{c(c-1)\delta(T+1)(1-c\delta)^T}{(c+1)[c+1-c-1+2c\delta(T+1)]} \\ &= \frac{(c-1)(1-c\delta)^T}{2(c+1)}. \end{aligned} \quad (\text{D14})$$

We choose $c := \frac{3}{2}$, for which we have $\nu = 1 - c\delta \geq (1 - \delta)^2 = \mu^2 = \lambda^4$ because $\delta \leq 1/2$. Thus, we get

$$\frac{\|\eta(T) - \eta(\infty)\|_1}{2} \geq \frac{(\frac{3}{2} - 1)\lambda^{4T}}{2(\frac{3}{2} + 1)} = \frac{\lambda^{4T}}{10}, \quad (\text{D15})$$

as claimed. $\blacksquare$

APPENDIX E: ESTIMATING EXPECTATION VALUES, OVERLAPS, AND NORMALIZATION CONSTANTS

Here we discuss how to read-off information from the output state of the quantum algorithm. Let us assume ρ is a state generated by the probabilistic algorithm, e.g., encoding the solution to a Lyapunov equation, or a matrix inverse. It is straightforward to measure the expectation $\text{tr}(\rho O)$ of simple observables, such as Pauli matrices $O \in \{I, \sigma_X, \sigma_Y, \sigma_Z\}^{\otimes n}$ or a linear combination thereof $O = \sum_i a_i O_i$ where $a_i \in \mathbb{R}$.

We can also interpret ρ itself as an observable. The task is then to estimate the expectation $\langle \psi | \rho | \psi \rangle$ with respect to another state $|\psi\rangle$. If we know the state-preparation circuit $U|0\rangle = |\psi\rangle$ then we can simply prepare $U^\dagger \rho U$ and measure the overlap with the zero state. If we do not know U , but have access to copies of $|\psi\rangle$ then we can use the well-known SWAP test as follows. We start the computation from $|+\rangle\langle+| \otimes \rho \otimes |\psi\rangle\langle\psi|$, where the first register is an ancilla qubit in state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. We apply a circuit C_S that swaps the second and third registers if the ancilla is in state $|1\rangle$. Finally, we measure the $\sigma_X \otimes I \otimes I$ observable. It can be verified that

$$\begin{aligned} \text{tr}\left[(\sigma_X \otimes I \otimes I)C_S(|+\rangle\langle+| \otimes \rho \otimes |\psi\rangle\langle\psi|)C_S^\dagger\right] \\ = \langle \psi | \rho | \psi \rangle. \end{aligned} \quad (\text{E1})$$

It is also possible to estimate matrix entries $\langle \phi | \rho | \psi \rangle$. If we know the state-preparation circuits $U|0\rangle = |\psi\rangle$ and

$V|0\rangle = |\phi\rangle$ we can use the well-known Hadamard test as follows. We start from $|+\rangle\langle+| \otimes \rho$ and use a circuit $C_{U,V}$ that applies $U^\dagger$ if the ancilla is in state $|1\rangle$, and $V^\dagger$ if it is in state $|0\rangle$. Finally, we measure $\sigma_X \otimes |0\rangle\langle 0|$. It can be verified that

$$\text{tr}\left[(\sigma_X \otimes |0\rangle\langle 0|)C_{U,V}(|+\rangle\langle+| \otimes \rho)C_{U,V}^\dagger\right] = \text{Re} \langle \phi | \rho | \psi \rangle. \quad (\text{E2})$$

The imaginary part is obtained by measuring $\sigma_Y \otimes |0\rangle\langle 0|$ instead.

