

Quantum algorithm for nonhomogeneous linear partial differential equationsJuan Miguel Arrazola,^{1,*} Timjan Kalajdzievski,¹ Christian Weedbrook,¹ and Seth Lloyd²¹*Xanadu, 372 Richmond Street W, Toronto, Ontario, Canada M5V 1X6*²*Massachusetts Institute of Technology, Department of Mechanical Engineering, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*

(Received 21 September 2018; revised manuscript received 22 February 2019; published 4 September 2019)

We describe a quantum algorithm for preparing states that encode solutions of nonhomogeneous linear partial differential equations. The algorithm is a continuous-variable version of matrix inversion: it efficiently inverts differential operators that are polynomials in the variables and their partial derivatives. The output is a quantum state whose wave function is proportional to a specific solution of the nonhomogeneous differential equation, which can be measured to reveal features of the solution. The algorithm consists of three stages: preparing fixed resource states in ancillary systems, performing Hamiltonian simulation, and measuring the ancilla systems. The algorithm can be carried out using standard methods for gate decompositions, but we improve this in two ways. First, we show that for a wide class of differential operators, it is possible to derive exact decompositions for the gates employed in Hamiltonian simulation. This avoids the need for costly commutator approximations, reducing gate counts by orders of magnitude. Additionally, we employ methods from machine learning to find explicit circuits that prepare the required resource states. We conclude by studying two example applications of the algorithm: solving Poisson's equation in electrostatics and performing one-dimensional integration.

DOI: [10.1103/PhysRevA.100.032306](https://doi.org/10.1103/PhysRevA.100.032306)**I. INTRODUCTION**

Following the discovery of quantum algorithms for factoring, database search, and universal simulation of quantum systems [1–3], decades of work have led to the uncovering of numerous quantum algorithms capable of outperforming existing classical methods. Examples include quantum algorithms for algebraic problems such as Pell's equation and the Jones polynomial [4–7], semidefinite programming [8,9], machine learning [10–12], and ordinary differential equations [13–21]. Quantum computers also excel at solving linear systems of equations [22–26]. Here, given an $N \times N$ sparse matrix A and a vector $\mathbf{b} = (b_1, \dots, b_N)$, the goal is to find a vector $\mathbf{x} = (x_1, \dots, x_N)$ satisfying the equation $A\mathbf{x} = \mathbf{b}$. Quantum algorithms for this problem take as input the quantum state $|\mathbf{b}\rangle = \sum_{i=1}^N b_i|i\rangle$ and efficiently perform matrix inversion to prepare the state $|\mathbf{x}\rangle = A^{-1}|\mathbf{b}\rangle$ encoding the solution of the linear system of equations.

We study a continuous version of this problem where the inputs are a function $f(\mathbf{x})$ over \mathbb{R}^N and a differential operator A . In its most general form, A is expressed as a function of the variables and their partial derivatives: $A = A(x_1, \dots, x_N, \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_N})$. The task is to find a function $\psi(\mathbf{x})$ satisfying the linear partial differential equation $A\psi(\mathbf{x}) = f(\mathbf{x})$, which is said to be nonhomogeneous whenever $f(\mathbf{x}) \neq 0$. In direct analogy to the case of a linear system of equations, a specific solution to the nonhomogeneous problem can be found by obtaining the inverse operator A^{-1} and computing the function $\psi(\mathbf{x}) = A^{-1}f(\mathbf{x})$.

In this work, we present a quantum algorithm for finding solutions to nonhomogeneous linear partial differential equations. More specifically, we show how to solve equations of the form $A\psi(\mathbf{x}) = f(\mathbf{x})$, where A is a polynomial in the variables and their partial derivatives. We describe the algorithm in the continuous-variable (CV) model of quantum computing [27,28], but the algorithm can be implemented in any model for universal quantum computing. Similar to quantum algorithms for linear systems of equations, the algorithm takes as input a state $|f\rangle$ encoding the nonhomogeneous function and outputs a state $|\psi\rangle$ whose wave function is proportional to a specific solution of the partial differential equation. In this sense, the algorithm is a continuous-variable version of the quantum algorithm for linear systems of equations. For differential equations of fixed order, the runtime is polynomial in the dimension—an exponential improvement over the best known classical techniques for solving partial differential equations.

