

Function extraction via Chebyshev interpolation of quantum integral estimates

Gumaro Rendón 

Fujitsu Research of America, Inc., 4655 Great America Pkwy, Santa Clara, California 95054, USA



(Received 11 August 2025; accepted 23 March 2026; published 30 April 2026)

We present an algorithm for extracting a smooth, positive-definite function $\psi(x)$ encoded in a quantum state vector of size 2^n , while avoiding exponentially suppressed subnormalization of a naive extraction step. The method addresses a known limitation in quantum algorithms for differential equations, namely, the efficient retrieval of classical information from quantum-encoded solutions. It applies to settings where the target solution is smooth and positive definite, such as certain instances of the heat equation and related diffusive models arising in fluid dynamics and finance.

DOI: [10.1103/r6xs-236q](https://doi.org/10.1103/r6xs-236q)

I. INTRODUCTION

There is a long-standing problem, an often neglected problem, in quantum computing algorithms for linear systems and differential equations: solution extraction. It all goes back to that famous work [1], in which authors achieve exponential improvements with respect to problem size for the matrix inversion of a linear system. The assumption made in regard to the solution vector was that we did not need to know the solution itself, but features of it that can be extracted efficiently on a quantum computer. They argue that accessing each element to know the full solution spoils the exponential advantage.

Similar assumptions have been made since this seminal work for quantum linear systems or differential equations (e.g., Refs. [2–5]). This is done because explicit solution extraction from quantum state vector may spoil the exponential advantage with respect to system size. We propose here an antidote for this problem. But first, we will briefly explain with a one-dimensional abstraction why this problem exists. Suppose that we have an unknown smooth function $\psi(x)$ that is normalized:

$$\int_{-1}^1 |\psi(x)|^2 dx = 1. \quad (1)$$

We assume that this function can be stored in a quantum state vector with n qubits in the following way:

$$\langle 0|_a U_\psi |0\rangle|0\rangle_a = a_\psi |\psi\rangle, \quad (2)$$

where

$$|\psi\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=0}^{2^n-1} \psi(x_j) |j\rangle, \quad (3)$$

$\mathcal{N} = \sum_x |\psi(x)|^2 \approx 2^{n-1}$, and $x_j = (2j/2^n - 1)$. Here, the ancillary register $|0\rangle_a$ is part of the algorithm to prepare the solution on the main register, and a_ψ is the subnormalization expected if the algorithm to prepare the target state is nonunitary.

This is in many cases how we expect the final vector solution to be encoded in algorithms for quantum linear system problems (see Refs. [1,2,6,7]) or other means of solving differential equations on a quantum computer [3,8]. However, these algorithms usually stop short of extracting the solution information or assume that the required quantities, e.g., $\langle \psi | O | \psi \rangle$, are easy to extract. Here, we address the problem that is the exponential subnormalization that happens when trying to extract the solution from a single site:

$$\langle \psi | j \rangle \langle j | \psi \rangle \propto |\psi(x_j)|^2 / 2^n. \quad (4)$$

There is a modest quadratic improvement by using quantum amplitude estimation (QAE [9]) where the cost would go like $O(|\psi(x_j)|/\sqrt{2^n})$. Moreover, the existing methods of tomography like classical shadows [10] do not help in this case because these would estimate each $\langle \psi | j \rangle \langle j | \psi \rangle$ with additive error ϵ at a cost that is of the order $O(1/\epsilon^2)$, but the subnormalization $1/2^n$ remains; this would mean that we would need to adjust the target ϵ accordingly to compensate for this exponential subnormalization.

There is some work where authors use polynomial expansions and tensor networks to extract the encoded information in a quantum state of size 2^n (see Ref. [11]); however, this method relies on variational optimization, which lacks guarantees of convergence because the surface is unknown. There have also been some developments in the use of orthogonal polynomial inspired variational quantum algorithms to extract information from quantum state vectors [12] whose time complexity ultimately scales linearly with n ; however, as we will see, we manage to decouple the number of encoding qubits with the total time complexity at least in the large error regimes. Another method [13] relying on estimating the overlap of the solution with discrete representations of an orthogonal polynomial basis achieves a similar result to the

*Contact author: grendon@fujitsu.com

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

one presented here, but here we lack the overhead from the circuits required to prepare the orthogonal polynomials and instead we rely on interpolation.

With this work, we aim to offer an alternative with different strengths, which also ameliorates this problem of solution information extraction from memory for a subset of problems. We assume that $\psi(x) > 0$ and that $\psi(x)$ is analytic on the interval of interest with derivatives bounded by $|\frac{d^k \psi(x)}{dx^k}| \leq (\Lambda)^{k+1}$, where $\Lambda_k = \max_{x,j \in [k]} (\psi^{(j)}(x))^{1/(j+1)}$ with $[k] := \{j \in \mathbb{N} | j \leq k\}$ and $\Lambda = \Lambda_\infty$. Such solutions can be encountered in the initial value problem of the heat equation regardless of the smoothness of the initial value [14], or other parabolic equations with a diffusion term [15,16].

The main findings of our work here are summarized in the following theorem:

Theorem 1. Provided an unknown analytic function $\psi(x)$ for $x \in [-1, 1]$ that is normalized:

$$\int_{-1}^1 |\psi(x)|^2 dx = 1,$$

and whose derivatives are bounded by $|\frac{d^j \psi(x)}{dx^j}| \leq \Lambda^{j+1}$, which can be stored in a quantum state vector with n qubits in the following way:

$$\langle 0|_a U_\psi |0\rangle|0\rangle_a = a_\psi |\psi\rangle,$$

where

$$|\psi\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=0}^{2^n-1} \psi(x_j) |j\rangle,$$

$\mathcal{N} = \sum_j |\psi(x_j)|^2$, and $x_j = 2j/2^n - 1$, one can estimate it at a quantum gate cost that asymptotically behaves with the following scaling:

$$\tilde{O}\left(\frac{1}{a_\psi} \frac{\Lambda^4}{\epsilon_{\text{total}}} \frac{\max_x \psi(x)}{\min_x \psi(x)}\right),$$

and the time complexity is

$$\tilde{O}\left(\frac{1}{a_\psi} \frac{\Lambda^3}{\epsilon_{\text{total}}} \frac{\max_x \psi(x)}{\min_x \psi(x)}\right).$$

II. EXTRACTING THE AMPLITUDES FROM THE QUANTUM STATE VECTOR

The method can be split into four important steps:

- (1) Sample the squared integral of the solution.
- (2) Interpolate those points.
- (3) Differentiate that interpolant to obtain the integrand.
- (4) Finally, take the square root of that approximation to obtain back the solution.