Another interesting task is the estimation of the normalization constant, i.e., the denominator in Eq. (B9). This quantity can be interpreted as the probability that the algorithm stops without ever performing a restart (see Fig. 1 for intuition). This fact is used in Ref. [18, Proposition 11] to estimate the partition function of Gibbs states. More generally, suppose we execute the algorithm n_{runs} times and keep track of the total number of restarts n_{restarts} . Then

$$\begin{aligned} \sum_{k=0}^T r_k R_k \text{tr}[\mathcal{E}^k(\rho_0)] \\ = \sum_{k=0}^T \mathbb{P}(\text{stop at iteration } k \mid \text{algorithm not restarted}) \\ \approx \frac{n_{\text{runs}}}{n_{\text{runs}} + n_{\text{restarts}}}. \end{aligned} \quad (\text{E3})$$

As an example of application, consider the estimation of $\text{tr}(X)$ where X is the solution to the discrete-time Lyapunov equation $AXA^\dagger - X + B = 0$. From Theorem 1 the expected output state of the algorithm is ϵ -close in trace distance to the normalized solution. From Appendix C 1 we have that the outputs state is normalized by $\sum_{k=0}^T r_k R_k \text{tr}[\mathcal{E}^k(\rho_0)] = \text{tr}[\mathcal{S}(T)]$, and thus we can use Eq. (E3) as an estimator. Inspecting Eq. (C17) in the proof of the theorem we find an upper bound on the relative error of the normalization constant:

$$\begin{aligned} \frac{1}{2} \left\| \frac{X}{\text{tr}(X)} - \mathbb{E}[\rho] \right\|_1 &\leq \frac{\text{tr}[\mathcal{S}(\infty)] - \text{tr}[\mathcal{S}(T)]}{\text{tr}[\mathcal{S}(\infty)]} \\ &= \left| \frac{\text{tr}(X) - \text{tr}[\mathcal{S}(T)]}{\text{tr}(X)} \right| (\text{relative error}) \\ &\leq \|A\|^{2(T+1)} \leq \epsilon. \end{aligned} \quad (\text{E4})$$

Thus the relative bias of the estimator decreases exponentially in T .

APPENDIX F: COMPARISON OF ALGORITHMS FOR THE CONTINUOUS-TIME LYAPUNOV EQUATION

Here we discuss four quantum algorithms for solving the continuous-time Lyapunov equation. The algorithms

TABLE II. Quantum algorithms for the continuous-time Lyapunov equation $AX + XA + B = 0$. For all algorithms we assume negative definite A with $\|A\| = 1/2$ and B positive semidefinite with $\text{tr}[B] = 1$. A (α, ϵ) block encoding U_X of X means $\|X - \alpha \Pi U_X \Pi\| \leq \epsilon$ for a suitable projector Π . Query complexity is in terms of access to the stated input oracles.

Algorithm	Input	Output	Primitives	Runtime or query complexity
[11]	s -sparse matrix oracle for A , state preparation of $ \text{vec}(-B)\rangle$	Pure state $ \tilde{x}\rangle$ with $\ \tilde{x}\rangle - \text{vec}(X)\rangle \ _2 \leq \epsilon$	HHL	$\tilde{O}\left(\frac{\log(N)s^2\kappa^2}{\epsilon}\right)^a$
[15, Theorem 4]	Block encoding of A, B	$(\ B\ , \epsilon)$ block encoding of X	LCU, imaginary-time Hamiltonian simulation	$\tilde{O}\left(\sqrt{\frac{\ B\ \kappa^5}{\epsilon}}\right)^b$
[17, Theorem 1]	block encoding of A, B	$(\alpha\kappa\ B\ \log(\kappa\ B\ /\epsilon), \epsilon)$ Block encoding of X^c	LCU, real-time Hamiltonian simulation	$\mathcal{O}\left(\kappa \text{polylog}\left(\frac{\kappa\ B\ }{\epsilon}\right)\right)^b$
This work	Block encoding of A , state preparation for B	Mixed state $\mathbb{E}[\rho]$ with $\ \mathbb{E}[\rho] - \frac{X}{\text{tr}[X]}\ _1 \leq \epsilon$	Conditional restarts, imaginary-time Hamiltonian simulation	$\mathcal{O}\left(\frac{\kappa^3}{\epsilon} \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{\kappa}{\epsilon}\right)\right)$

^aIncludes amplitude amplification. Could be improved by replacing HHL with recent quantum linear system solvers.