The algorithm consists of three stages: preparing fixed resource states in ancillary systems, performing Hamiltonian simulation, and measuring the ancilla systems. Although the algorithm can be carried out using standard methods for gate decompositions, we improve on this in two ways. First, for several cases of interest, we introduce exact decomposition formulas that circumvent the use of commutator approximations for Hamiltonian simulation, leading to shorter circuits by orders of magnitude. Additionally, based on recent results on state preparation using quantum neural networks [29–31], we show how short-depth quantum circuits can be directly optimized to prepare required resource states with high fidelity. Finally, we validate the performance of the algorithm through numerical simulations by studying two example applications: solving Poisson's equation in electrostatics and performing one-dimensional integration.

*juanmiguel@xanadu.ai

II. QUANTUM ALGORITHM

The algorithm takes as inputs (i) a classical description of a linear differential operator A and (ii) a quantum state $|f\rangle$ of N registers with wave function $\langle \mathbf{x}|f\rangle = f(\mathbf{x})$, where $|\mathbf{x}\rangle = |x_1\rangle \dots |x_N\rangle$. For definiteness, we consider the registers to be modes of the quantized electromagnetic field with associated position \hat{X} and momentum \hat{P} quadrature operators. This choice is for convenience: the algorithm can in principle be carried out in any physical model of quantum computing.

The quadrature operators \hat{X}_k and \hat{P}_k acting on mode k can be defined in terms of their action on an arbitrary state: $\hat{X}_k \int dx^N \psi(\mathbf{x})|\mathbf{x}\rangle = \int dx^N x_k \psi(\mathbf{x})|\mathbf{x}\rangle$ and $\hat{P}_k \int dx^N \psi(\mathbf{x})|\mathbf{x}\rangle = -\frac{i}{2} \int dx^N \frac{\partial}{\partial x_k} \psi(\mathbf{x})|\mathbf{x}\rangle$ for all $k = 1, \dots, N$, where we have set $\hbar = 1/2$. Note that the action of the momentum operator is equivalent to differentiation with respect to position. A linear differential operator A can thus be equivalently cast as an operator \hat{A} on the Hilbert space of an N -mode quantum system. The operator \hat{A} is then a polynomial of the position and momentum operators. We focus on Hermitian operators, in which case \hat{A} can be viewed as a Hamiltonian for the N -mode system.

We follow the Fourier decomposition technique of Ref. [25], but note that the methods proposed in Ref. [32] for solving linear systems of equations using continuous-variable quantum computers could also potentially be adapted to solving partial differential equations. Let $g(x)$ be an odd function satisfying $\int_0^\infty g(x)dx = 1$. It holds that $a^{-1} = \int_0^\infty g(ax)dx$ for $a \neq 0$. Choosing $g(x) = x e^{-x^2/2}$ and writing $g(x)$ in terms of its Fourier transform $g(x) = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^\infty dy y e^{-y^2/2} e^{ixy}$ we have

$$a^{-1} = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^\infty dx \Theta(x) \int_{-\infty}^\infty dy y e^{-y^2/2} e^{-iaxy}, \quad (1)$$

where $\Theta(x)$ is the Heaviside step function. Let $\{|a\rangle\}$ be the eigenbasis of \hat{A} with corresponding eigenvalues $a \in \mathbb{R}$. Since \hat{A}^{-1} and $e^{-i\hat{A}xy}$ are both diagonal in the basis $\{|a\rangle\}$, Eq. (1) implies that \hat{A}^{-1} can be expressed as

$$\hat{A}^{-1} = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^\infty dx dy \Theta(x) y e^{-y^2/2} e^{-i\hat{A}xy}. \quad (2)$$

To implement the action of \hat{A}^{-1} on a target state $|f\rangle$, consider the unnormalized two-mode resource state

$$|s\rangle|1\rangle := \int_{-\infty}^\infty dx \Theta(x)|x\rangle \int_{-\infty}^\infty \frac{i}{\sqrt{2\pi}} dy y e^{-y^2/2}|y\rangle. \quad (3)$$

We refer to the state $|s\rangle$ as a step function state. In its current form, $|s\rangle$ is unnormalizable, but as we discuss shortly, this can be remedied by employing a step function of finite length. Additionally, we recognize $y e^{-y^2/2}$ as the unnormalized wave function of a single photon and consequently $|1\rangle$ as a single-photon state up to a global phase and normalization.