In what follows, we will describe what each of those steps consists of.

A. Squared amplitude integral estimation with binary segmentation

Here, we summarize the results from squared amplitude estimation, with which we avoid the scaling factor $1/2^n$ because it gets absorbed into the very definition of the integral (the differential factor).

Lemma 1 (Probability integral estimation). Given an analytic function $\psi(x)$, and state vector with positive-definite amplitudes $|\psi\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=0}^{2^n-1} \psi(x_j) |j\rangle$, and a unitary such that $\tilde{U}_\psi |0\rangle = |\psi\rangle$, where $x_j = 2j/2^n - 1$, we can estimate the integral

$$\Psi(\hat{x}) = \int_{-1}^{\hat{x}} \psi^2(x') dx',$$

where $\hat{x} = 2X/2^n - 1$, and $X \in \{0, 1, 2, \dots, 2^n - 1\}$, within an error ϵ_ψ with a time complexity of

$$\tilde{O}\left(\frac{\ln \frac{1}{\epsilon_\psi} \max_x \psi(x)}{\epsilon_\psi}\right),$$

and a query complexity that scales like

$$O\left(\frac{\ln^2 \frac{1}{\epsilon_\psi} \max_x \psi(x)}{\epsilon_\psi}\right), \quad (5)$$

provided that $\epsilon_\psi \leq n \max_x \psi^2(x)$.

Remark. An alternative approach using existing comparator/indicator circuits would bring the overall query complexity to

$$O\left(\frac{n^2 \max_x \psi(x)}{\epsilon_\psi}\right), \quad (6)$$

and the same for the time complexity. This is obtained using the circuit in Ref. [17]. For a more comprehensive review on comparator circuits, see Ref. [18].

The n^2 scaling comes from the Quantum Fourier Transform (QFT) required for the addition and subtraction operations in the comparator circuit according to the version in Ref. [17].

Thus, the method for integral estimation presented here has a better time complexity than the one using existing comparator circuits by avoiding the use of QFT-based arithmetic.

For a full proof of Lemma 1, see Appendix C. If instead of the unitary $\tilde{U}_\psi$, we have U_ψ from Eq. (2), this follows:

Corollary 1 (Nonunitary probability integral estimation). Given an analytic function $\psi(x)$, and state vector with positive-definite amplitudes $|\psi\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{j=0}^{2^n-1} \psi(x_j) |j\rangle$, and a unitary such that $\langle 0|_a U_\psi |0\rangle|0\rangle_a = a_\psi |\psi\rangle$, where $x_j = 2j/2^n - 1$, we can estimate the integral $\Psi(\hat{x}) = \int_{-1}^{\hat{x}} \psi^2(x') dx'$ where $\hat{x} = 2X/2^n - 1$, and $X \in \{0, 1, 2, \dots, 2^n - 1\}$, within an error ϵ with a time complexity of

$$\tilde{O}\left(\frac{1}{a_\psi} \frac{\ln \frac{1}{\epsilon_\psi} \max_x \psi(x)}{\epsilon_\psi}\right),$$

and a query complexity that scales like

$$O\left(\frac{1}{a_\psi} \frac{\ln^2 \frac{1}{\epsilon_\psi} \max_x \psi(x)}{\epsilon_\psi}\right), \quad (7)$$

provided that $\epsilon_\psi \leq n \max_x \psi^2(x)$.

With these results, we are certain of being able to estimate the integrals $\Psi(x)$ without having to run QAE for $\sim 2^n$ points separately. However, if we want to have the functional form of $\Psi(x)$ throughout the whole interval, we must perform some kind of interpolation.

B. Estimating the integral Cebyshev expansion with a few evaluations

Here, we explain how we obtain the integral of the squared amplitude, $\Psi(x)$, for the whole interval with just a few samples. Recalling that

$$\Psi(x) = \int_{-1}^x \psi(x')^2 dx', \quad (8)$$

we can now use Chebyshev interpolation to obtain the expansion coefficients of the interpolant through the following linear system: $\mathbf{V}a = f$, where $(f)_k = \Psi(x_k)$, and x_k are a choice of interpolation nodes. We have that

$$\mathbf{V} = \begin{pmatrix} u_0(x_{\text{cheb},1}) & u_1(x_{\text{cheb},1}) & \dots & u_{M-1}(x_{\text{cheb},1}) \\ u_0(x_{\text{cheb},2}) & u_1(x_{\text{cheb},2}) & \dots & u_{M-1}(x_{\text{cheb},2}) \\ \vdots & \vdots & \ddots & \vdots \\ u_0(x_{\text{cheb},M}) & u_1(x_{\text{cheb},M}) & \dots & u_{M-1}(x_{\text{cheb},M}) \end{pmatrix}. \quad (9)$$

With this, one can obtain the coefficients, a_j , with $a = \mathbf{V}^{-1}f$. Like in Refs. [19,20], the choice of interpolating nodes $x_{\text{cheb},k}$ and the interpolating set of polynomials are the Chebyshev nodes and polynomials. In that case, u_j is defined by

$$u_j(x) := \begin{cases} \sqrt{\frac{1}{M}} T_0(x), & j = 0, \\ \sqrt{\frac{2}{M}} T_j(x), & j = 1, 2, \dots, M-1, \end{cases} \quad (10)$$

where T_j is the standard j th Chebyshev polynomial, $T_j(x) := \cos(j \cos^{-1} x)$. The node collocation is described by

$$x_{\text{cheb},k} = \cos\left(\frac{2k-1}{2M}\pi\right), \quad k \in \{1, 2, \dots, M\}. \quad (11)$$