^bSuccess probability p not included. Boosting success probability to 1 would lead to an additional factor $\mathcal{O}(1/p)$ from amplitude amplification.

^cFor some real constant $\alpha \neq 0$.

are highly heterogeneous with respect to the assumptions on the problem. To make the comparison possible we focus on the problem instance $AX + XA^\dagger + B = 0$ where $A, B \in \mathbb{C}^{N \times N}$, A is negative definite with $\|A\| = 1/2$, and B is positive semidefinite with $\text{tr}[B] = 1$. Furthermore, the algorithms use different query access models and produce output in different formats. Thus their runtime or query complexity comparison is for informational purposes and should not be interpreted as conclusive. Table II summarizes the algorithms in terms of input, output, primitives and runtime or query complexity, where the $\tilde{O}$ notation suppresses polylogarithmic factors and κ denotes the condition number of A . We did not include [16, Lemma 5], which can solve the Riccati equation $AX + X^\dagger A^\dagger + B = X^\dagger D X$ when $D, B \succ 0$ and $D^{-1}A$ is Hermitian. To solve our problem instance we would need $D = 0$, which is not allowed.

1. Sun and Zhang [11]

The algorithm in Ref. [11] prepares a pure quantum state proportional to the solution of the vectorized Lyapunov equation $(A \otimes I + I \otimes A) \text{vec}(X) = \text{vec}(-B)$, where I is the $N \times N$ identity and $\text{vec}(B)$ stacks the columns of B . Their input consists of a quantum circuit that prepares the state $|\text{vec}(-B)\rangle = \text{vec}(B)/\|\text{vec}(B)\|_2$ and an oracle for $(A \otimes I + I \otimes A)$. Assuming A is also s -sparse, the latter oracle can be implemented with a constant number of queries to a sparse oracle for A . Their work then applies the

HHL algorithm, which is a quantum linear system solver, to approximate the normalized state $|\text{vec}(X)\rangle$. This state is a normalized solution to the equation up to error ϵ in ℓ_2 distance. The runtime of HHL is $\tilde{O}(\log(N)s^2\kappa^2/\epsilon)$, which, however, could be improved by using modern quantum linear system solvers.

2. Clayton *et al.* [15]

The algorithm in Ref. [15] constructs a block encoding of the solution X . It begins by discretizing the integral solution to the continuous-time Lyapunov equation into a sum. Recall from our main text that the k th term of the sum is of the form $e^{k\Delta A} B e^{k\Delta A^\dagger}$. In Ref. [15] a block-encoding oracle for B is provided by assumption, while block encodings for $e^{k\Delta A}$ and $e^{k\Delta A^\dagger}$ shall be produced by imaginary-time Hamiltonian simulation methods. It is worthwhile to note that the product of block encodings is not the block encoding of the products in general, $U_{e^{k\Delta A}} U_B U_{e^{k\Delta A^\dagger}} \neq U_{e^{k\Delta A} B e^{k\Delta A^\dagger}}$, so achieving this will require additional ancilla registers [24, Lemma 53]. Clayton *et al.* then take a linear combination of unitaries (LCUs) with all the terms in the sum. The final result is obtained upon postselecting all the ancilla registers on the $|0\rangle$ state, which introduces a probability success. We do not include this factor in the complexity in Table II. Assuming the best possible normalization factor for the block encodings of B and A , i.e., their operator norm, the algorithm

outputs an approximation to $X/\|B\|$. For the query complexity of the algorithm we refer to the proof in Ref. [15, Theorem 4], which gives $\tilde{O}(\sqrt{\|B\|\kappa^5/\epsilon})$.