Given an input state $|f\rangle$, the algorithm starts by preparing the resource states of Eq. (3). A global unitary $e^{-i\hat{A}\hat{X}\hat{Y}}$ is subsequently applied to all systems, where \hat{X} and \hat{Y} are respectively the position operators of the two resource modes. This transformation is equivalent to performing evolution under a Hamiltonian $\hat{H} = \hat{A}\hat{X}\hat{Y}$ for unit time. The result is

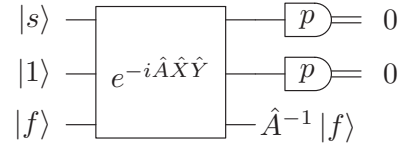


FIG. 1. Schematic representation of the quantum algorithm. The state $|f\rangle$ is given as input. Two resource states are prepared: a single photon $|1\rangle$ and a step function state $|s\rangle$. A global unitary $e^{-i\hat{A}\hat{X}\hat{Y}}$ is applied to all three systems, which is equivalent to evolution under the Hamiltonian $\hat{A}\hat{X}\hat{Y}$ for unit time. This is followed by a homodyne momentum measurement on the resource modes. Postselecting on the outcome $p = 0$ on both modes yields the desired output state $\hat{A}^{-1}|f\rangle$.

the output state

$$|\Psi\rangle = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^\infty dx dy \Theta(x) y e^{-y^2/2} e^{-i\hat{A}xy} |f\rangle |x\rangle |y\rangle. \quad (4)$$

Performing a momentum homodyne measurement on both resource modes and postselecting on observing the outcome $p = 0$ on both modes, i.e., projecting onto the state $|0_{p_x}\rangle|0_{p_y}\rangle$, yields

$$\begin{aligned} & (\mathbb{1} \otimes |0_{p_x}\rangle\langle 0_{p_x}| \otimes |0_{p_y}\rangle\langle 0_{p_y}|) |\Psi\rangle \\ &= \left(\frac{i}{\sqrt{2\pi}} \int_{-\infty}^\infty dx dy \Theta(x) y e^{-y^2/2} e^{-i\hat{A}xy} |f\rangle \right) |0_{p_x}\rangle |0_{p_y}\rangle \\ &= (\hat{A}^{-1}|f\rangle) |0_{p_x}\rangle |0_{p_y}\rangle, \end{aligned} \quad (5)$$

where we have used the relation $\langle 0_{p_x}|x\rangle = \langle 0_{p_y}|y\rangle = 1$. The output is thus the desired state $|\psi\rangle = \hat{A}^{-1}|f\rangle$ with wave function $\psi(\mathbf{x}) = A^{-1}f(\mathbf{x})$ up to normalization. The algorithm is depicted in Fig. 1. An ideal step function state is unphysical since its wave function is not square integrable. Instead, we consider a step function state of finite width L given by

$$|s_L\rangle = \frac{1}{\sqrt{L}} \int_0^L dx |x\rangle. \quad (6)$$

The result of employing this state in the algorithm is an output state

$$|\psi\rangle = \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^\infty dy y e^{-y^2/2} e^{-i\hat{A}xy} |f\rangle. \quad (7)$$