These polynomials fulfill the discrete orthonormality condition [21] with respect to the collocation nodes, that is,

$$\sum_{k=1}^M u_i(x_{\text{cheb},k}) u_j(x_{\text{cheb},k}) = \delta_{ij} \quad (12)$$

for all $0 \leq i, j < M$. With this, we know that the condition number for $\mathbf{V}$ is optimal, or $\kappa(\mathbf{V}) = \sigma_{\max}(\mathbf{V})/\sigma_{\min}(\mathbf{V}) = 1$. The number of nodes needed for a target error $\epsilon_{\text{cheb}} = |\Psi(x) - P_{M-1}\Psi(x)|$ is estimated and given in the following lemma:

Lemma 2 (Chebyshev integral interpolation). Given the following integral

$$\Psi(x) = \int_{-1}^x \psi(x')^2 dx',$$

where $|\frac{d^k \psi(x)}{dx^k}| \leq (\Lambda)^{k+1}$ for $x \in [0, 1]$, we can interpolate $\Psi(x)$ through

$$P_{M-1}\Psi(x) = \sum_{j=0}^{M-1} a_j u_j(x),$$

with a target error ϵ_{cheb} , where

$$u_j(x) := \begin{cases} \sqrt{\frac{1}{M}} T_0(x), & j = 0, \\ \sqrt{\frac{2}{M}} T_j(x), & j = 1, 2, \dots, M-1, \end{cases}$$

$$T_j(x) := \cos(j \cos^{-1} x),$$

using

$$M = O(\Lambda + \ln(1/\epsilon_{\text{cheb}}))$$

samples of Ψ , $(f)_k = \Psi(x_{\text{cheb},k})$, with $x_{\text{cheb},k} = \cos(\frac{2k-1}{2M}\pi)$. Here, a_j are obtained by solving the linear system:

$$\mathbf{V}a = f,$$

where

$$\mathbf{V} = \begin{pmatrix} u_0(x_{\text{cheb},1}) & u_1(x_{\text{cheb},1}) & \dots & u_{M-1}(x_{\text{cheb},1}) \\ u_0(x_{\text{cheb},2}) & u_1(x_{\text{cheb},2}) & \dots & u_{M-1}(x_{\text{cheb},2}) \\ \vdots & \vdots & \ddots & \vdots \\ u_0(x_{\text{cheb},M}) & u_1(x_{\text{cheb},M}) & \dots & u_{M-1}(x_{\text{cheb},M}) \end{pmatrix}.$$

For a full derivation of these bounds, see Appendix B. With these results, we have elucidated the path to obtain an interpolant of the integral $\Psi(x) = \int_{-1}^x \psi^2(x') dx'$; now, to obtain the integrand, one must just differentiate $\Psi(x)$ with respect to x . In what follows, we show the results for uncertainty propagation after differentiation and due to perturbed node collocation (we cannot exactly sample the integral at Chebyshev nodes).

C. Extracting the integrand ψ^2

Here, we explain how we obtain an approximation of the integrand ψ^2 out of the interpolant for $\Psi(x)$ [see Eq. (8)]. We also list the lemma that tells us how the errors, ϵ_ψ , from the estimated $\Psi(x)$ propagate toward this approximant of ψ^2 with error ϵ_{ψ^2} (for the full proof of Lemma 3, see Appendix D).

First, note that we are actually going to obtain the coefficients from

$$\mathbf{V}_{\text{pert}} a' = f', \quad (13)$$

where f' is the integral sample with errors introduced from the finite-resources amplitude estimation and from using perturbed node collocations, $x_{\text{mock},k}$. $\mathbf{V}_{\text{pert}}$ is the corresponding slightly perturbed Chebyshev-Vandermonde matrix:

$$\mathbf{V}_{\text{pert}} = \begin{pmatrix} u_0(x_{\text{mock},1}) & u_1(x_{\text{mock},1}) & \dots & u_{M-1}(x_{\text{mock},1}) \\ u_0(x_{\text{mock},2}) & u_1(x_{\text{mock},2}) & \dots & u_{M-1}(x_{\text{mock},2}) \\ \vdots & \vdots & \ddots & \vdots \\ u_0(x_{\text{mock},M}) & u_1(x_{\text{mock},M}) & \dots & u_{M-1}(x_{\text{mock},M}) \end{pmatrix}, \quad (14)$$

where the perturbation comes from obtaining the amplitudes at slightly off collocation nodes, $x_{\text{mock},k}$. This comes from using a finite-size grid, even though exponential in n .

The error propagation from the integral estimation toward the integrand estimation can be summarized in the following lemma:

Lemma 3 (Error propagation to derivative of interpolant). Provided we estimate $\Psi(x)$ through $\tilde{\Psi}(x)$ at the perturbed nodes $x_{\text{mock},k}$ closest to the Chebyshev nodes

$$x_{\text{cheb},k} = \cos\left(\frac{2k-1}{2M}\pi\right), \quad k \in \{1, 2, \dots, M\},$$

where $x_{\text{mock},k} \in \{2(j/2^n) - 1 \mid j \in \mathbb{N} \mid j < 2^n\}$ with error $|\tilde{\Psi}(x_{\text{mock},j}) - \Psi(x_{\text{mock},j})| \leq \epsilon_\psi$, and the truncation error from

Chebyshev interpolation is $|\Psi(x) - P_{M-1}\Psi(x)| \leq \epsilon_{\text{cheb}} \sim \epsilon_\Psi$, we can obtain an approximation of its derivative with

$$\tilde{\psi}(x) = \sum_{j=0}^{M-1} j a'_j U_j(x)$$

within an error

$$O(M^3 \epsilon_\Psi), \quad (15)$$

where a' are the coefficients that solve the following perturbed Chebyshev-Vandermonde linear system:

$$\mathbf{V}_{\text{pert}} a' = f',$$

where $f_j = \tilde{\Psi}(x_{\text{mock},j})$, and

$$\mathbf{V}_{\text{pert}} = \begin{pmatrix} u_0(x_{\text{mock},1}) & u_1(x_{\text{mock},1}) & \dots & u_{M-1}(x_{\text{mock},1}) \\ u_0(x_{\text{mock},2}) & u_1(x_{\text{mock},2}) & \dots & u_{M-1}(x_{\text{mock},2}) \\ \vdots & \vdots & \ddots & \vdots \\ u_0(x_{\text{mock},M}) & u_1(x_{\text{mock},M}) & \dots & u_{M-1}(x_{\text{mock},M}) \end{pmatrix}$$

and

$$u_j(x) := \begin{cases} \sqrt{\frac{1}{M}} T_0(x), & j = 0, \\ \sqrt{\frac{2}{M}} T_j(x), & j = 1, 2, \dots, M-1. \end{cases}$$