3. Somma *et al.* [17]

The algorithm in Ref. [17] constructs a block encoding of the solution X using an LCU and real-time Hamiltonian simulation as the main primitives. Their algorithm solves the more general problem $A'X + XB' = C'$. To specialize to our problem instance we set $A' = -A$, $B' = -A^\dagger = -A$, $C' = B$. Inputs of the algorithm are block encodings of the matrices A and B . The output is a quantum circuit $U_{X/x}$ for a block encoding of the normalized solution matrix X/x with $x \geq \kappa\|B\|$, $Q = -A \otimes I - I \otimes A$. The vectorized form of the Lyapunov equation reads $\text{vec}(X) = Q^{-1} \text{vec}(B)$. Now Somma *et al.* consider an integral representation of Q^{-1} and express X in terms of this integral after reverting the vectorization of the Lyapunov equation. Truncating and discretizing the integrals results in an LCU applied to the block encoding of B . The unitaries in the LCU are Hamiltonian simulations of the form $e^{-i\omega t A}$ with real ω . Reference [17] considers an implementation of the Hamiltonian simulation via QSVT to error $\|\Pi U_{X/x} \Pi - X/x\| \leq \epsilon$. Note that $Q \succ 0$ and we assume access to a block encoding of A for comparable input models. Hence, we use Theorem 1, Case 2 [17] to achieve query complexity $\mathcal{O}[\kappa \text{polylog}(\kappa\|B\|/\epsilon)]$ for $x \propto \kappa\|B\| \log(\kappa\|B\|/\epsilon)$. To use the block encoding $U_{X/x}$ in a concrete task, such as preparing a state $\propto X|\psi\rangle$ or computing matrix elements $\langle\phi|X|\psi\rangle$, requires additional queries to the oracle owing to the success probability $p < 1$ of preparing the block encoding. This leads to an additional factor of either $\mathcal{O}(1/\sqrt{p})$ if amplitude amplification is available or $\mathcal{O}(1/p)$ if the state preparation is repeated upon the wrong postselection outcome. We do not include this factor in the complexity in Table II. Note that their Theorem 1, Case 4 [17] for $Q \succ 0$ requires access to a block encoding of $\sqrt{A}$ but gives improved query complexity. This assumption would also improve the query complexity of our algorithm but we do not consider it here.

4. Our work

Consider our algorithm for the continuous-time Lyapunov equation. Recall that we have three additive error terms in our algorithm: discretization of integral ϵ_1 , finite limit of integration ϵ_2 , and approximate implementation error $\epsilon_\sim$. Suppose we want a total error ϵ , so we set each individual term to $\epsilon/3$. This requires a block encoding of $M = e^{\Delta A}$. As discussed in Appendix C 4, given access to a block encoding of $-A$ we can construct a suitable approximate block encoding with $\mathcal{O}[\kappa \log(T/\epsilon)]$ queries at each step. From Theorem 2 our algorithm uses parameters $T = (3\kappa^2/2\epsilon) \log(3/\epsilon)$ and $\Delta = \epsilon/3\kappa$. By Corollary 1 our algorithm is expected to run for $T + 1$ steps. Thus we

expect to require at most

$$\mathcal{O}\left[T\kappa \log\left(\frac{T}{\epsilon}\right)\right] = \mathcal{O}\left[\frac{\kappa^3}{\epsilon} \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{\kappa}{\epsilon}\right)\right] \quad (\text{F1})$$

queries to the block encoding of $-A$. Here we dropped $\log \log(1/\epsilon)$ terms.