In this case, instead of the ideal inverse operator \hat{A}^{-1} , the operator being applied to $|f\rangle$ is a truncated Fourier decomposition of \hat{A}^{-1} . Here and henceforth we use $\hat{A}_{\text{approx}}^{-1}$ to denote any approximation to the ideal inverse operator. The effect of a finite width is best understood by considering the action of $\hat{A}_{\text{approx}}^{-1}$ on an eigenstate $|a\rangle$:

$$\begin{aligned} \hat{A}_{\text{approx}}^{-1}|a\rangle &= \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^\infty dy y e^{-y^2/2} e^{-i\hat{A}xy} |a\rangle \\ &= \left(\frac{1}{a} - \frac{e^{-a^2 L^2}}{a} \right) |a\rangle. \end{aligned} \quad (8)$$

The effect of a truncated step function is an exponentially small correction from the ideal result $a^{-1}|a\rangle$. The correction is only significant for small eigenvalues such that $a \lesssim 1/L$, i.e.,

the value of L determines the smallest eigenvalue a for which the approximation is adequate. This result can then be used to analyze the effect of the approximation on an arbitrary state $|f\rangle$ whenever it is possible to express it in the eigenbasis of \hat{A} : $|f\rangle = \int da \phi(a)|a\rangle$.

Even a step function of finite width is an idealization since it is not continuous at either $x = 0$ or $x = L$: any physical wave function will exhibit a smooth transition around these points. The effect of this finite rise time can be modeled by approximating the ideal step function in terms of a continuous function. Here we consider the error function $\frac{1}{2}(1 + \text{erf}[kx])$ as an approximate step function, where the approximation improves with larger $k > 0$. We then have

$$\begin{aligned} \hat{A}_{\text{approx}}^{-1}|a\rangle &= \frac{i}{\sqrt{8\pi}} \int_0^\infty dx (1 + \text{erf}[kx]) \int_{-\infty}^\infty dy y e^{-y^2/2} e^{-i\hat{A}xy} |a\rangle \\ &= \frac{1}{a} \left(\frac{\sqrt{2}}{\sqrt{2 + \frac{a^2}{k^2}}} \right) |a\rangle = \frac{1}{a} \left[1 - \frac{a^2}{2k^2} + O\left(\frac{a^4}{k^4}\right) \right] |a\rangle. \end{aligned} \quad (9)$$

This induces another correction from the ideal scenario, but in this case the effect is significant only for large eigenvalues a such that $a \gtrsim k$. Thus approximations to an ideal step function state lead to deviations that are relevant only for very small or very large eigenvalues.

Finally, physical homodyne measurements have finite precision, whose effect on the resulting output state must be taken into account. We model the finite-precision homodyne measurement as a projection onto finitely squeezed states, as opposed to momentum eigenstates which are infinitely squeezed. This is the same strategy employed, for example, in classical simulators [33]. The state corresponding to the $p = 0$ outcome of a homodyne measurement with precision Δ is $|\Delta\rangle = \frac{1}{\pi^{1/4}\sqrt{\Delta}} \int dp e^{-p^2/2\Delta^2} |p\rangle$. The resulting output state is given by

$$\begin{aligned} &(\mathbb{1} \otimes |\Delta\rangle \langle \Delta| \otimes |\Delta\rangle \langle \Delta|) |\Psi\rangle \\ &= \frac{i}{\sqrt{2\pi}} \int_0^\infty dx \int_{-\infty}^\infty dy y e^{-y^2/2} e^{-i\hat{A}xy} g(x, y, \Delta) |f\rangle |\Delta\rangle |\Delta\rangle \\ &= (\hat{A}_{\text{approx}}^{-1} |f\rangle) |\Delta\rangle |\Delta\rangle, \end{aligned} \quad (10)$$

where

$$\begin{aligned} g(x, y, \Delta) &= \frac{1}{\sqrt{\pi}\Delta} \int_{-\infty}^\infty dp dq e^{-p^2/2\Delta^2 - ipx} e^{-q^2/2\Delta^2 - iqy} \\ &= e^{-(x^2+y^2)\Delta^2/2}. \end{aligned} \quad (11)$$