Also, the aforementioned error on the estimate of $\psi(x)$ is achieved provided the perturbation on the nodes $|x_{\text{cheb},j} - x_{\text{mock},j}| = O(1/2^n)$ is small enough such that $\|\mathbf{V}_{\text{pert}}^{-1}\| = O(1)$, and $|\Psi(x_{\text{cheb},j}) - \Psi(x_{\text{mock},j})| = O(\Lambda^2/2^n) \sim \epsilon_\Psi$.

With these results, we have with us access to the integrand $\psi^2(x)$ and estimates on the resources required to achieve a target error on $\psi^2(x)$. That is, for some target error ϵ_{ψ^2} , we need to sample the integral $\Psi(x)$ with a tolerance $O(\epsilon_{\psi^2}/M^3)$, where M is expected to be $O(\ln 1/\epsilon_{\text{cheb}})$. Finally, to approximate $\psi(x)$, we need to take the square root of the approximant of $\psi^2(x)$.

D. Taking the square root of the approximant of ψ^2

Here, we briefly explain how we are able to take the square root of the approximant of ψ^2 without introducing any discontinuities and large uncertainties. For that, we will make use of the following lemma:

Lemma 4. Given a positive-definite function $f(x)$ with finite first derivative for $x \in [-1, 1]$, which we have estimated through $\tilde{f}(x)$ within error ε , we can estimate $\sqrt{f(x)}$ through

$$\sqrt{\tilde{f}(x)}, \quad (16)$$

which would have an error with respect to $\sqrt{f(x)}$:

$$|\sqrt{f(x)} - \sqrt{\tilde{f}(x)}| = O\left(\frac{\varepsilon}{\min_s(\sqrt{f(x)})}\right),$$

provided that $\varepsilon \leq \min_s f(x)$.

Proof. We start by recalling the following bound for an analytic function $g(x)$:

$$|g(x \pm \varepsilon) - g(x)| \leq \varepsilon \max_{y \in [x-\varepsilon, x+\varepsilon]} (g'(y)).$$

In order to use this identity, $\tilde{f}(x)$ must also be positive definite; thus, we assume that $\varepsilon \leq \min_s f(x)$ such that the estimate

$\tilde{f}(x)$ remains positive definite. Now, we look at the first derivative of $\sqrt{f(x)}$: $\frac{d\sqrt{f(x)}}{dx} = \frac{1}{2\sqrt{f}}$. With this, we know the error propagated to $\sqrt{\tilde{f}(x)}$ is

$$|\sqrt{f(x)} - \sqrt{\tilde{f}(x)}| = O\left(\frac{\varepsilon}{\min_s(\sqrt{f(x)})}\right) \quad (17)$$

for some $\varepsilon < \min_s f(x)$. ■

Finally, our results can be summarized in the main theorem (Theorem 1). The joint use of our previous results to prove this main result goes as follows:

Proof. Using Lemma 4, we find that the error propagated from taking the square root of the approximation of ψ^2 , $2 \sum_j j a'_j U_j(x)$, is $\epsilon_\psi = O(\frac{\epsilon_{\psi^2}}{\min_x \psi(x)})$ provided that $\epsilon_{\psi^2} \leq \min_s \psi(x)$. Now, using the results from Lemma 3, we know that at the same time the error from our estimates of $\Psi(x_{\text{mock},k})$ propagates to $\psi(x)$ in the following way: $\epsilon_\psi = O(\frac{\epsilon_{\psi^2}}{\min_x \psi(x)}) = \tilde{O}(\frac{M^3 \epsilon_\Psi}{\min_x \psi(x)})$.

Putting all of these results together with the quantum cost estimates from Lemma 1 and Corollary 1, we obtain that we can extract $\psi(x)$ from memory (a unitary U_ψ , more specifically) through the following cost: $\tilde{O}(\frac{1}{a_\psi} \frac{M^3}{\epsilon_{\text{total}}} \frac{\max_x \psi(x)}{\min_x \psi(x)})$.

Finally, we have that $M = O(\Lambda + \ln(1/\epsilon_{\text{cheb}}))$ from Lemma 4. Now, if we choose $\epsilon_{\text{cheb}} \sim \epsilon_{\text{total}}$, we have $\tilde{O}(\frac{1}{a_\psi} \frac{\Lambda^3}{\epsilon_{\text{total}}} \frac{\max_x \psi(x)}{\min_x \psi(x)})$. With this, we have proved Theorem 1. ■

III. CONCLUSION

We have provided here a method to extract solutions from quantum state vector with an exponential improvement over naive quantum amplitude estimation. The constraints are that $\psi(x)$ must be positive definite and analytic, and $\min_x \psi(x)$ is not very small. This type of solutions one straightforwardly encounters in diffusive differential equations like those encountered in option pricing and the heat equation.

In the future, we would like to extend this analysis to problems where there are zero crossings present, shock or other singularities, and in more general setups.

ACKNOWLEDGMENTS

I would like to thank Dr. Sarvagya Upadhyay of Fujitsu Research of America, Inc., for his valuable feedback on the manuscript. I am also deeply grateful to Dr. Hirotaka Oshima, Dr. Quoc Hoan Tran, and Dr. Yasuhiro Endo of Fujitsu Research of Japan, Inc., for their insightful comments and suggestions throughout the course of the project. I would like to thank Štěpán Šmíd for his useful comments on the final draft and directing me toward efficient comparator circuits.

DATA AVAILABILITY

No data were created or analyzed in this study.