-
- [1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems* (SIAM, Philadelphia, PA, 2005).
 - [2] Z. Gajic and M. T. J. Qureshi, *Lyapunov Matrix Equation in System Stability and Control* (Dover Publications, Mineola, NY, 2008).
 - [3] S. Bravyi and R. König, Classical simulation of dissipative fermionic linear optics, *Quantum Inf. Comput.* **12**, 925 (2012).
 - [4] A. Purkayastha, Lyapunov equation in open quantum systems and non-Hermitian physics, *Phys. Rev. A* **105**, 062204 (2022).
 - [5] P. Del Moral and A. Niclas, A Taylor expansion of the square root matrix functional, *J. Math. Anal. Appl.* **465**, 259 (2018).
 - [6] M. G. A. Paris, Quantum estimation for quantum technology, *Int. J. Quantum Inf.* **7**, 125 (2009).
 - [7] G. Wójtcowicz, S. F. Huelga, M. M. Rams, and M. B. Plenio, Quantum Fisher information from tensor-network integration of the Lyapunov equation, *Phys. Rev. A* **112**, 052454 (2025).
 - [8] R. H. Bartels and G. W. Stewart, Algorithm 432 [C2]: Solution of the matrix equation $AX + XB = C$ [F4], *Commun. ACM* **15**, 820 (1972).
 - [9] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Studies in Applied Mathematics Vol. 15 (SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994).
 - [10] V. Simoncini, Computational methods for linear matrix equations, *SIAM Rev.* **58**, 377 (2016).
 - [11] H. Sun and J. Zhang, Solving Lyapunov equation by quantum algorithm, *Control Theory Technol.* **15**, 267 (2017).
 - [12] Pedro C. S. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, Optimal scaling quantum linear-systems solver via discrete adiabatic theorem, *PRX Quantum* **3**, 040303 (2022).
 - [13] A. M. Dalzell, A shortcut to an optimal quantum linear system solver, *ArXiv:2406.12086*.
 - [14] M. E. S. Morales, L. Pira, P. Schleich, K. Koor, P. C. S. Costa, D. An, A. Aspuru-Guzik, L. Lin, P. Rebentrost, and D. W. Berry, Quantum linear system solvers: A survey of algorithms and applications, *ArXiv:2411.02522*.
 - [15] C. Clayton, J. Leng, G. Yang, Y.-L. Qiao, M. C. Lin, and X. Wu, Differentiable quantum computing for large-scale linear control, *ArXiv:2411.01391*.
 - [16] N. Liu, Q. Wang, M. M. Wilde, and Z. Zhang, Quantum algorithms for matrix geometric means, *npj Quantum Inf.* **11**, 101 (2025).

- [17] R. D. Somma, G. H. Low, D. W. Berry, and R. Babbush, Quantum algorithm for linear matrix equations, [ArXiv:2508.02822](#).
- [18] D. Zhang, J. L. Bosse, and T. Cubitt, Dissipative quantum Gibbs sampling, [ArXiv:2304.04526](#).
- [19] P. C. S. Costa, D. An, R. Babbush, and D. Berry, The discrete adiabatic quantum linear system solver has lower constant factors than the randomized adiabatic solver, [Quantum](#) **9**, 1887 (2025).
- [20] D. Orsucci and V. Dunjko, On solving classes of positive-definite quantum linear systems with quadratically improved runtime in the condition number, [Quantum](#) **5**, 573 (2021).
- [21] H. Mori, Y. Kikuchi, M. Benedetti, and M. Rosenkranz, Sparsity-dependent complexity lower bound of quantum linear system solvers, [ArXiv:2601.16697](#).
- [22] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, [Phys. Rev. Lett.](#) **103**, 150502 (2009).
- [23] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, [Quantum](#) **3**, 163 (2019).
- [24] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics, [ArXiv:1806.01838](#).
- [25] Here we use that $1/\ln(1/\|A'\|) = 1/\ln(1/\sqrt{1-1/\kappa}) < 2\kappa$ for $\kappa > 1$.
- [26] A. Sinap and W. Van Assche, Polynomial interpolation and Gaussian quadrature for matrix-valued functions, [Linear Algebra Appl.](#) **207**, 71 (1994).
- [27] R. A. Horn and C. R. Johnson, *Matrix Analysis* (Cambridge University Press, New York City, NY, 2017), 2nd ed.