As before, the approximation of the inverse operator is best expressed in terms of its action on the eigenstate $|a\rangle$:

$$\begin{aligned} \hat{A}_{\text{approx}}^{-1}|a\rangle &= \frac{i}{\sqrt{2\pi}} \int_0^\infty dx \int_{-\infty}^\infty dy y e^{-y^2/2} e^{-i\hat{A}xy} g(x, y, \Delta) |a\rangle \\ &= \frac{2a\sqrt{\pi}\Delta}{\sqrt{1 + \Delta^2(a^2 + \Delta^2 + \Delta^4)}} |a\rangle \\ &= \left[\frac{2\sqrt{\pi}\Delta}{a} + O\left(\frac{\Delta^3}{a^3}\right) \right] |a\rangle. \end{aligned} \quad (12)$$

Up to normalization, this state is equal to the desired state $a^{-1}|a\rangle$ except for a correction $O(\Delta^3/a^3)$ that is only relevant when $a \lesssim \Delta$.

Comparing to Eq. (8), the dominant error for small values of a arises from the finite width of the step function state, whose effect is exponential in a for $a \lesssim 1/L$.

By expressing the state $|f\rangle$ in terms of the eigenbasis $\{|a\rangle\}$ of \hat{A} , we note that the action of $\hat{A}_{\text{approx}}^{-1}$ on $|f\rangle$ introduces an overall constant factor $2\sqrt{\pi}\Delta$ and therefore the probability of successfully projecting onto the desired output state satisfies $\text{Pr}(\text{success}) = \|\hat{A}_{\text{approx}}^{-1}|f\rangle\|^2 = O(\Delta^2)$. In the Appendixes, we discuss how coarse-graining measurement outputs can also be used to increase the probability of success at the cost of an increase in the approximation error.

Combining the effects of a finite-width step function and a finite-precision measurement leads to an approximation

$$\begin{aligned} \hat{A}_{\text{approx}}^{-1}|a\rangle &= \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^\infty dy y e^{-y^2/2} e^{-i\hat{A}xy} g(x, y, \Delta) |a\rangle \\ &= \frac{2a\sqrt{\pi}\Delta(1 - e^{L^2(a^2 + \Delta^2 + \Delta^4)/2(1 + \Delta^2)})}{\sqrt{1 + \Delta^2(a^2 + \Delta^2 + \Delta^4)}} |a\rangle \\ &= 2\sqrt{\pi}\Delta F(a) |a\rangle, \end{aligned} \quad (13)$$

where

$$F(a) := \frac{a(1 - e^{-L^2(a^2 + \Delta^2 + \Delta^4)/2(1 + \Delta^2)})}{\sqrt{1 + \Delta^2(a^2 + \Delta^2 + \Delta^4)}} |a\rangle, \quad (14)$$

which is an approximation to a^{-1} .

The imprecision of the algorithm arises due to the difference between the ideal inverse operator \hat{A}^{-1} and the resulting approximation $\hat{A}_{\text{approx}}^{-1}$. We can express \hat{A} in its eigenbasis as $\hat{A} = \sum_a \sum_{k=1}^{m(a)} a|a, k\rangle \langle a, k|$, where $\hat{A}|a, k\rangle = a|a, k\rangle$ and $m(a)$ is the multiplicity of eigenvalue a . We then have

$$\hat{A}^{-1} = \sum_a \sum_{k=1}^{m(a)} \frac{1}{a} |a, k\rangle \langle a, k|, \quad (15)$$

$$\hat{A}_{\text{approx}}^{-1} = \sum_a \sum_{k=1}^{m(a)} F(a) |a, k\rangle \langle a, k|. \quad (16)$$

The relationship between $F(a)$ and a^{-1} is illustrated in Fig. 2 for $L = 7$ and $\Delta = 0.1$.