APPENDIX A: PERTURBATION ON LINEAR SYSTEM MATRIX, V

Regarding the perturbed matrix $\mathbf{V}_{\text{pert}}$, we assumed that the norm of its inverse was $\|\mathbf{V}_{\text{pert}}^{-1}\|_2 = O(1)$. The original

Chebyshev-Vandermonde, $\mathbf{V}_{\text{cheb}}$, matrix has all eigenvalues $\lambda_j = 1$, hence the stability. However, we are limited to collocating the sampling nodes at equispaced distance. To get an idea on why we can assume $\|\mathbf{V}_{\text{pert}}^{-1}\|_2 = O(1)$, we first consider the upper bounds on eigenvalue perturbation [22]:

$$|\lambda_j - \lambda| \leq \|\mathbf{V} - \mathbf{V}_{\text{pert}}\|_2. \quad (\text{A1})$$

Now, in order to estimate the size of the spectral perturbation on the matrix, we look at the perturbation on each of its elements:

$$\begin{aligned} \|\mathbf{V} - \mathbf{V}_{\text{pert}}\|_{\max} &= \max_{k, j \leq M-1} |u_j(x_{\text{cheb},k}) - u_j(x_{\text{mock},k})| \\ &\leq |x_{\text{cheb},k} - x_{\text{mock},k}| \max_{x \in [-1,1]} \left| \frac{d}{dx} u_j(x) \right|. \end{aligned} \quad (\text{A2})$$

First, the error on the collocation points is

$$|x_{\text{cheb},k} - x_{\text{mock},k}| \leq 2/N. \quad (\text{A3})$$

To bound the derivative $|\frac{d}{dx} u_j(x)|$, we first note that the derivative of $T_j(x)$ is

$$\frac{dT_j(x)}{dx} = jU_{j-1}(x), \quad (\text{A4})$$

where $U_j(x)$ are the Chebyshev polynomials of second kind:

$$U_j(x) = \frac{\sin((j+1)\arccos(x))}{\sin(\arccos(x))}. \quad (\text{A5})$$

We also note that

$$\max_x |U_j(x)| \leq j+1, \quad (\text{A6})$$

which in turn means that

$$\max_x \left| \frac{dT_j(x)}{dx} \right| \leq j^2. \quad (\text{A7})$$

This means the maximum norm on the perturbation is

$$\begin{aligned} \|\mathbf{V} - \mathbf{V}_{\text{pert}}\|_{\max} &\leq \frac{2(M-1)^2}{N} \sqrt{\frac{2}{M}} \\ &\leq \frac{3M^{3/2}}{N}. \end{aligned} \quad (\text{A8})$$

Thus, the corresponding spectral norm is

$$\begin{aligned} \|\mathbf{V} - \mathbf{V}_{\text{pert}}\|_2 &\leq M \|\mathbf{V} - \mathbf{V}_{\text{pert}}\|_{\max} \\ &\leq \frac{3M^{5/2}}{N}. \end{aligned} \quad (\text{A9})$$

Bauer and Fike's lemma means that

$$\lambda_{\min}(\mathbf{V}_{\text{pert}}) \geq 1 - \|\mathbf{V} - \mathbf{V}_{\text{pert}}\|_2; \quad (\text{A10})$$

thus,

$$\begin{aligned} \|\mathbf{V}_{\text{pert}}\|_2 &= \lambda_{\min}^{-1}(\mathbf{V}_{\text{pert}}) \leq 1 + 2\|\mathbf{V} - \mathbf{V}_{\text{pert}}\|_2 \\ &\leq 1 + \frac{6M^{5/2}}{N}, \end{aligned} \quad (\text{A11})$$

provided that $\frac{6M^{5/2}}{N} \leq 1$. These findings are summarized in the following lemma:

Lemma A1. A Chebyshev-Vandermonde $M \times M$ matrix, constructed with M nodes s'_j , which are the closest points to the Chebyshev nodes s_j , and which can also be placed on an equispaced grid on $x \in [-1, 1]$ with N points, has

$$\|\mathbf{V}_{\text{pert}}^{-1}\|_2 \leq 1 + \frac{6M^{5/2}}{N},$$

provided that $\frac{6M^{5/2}}{N} \leq 1$.

We expect this deviation from 1 to be very small since $N = 2^n$ grows exponentially with the number of qubits and $M = O(\Lambda + \ln 1/\epsilon)$. For that reason, it is safe to assume that the perturbation is small and thus $\|\mathbf{V}_{\text{pert}}\|_2 = O(1)$.

APPENDIX B: PROOF OF CHEBYSHEV INTEGRAL INTERPOLATION LEMMA

Proof. With this we can now use the convergence rates for Chebyshev interpolation (see Lemma 1 of Ref. [20]):

$$\epsilon_{\text{cheb}} = |\Psi(x) - P_{M-1}\Psi(x)| \leq \frac{1}{2^{M-1}M!} \max_s \left| \frac{d^M \Psi(x)}{dx^M} \right|. \quad (\text{B1})$$

First, we note that

$$\frac{d\Psi(x)}{dx} = \psi^2. \quad (\text{B2})$$

Also, by Leibniz' general rule,

$$\frac{d^j \psi^2}{dx^j} = \sum \binom{j}{k} \psi^{(j-k)}(x) \psi^{(k)}(x). \quad (\text{B3})$$

Thus, a bound is

$$\left| \frac{d^j \psi^2}{dx^j} \right| \leq (\Lambda)^{j+1} \sum \binom{j}{k} = (\Lambda)^{j+1} 2^j = (2\Lambda)^{j+1}/2. \quad (\text{B4})$$

Thus,

$$\begin{aligned} \epsilon_{\text{cheb}} &= |\Psi(x) - P_{M-1}\Psi(x)| \leq \frac{1}{2^{M-1}M!} \max_s \left| \frac{d^{M-1} \psi^2(x)}{dx^{M-1}} \right| \\ &\leq \frac{1}{M!} (\Lambda)^M. \end{aligned} \quad (\text{B5})$$

Using Stirling bounds for $M!$, for $M \geq 1$, we find that

$$\begin{aligned} \epsilon_{\text{cheb}} &= |\Psi(x) - P_{M-1}\Psi(x)| \\ &\leq \frac{1}{\sqrt{2\pi M}(M/e)^M} (\Lambda)^M \\ &\leq \frac{1}{(M/e)^M} (\Lambda)^M \\ &= \exp(-M(\ln M - 1 - \ln \Lambda)). \end{aligned} \quad (\text{B6})$$

We can write this inequality in the following way:

$$\frac{M}{\Lambda e} \ln \left(\frac{M}{\Lambda e} \right) \leq \frac{\ln \frac{1}{\epsilon_{\text{cheb}}}}{\Lambda e}. \quad (\text{B7})$$

We note that this has the following solution (see Ref. [23]):

$$M \leq \Lambda e \exp \left(W_0 \left(\frac{\ln(1/\epsilon_{\text{cheb}})}{\Lambda e} \right) \right), \quad (\text{B8})$$

where W_0 stands for the Lambert- W function. We now note the following bound [24]:

$$W_0(x) \leq \ln \left(\frac{2x+1}{1+\ln(x+1)} \right). \quad (\text{B9})$$

For $x > 0$, we can use simpler looser bound:

$$W_0(x) \leq \ln(2x+1). \quad (\text{B10})$$

Using this last bound, we obtain

$$\begin{aligned} M &\leq \Lambda e \exp \left(\ln \left(\frac{2 \ln(1/\epsilon_{\text{cheb}})}{\Lambda e} + 1 \right) \right) \\ &= \Lambda e + 2 \ln(1/\epsilon_{\text{cheb}}). \end{aligned} \quad (\text{B11})$$

So, the number of nodes M corresponding to a target error ϵ_{cheb} is bounded through the inequality above. It means that to ensure we have an error lower than a target error of ϵ_{cheb} , we must choose an $M \geq \Lambda e + \ln(1/\epsilon_{\text{cheb}})$. So, to achieve an interpolation error lower than or equal to ϵ_{cheb} , one may choose an integer value for M such that

$$M = \Theta(\Lambda + \ln(1/\epsilon_{\text{cheb}})), \quad (\text{B12})$$

but fulfills $M \geq \Lambda e + \ln(1/\epsilon_{\text{cheb}})$.

APPENDIX C: PROOF OF LEMMA ON THE PROBABILITY INTEGRAL ESTIMATION

Proof. We can estimate the integral with the following left-Riemann sum:

$$\Psi(\hat{x}) = \frac{1}{2^{n-1}} \sum_{j=0}^{X-1} \psi^2(x_j) + O\left(\frac{\Lambda^2}{2^n}\right). \quad (\text{C1})$$

We can further estimate this sum with the renormalized one

$$\Psi(\hat{x}) = \frac{1}{\mathcal{N}} \sum_{j=0}^{X-1} \psi^2(x_j) + O\left(\frac{\Lambda^2}{2^n}\right) \quad (\text{C2})$$

because $\mathcal{N}/2^{n-1} = 1 + O(\Lambda^2/2^n)$. This last statement can be corroborated by thinking of $\mathcal{N}/2^n$ as the left-Riemann sum approximation of $\int_0^1 \psi^2(x') dx' = 1$ through

$$\mathcal{N}/2^{n-1} = \frac{1}{2^{n-1}} \sum_{j=0}^{2^n-1} \psi^2(x_j) = 1 + O(\Lambda^2/2^n).$$

Here, we now consider $\hat{x} = 2X/2^n - 1$, for which we can always find the following binary decomposition:

$$X = X_0 + 2X_1 + \dots + 2^{m-1}X_{m-1}, \quad (\text{C3})$$

where $X_j \in \{0, 1\}$ and we truncate this expansion so that $X_{m-1} = 1$. Thus, the procedure for estimating this sum of probabilities can be split into segments defined by this decomposition:

$$\sum_{p=0}^{m-1} q_p^2 = \frac{1}{\mathcal{N}} \sum_{j=0}^{X-1} \psi^2(x_j), \quad (\text{C4})$$

where

$$q_p = \sqrt{\frac{1}{\mathcal{N}} \sum_{k=W_p}^{W_p+2^p-1} \psi^2(x_k)} \quad (\text{C5})$$

and

$$W_p = \sum_{k=p-1}^{m-1} 2^k X_k. \quad (\text{C6})$$

We can make a further approximation by truncating the q_p^2 sum

$$\frac{1}{\mathcal{N}} \sum_{j=0}^{X-1} \psi^2(x_j) \approx \sum_{p=l}^{m-1} q_p^2. \quad (\text{C7})$$

The error from this truncation, labeled as ϵ_{coarse} , is going to be

$$\epsilon_{\text{coarse}} = O\left(\frac{2^l}{2^n}\right). \quad (\text{C8})$$

Thus, we only need to do $m-l$ routines that approximate each of the q_p to obtain an estimate of

$$\sum_{p=0}^{m-1} q_p^2 = \frac{1}{\mathcal{N}} \sum_{j=0}^{X-1} \psi^2(x_j). \quad (\text{C9})$$

As we will see, splitting the whole sum into these segments will make the sum estimation efficient. Supposing we have the unitary that prepares the state $\frac{1}{\sqrt{\mathcal{N}}} \sum_x \psi(x)|x\rangle$, we can approximate this sum on a quantum computer through the following routine:

In Algorithm 1, “flipping the bits in the positions in p_{list} ” means applying Pauli- X gates to those qubits. These flips implement a virtual shift of the previously selected binary block, so that the interval

$$\{W_p, \dots, W_p + 2^p - 1\}$$

is mapped to a standard subspace in which all bits from position p to the right are zero. Hence, after the flips, measuring the probability of the suffix $00 \dots 0$ from bit p to the right yields

$$q_p^2 = \frac{1}{\mathcal{N}} \sum_{k=W_p}^{W_p+2^p-1} \psi^2(x_k).$$

For example, if $X = 01011$ and we consider the step $p = m-2$, then the relevant block is the one from $X = 00001$ to $X = 00011$, and

$$q_p^2 = \frac{1}{\mathcal{N}} (\psi^2(x_{00001}) + \psi^2(x_{10001}) + \dots + \psi^2(x_{11101})).$$

In this case, the prescribed Pauli- X flips map basis states of the form $|***01\rangle$ to $|***00\rangle$, so the subsequent measurement of the all-zero suffix indeed returns the desired summed probability mass over that block.