Finally, the error in the approximation can be quantified in terms of the distance between the resulting output states. More precisely, let

$$|\psi\rangle = \hat{A}^{-1}|f\rangle = \sum_a \sum_{k=1}^{m(a)} \frac{1}{a} c_{a,k} |a, k\rangle, \quad (17)$$

$$|\tilde{\psi}\rangle = \hat{A}_{\text{approx}}^{-1}|f\rangle = \sum_a \sum_{k=1}^{m(a)} F(a) c_{a,k} |a, k\rangle \quad (18)$$

be unnormalized output states. Defining the error as $\varepsilon := \|\psi - \tilde{\psi}\|_1$ we obtain

$$\varepsilon = \sum_a \sum_{k=1}^{m(a)} |c_{a,k}| \left| \frac{1}{a} - F(a) \right|. \quad (19)$$

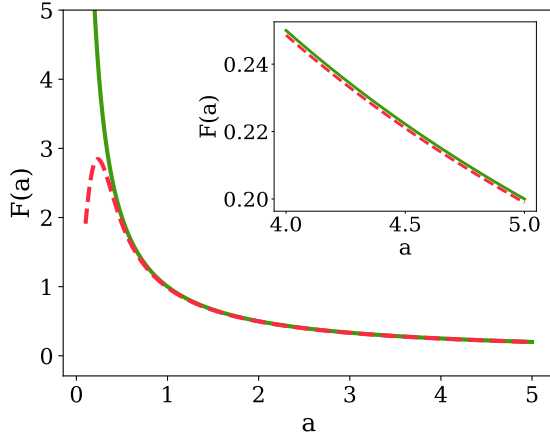


FIG. 2. Function a^{-1} (solid green) and the approximation $F(a)$ (dashed red) for $L = 7$ and $\Delta = 0.1$. The inset shows a closeup of the two functions for larger values of a . The approximation is excellent except for small values of a .

This error is significant if there are terms in the sum for which both $|c_{a,k}|$ and $|1/a - F(a)|$ are large. For $a \gg 1$, it holds that $|1/a - F(a)| \approx (a^2 + 2)\Delta^2/(2a^3) + O(\Delta^4) = O(\Delta^2/a)$. This error is bounded even for large a . For $a \ll 1/L$ we have $F(a) \approx 0$ and therefore $|1/a - F(a)| \approx 1/a$, which can be very large for small a . This is the dominant source of error arising from the approximation of the inverse operator.

To choose L , for a given $\delta > 0$, we define the effective eigenvalue support of state $|f\rangle$ as $\text{supp}_\delta(f) = \{a : \exists k \text{ such that } |c_{a,k}| > \delta\}$. Normalizing the differential operator as $\|\hat{A}\|_\infty = 1$, we define

$$\kappa_\delta := \max_{a \in \text{supp}_\delta(f)} (1/a) \quad (20)$$

as the condition number of \hat{A} when restricted to the effective support of $|f\rangle$. Then, we can set the resulting error in the output state to sufficiently low values by choosing $L = O(\kappa_\delta^2)$. Overall, the algorithm is valid for any Hermitian differential operator \hat{A} , but special care needs to be taken for input states that have a large support over the smallest eigenvalues of \hat{A} .

III. HAMILTONIAN SIMULATION

The goal of Hamiltonian simulation is to find a quantum circuit that performs the transformation $e^{i\hat{H}t}$ for some Hamiltonian \hat{H} and time $t > 0$. The circuit is specified in terms of a universal gate set, which in this work we take to be the set

$$\{e^{i\frac{\pi}{2}(\hat{X}^2 + \hat{P}^2)}, e^{it_1\hat{X}}, e^{it_2\hat{X}^2}, e^{it_3\hat{X}^3}, e^{i\tau\hat{X}_1 \otimes \hat{X}_2}\}, \quad (21)$$

where t_1, t_2, t_3 , and τ are adjustable real parameters. The Fourier transform gate $\hat{F} := e^{i\frac{\pi}{2}(\hat{X}^2 + \hat{P}^2)}$ has the effect of mapping between the quadrature operators: $\hat{F}^\dagger \hat{X} \hat{F} = -\hat{P}$ and $\hat{F}^\dagger \hat{P} \hat{F} = \hat{X}$. The standard approach for performing Hamiltonian simulation is to employ a Trotter-Suzuki decomposition [34–37] to express the transformation $e^{it\hat{H}} = e^{it\sum_{j=1}^M \hat{H}_j}$ in terms of the product $e^{it\hat{H}} = \prod_{j=1}^M (e^{i\frac{t}{K}\hat{H}_j})^K + O(t^2/K)$. Following this, a sequence of commutator approximations are typically employed to decompose each term $e^{i\frac{t}{K}\hat{H}_j}$ into elements from the universal set [38]. We focus on the case where