The error on each estimate q_p^2 is bounded by

$$\epsilon_p^2 + 2q_p \epsilon_p. \quad (\text{C10})$$

ALGORITHM 1. Cumulative probability estimation.

Input: Binary decomposition $X = \sum_j X_j 2^j$,
samples of $\psi(x)$ stored in quantum
amplitudes

Output: Estimate of $\frac{1}{N} \sum_{j=0}^{X-1} \psi^2(x_j)$

$p_{\text{list}} \leftarrow \{\}$ // List of non-zero bits that
track cumulative virtual shift

for $p \in \{m-1, m-2, \dots, l\}$ **do**

if $X_p = 1$ **then**

 Estimate the quantity:

$$q_p = \sqrt{\frac{1}{N} \sum_{k=W_p}^{W_p+2^p-1} \psi^2(x_k)}$$

 with precision ε_p by measuring the
probability (or its square root for QAE)
of all zeros from bit p to the right
after flipping the bits (acting on them with
the Pauli-X gate) in the positions in p_{list} .

$p_{\text{list}} \leftarrow p_{\text{list}} + p$ // Append to index
list to track cumulative virtual
shift

end

end

Moreover, we have the following bound:

$$q_p^2 \leq \sum_p q_p^2 \sim \Psi(\hat{x}) \leq (1 + |\hat{x}|) \max_{x \leq \hat{x}} \psi^2(x). \quad (\text{C11})$$

With this, and the fact that $(1 + |\hat{x}|) \leq 2$, we know that $q_p \lesssim \sqrt{2} \max_{x \leq \hat{x}} \psi(x)$, which means the error on each estimate of q_p^2 is

$$O(\varepsilon_p^2 + \varepsilon_p \max_x \psi(x)). \quad (\text{C12})$$

From now on, we assume that all $\varepsilon_p = \varepsilon$. By repeated triangle inequality, we have that the total error on our estimate of $\sum_p q_p^2$ is

$$O((m-l)(\varepsilon^2 + \varepsilon \max_x \psi(x))). \quad (\text{C13})$$

Using the fact that $m \leq n$, and that $(n-l) = O(\ln 1/\epsilon_{\text{coarse}})$, we bound the error through the expression

$$O\left(\ln \frac{1}{\epsilon_{\text{coarse}}} (\varepsilon^2 + \varepsilon \max_x \psi(x))\right). \quad (\text{C14})$$

Finally, assuming that $\varepsilon < \max_x \psi(x)$, we have

$$O\left(\ln \frac{1}{\epsilon_{\text{coarse}}} \varepsilon \max_x \psi(x)\right). \quad (\text{C15})$$

Thus, if we want a target error ϵ_ψ on the total integral, using QAE we need to target an error $\varepsilon_p = \varepsilon = O(\epsilon_\psi / (\ln \frac{1}{\epsilon_{\text{coarse}}} \max_x \psi(x)))$ for each of the q_p . We may choose

an $\varepsilon = \Theta(\epsilon_\psi / (\ln \frac{1}{\epsilon_{\text{coarse}}} \max_x \psi(x)))$; with this, the cost of each QAE for every p run scales like $O(\ln \frac{1}{\epsilon_{\text{coarse}}} \max_x \psi(x) / \epsilon_\psi)$. Moreover, we may also need to set $\epsilon_{\text{coarse}} = \Theta(\epsilon_\psi)$. This gives a total query complexity that is going to scale like

$$O\left(\frac{\ln^2 \frac{1}{\epsilon_\psi} \max_x \psi(x)}{\epsilon_\psi}\right), \quad (\text{C16})$$

provided $\varepsilon \leq \max_x \psi(x)$. Also worth noting is that the time complexity of each p routine is

$$O\left(\frac{\ln \frac{1}{\epsilon_\psi} \max_x \psi(x)}{\epsilon_\psi}\right). \quad (\text{C17})$$

The overall query complexity of our method has a $\ln^2 \frac{1}{\epsilon_\psi}$ instead of the $\ln \frac{1}{\epsilon_\psi}$ factor of the time complexity due to the total sum estimations that can be run in parallel. Recalling that the $\ln \frac{1}{\epsilon_\psi}$ factor comes from the expression $n-l$, we note that for asymptotically small target error, we also recover a factor n^2 for the query complexity that the method using comparator circuits possesses (see Sec. II A). ■

APPENDIX D: PROOF OF LEMMA ON ERROR PROPAGATION TO THE DERIVATIVE OF INTERPOLANT

First, we look at the error on the vector of amplitudes coming from amplitude estimation and perturbed collocation nodes

$$|(f - f')_j| = O(\epsilon_\psi + \Lambda^2 |x_{\text{cheb},j} - x_{\text{mock},j}|), \quad (\text{D1})$$

where ϵ_ψ is the target error with which we estimate each $\Psi(s'_j)$ and the term $\Lambda^2 |x_{\text{cheb},j} - x_{\text{mock},j}|$ is coming from the perturbation of the nodes. We now assume that $\Lambda^2 |x_{\text{cheb},j} - x_{\text{mock},j}| = O(\epsilon_\psi)$, which is a loose assumption since $|x_{\text{cheb},j} - x_{\text{mock},j}| = O(1/2^n)$, an exponentially suppressed quantity. With this, we look at how the error is propagated to the expansion coefficients:

$$\|a - a'\|_1 = \|V_{\text{perturbed}}^{-1}(f - f')\|_1. \quad (\text{D2})$$

Using submultiplicativity, we obtain

$$\begin{aligned} \|a - a'\|_1 &= \|V_{\text{perturbed}}^{-1}(f - f')\|_1 \\ &\leq \|V_{\text{perturbed}}^{-1}\|_1 \|f - f'\|_1 \\ &\leq M \|V_{\text{perturbed}}^{-1}\|_1 \|f - f'\|_\infty \\ &\leq M^{3/2} \|V_{\text{perturbed}}^{-1}\|_2 \|f - f'\|_\infty \\ &= O(M^{3/2} \epsilon_\psi). \end{aligned} \quad (\text{D3})$$

Here, we have assumed $\|V_{\text{perturbed}}^{-1}\|_2 = O(1)$ (see Appendix A for proof). Ignoring the algorithmic error from using a finite-size M interpolation, we know that the propagated uncertainty is upper bounded by

$$\epsilon_{\psi^2} = \left| \sum_j (a - a')_j \frac{d}{dx} u_j(x) \right|$$

$$\begin{aligned} &\leq \sum_j \left| (a - a')_j \frac{d}{dx} u_j(x) \right| \\ &= \left\| (a - a') \odot \frac{d}{dx} u(x) \right\|_1, \end{aligned} \quad (\text{D4})$$

where $\odot$ symbolizes an elementwise multiplication of the vectors $(a - a')$ and $\frac{d}{dx} u(x)$. Now, using Hölder's inequality we obtain the following upper bound:

$$\left\| (a - a') \odot \frac{d}{dx} u(x) \right\|_1 \leq \| (a - a') \|_1 \left\| \frac{d}{dx} u(x) \right\|_\infty. \quad (\text{D5})$$

Here,

$$\left(\frac{d}{dx} u(x) \right)_j = \frac{d(u(x))_j}{dx} = \begin{cases} \sqrt{\frac{1}{M}} \frac{dT_j(x)}{dx}, & j = 0, \\ \sqrt{\frac{2}{M}} \frac{dT_j(x)}{dx}, & j \in \{1, \dots, M-1\}. \end{cases} \quad (\text{D6})$$

We have already established in Appendix A that

$$\max_s \left| \frac{dT_j(x)}{dx} \right| = O(j^2). \quad (\text{D7})$$

Thus, after differentiating the interpolant, we expect to have an error

$$\epsilon_{\psi^2} = O(M^3 \epsilon_\psi). \quad (\text{D8})$$

-
- [1] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [2] D. W. Berry, High-order quantum algorithm for solving linear differential equations, *J. Phys. A: Math. Theor.* **47**, 105301 (2014).
- [3] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Quantum algorithm for linear differential equations with exponentially improved dependence on precision, *Commun. Math. Phys.* **356**, 1057 (2017).
- [4] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, *SIAM J. Comput.* **46**, 1920 (2017).
- [5] A. M. Childs, J.-P. Liu, and A. Ostrander, High-precision quantum algorithms for partial differential equations, *Quantum* **5**, 574 (2021).
- [6] A. M. Childs and J.-P. Liu, Quantum spectral methods for differential equations, *Commun. Math. Phys.* **375**, 1427 (2020).
- [7] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2019), pp. 193–204.
- [8] D. Fang, L. Lin, and Y. Tong, Time-marching based quantum solvers for time-dependent linear differential equations, *Quantum* **7**, 955 (2023).
- [9] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, [arXiv:quant-ph/0005055](https://arxiv.org/abs/quant-ph/0005055).
- [10] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, *Nat. Phys.* **16**, 1050 (2020).
- [11] K. Miyamoto and H. Ueda, Extracting a function encoded in amplitudes of a quantum state by tensor network and orthogonal function expansion, *Quantum Inf. Process.* **22**, 239 (2023).
- [12] Y. Fang, C. Xue, T.-P. Sun, X.-F. Xu, X.-N. Zhuang, Y.-J. Wang, C.-C. Ye, T.-Y. Ma, J.-X. Zhang, H.-Y. Liu, *et al.*, PolyQROM: Orthogonal-polynomial-based quantum reduced-order model for flow field analysis, [arXiv:2504.21567](https://arxiv.org/abs/2504.21567).
- [13] C. Xue, Z.-Y. Chen, X.-N. Zhuang, Y.-J. Wang, T.-P. Sun, J.-C. Wang, H.-Y. Liu, Y.-C. Wu, Z.-L. Wang, and G.-P. Guo, End-to-end quantum vision transformer: Towards practical quantum speedup in large-scale models, [arXiv:2402.18940](https://arxiv.org/abs/2402.18940).
- [14] N. Linden, A. Montanaro, and C. Shao, Quantum vs. classical algorithms for solving the heat equation, *Commun. Math. Phys.* **395**, 601 (2022).
- [15] E. Devereux and A. Datta, Quantum algorithms for solving a drift-diffusion equation: A complexity analysis, *Phys. Rev. A* **112**, 042419 (2025).
- [16] M. Lubasch, Y. Kikuchi, L. Wright, and C. Mc Keever, Quantum circuits for partial differential equations in Fourier space, *Phys. Rev. Res.* **7**, 043326 (2025).
- [17] Y. Yuan, C. Wang, B. Wang, Z.-Y. Chen, M.-H. Dou, Y.-C. Wu, and G.-P. Guo, An improved QFT-based quantum comparator and extended modular arithmetic using one ancilla qubit, *New J. Phys.* **25**, 103011 (2023).
- [18] S. Wang, X. Li, W. J. B. Lee, S. Deb, E. Lim, and A. Chattopadhyay, A comprehensive study of quantum arithmetic circuits, *Philos. Trans. A* **383**, 20230392 (2025).
- [19] G. Rendon, All you need is Trotter, [arXiv:2311.01533v2](https://arxiv.org/abs/2311.01533v2).
- [20] G. Rendon, J. Watkins, and N. Wiebe, Improved accuracy for Trotter simulations using Chebyshev interpolation, *Quantum* **8**, 1266 (2024).
- [21] M. Kuian, L. Reichel, and S. Shiyanovskii, Optimally conditioned Vandermonde-like matrices, *SIAM J. Matrix Anal. Appl.* **40**, 1399 (2019).
- [22] F. L. Bauer and C. T. Fike, Norms and exclusion theorems, *J. Phys. A: Math. Theor.* **2**, 137 (1960).
- [23] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, On the LambertW function, *Adv. Comput. Math.* **5**, 329 (1996).
- [24] M. Hassani and A. Hoorfar, Inequalities on the Lambert function and hyperpower function, *JIPAM J. Inequal. Pure App. Math.* **9**, 51 (2008).