the Hamiltonian \hat{A} can be expressed as

$$\hat{A} = \lambda \mathbb{1} + \sum_{j=1}^N a_j \hat{X}_j + b_j \hat{P}_j + \alpha_j \hat{X}_j^2 + \beta_j \hat{P}_j^2, \quad (22)$$

where $\lambda, a_j, b_j, \alpha_j$, and β_j are real constants. This form encompasses a large class of differential operators, including for instance those defining Poisson’s equation, the heat equation, and the wave equation.

In the quantum algorithm, we perform evolution under the Hamiltonian $\hat{A}\hat{X}\hat{Y}$ for unit time. When \hat{A} is of the form of Eq. (22), after performing a Trotter-Suzuki decomposition, each term in the product will correspond—up to Fourier transforms exchanging \hat{X} and \hat{P} —to unitaries of the form $e^{it\hat{X}_j\hat{X}_k}$, $e^{it\hat{X}_j\hat{X}_k\hat{X}_l}$, or $e^{it\hat{X}_j^2\hat{X}_k\hat{X}_l}$, where the subindices denote which mode the operators act on. We now show how exact decompositions can be found for each of these transformations. An extension of this technique for more general gates can be found in Ref. [39].

First, note that the transformation $e^{it\hat{X}_j\hat{X}_k}$ is already part of the universal set. For the unitary $e^{it\hat{X}_j\hat{X}_k\hat{X}_l}$, it can be shown (see the Appendixes for details) that the following exact decomposition holds:

$$\begin{aligned} e^{i2t\hat{X}_j\hat{X}_k\hat{X}_l} &= e^{i2\hat{P}_j\hat{X}_k} e^{i2\hat{P}_j\hat{X}_l} e^{\frac{i\pi}{3}\hat{X}_j^3} e^{-i2\hat{P}_j\hat{X}_l} e^{-i2\hat{P}_j\hat{X}_k} \\ &\times e^{i2\hat{P}_k\hat{X}_l} e^{-\frac{i\pi}{3}\hat{X}_k^3} e^{-i2\hat{P}_k\hat{X}_l} e^{i2\hat{P}_j\hat{X}_j} e^{-\frac{i\pi}{3}\hat{X}_j^3} e^{-i2\hat{P}_j\hat{X}_k} \\ &\times e^{i2\hat{P}_j\hat{X}_k} e^{-\frac{i\pi}{3}\hat{X}_j^3} e^{-i2\hat{P}_j\hat{X}_k} e^{\frac{i\pi}{3}\hat{X}_j^3} e^{\frac{i\pi}{3}\hat{X}_k^3} e^{\frac{i\pi}{3}\hat{X}_l^3}. \end{aligned} \quad (23)$$

Note that gates of the form $e^{-i2\hat{P}_j\hat{X}_k}$ are equivalent to a controlled-phase gate up to Fourier transforms on the first mode. Finally, as shown in the Appendix, for the gate $e^{it\hat{X}_j^2\hat{X}_k\hat{X}_l}$, it is possible to derive an exact decomposition

$$e^{i2\alpha\hat{X}_j^2\hat{P}_k\hat{P}_l} = e^{2i\alpha\hat{X}_j\hat{X}_k\hat{X}_l} e^{-i\alpha\hat{X}_j^2\hat{P}_k^2} e^{-2i\alpha\hat{X}_j\hat{X}_k\hat{X}_l} e^{i\alpha\hat{X}_j^2\hat{P}_k^2} e^{-2i\alpha\hat{X}_j^2\hat{P}_l}. \quad (24)$$

Here, unitaries of the form $e^{i\alpha\hat{X}_j^2\hat{P}_k^2}$ and $e^{-2i\alpha\hat{X}_j^2\hat{P}_l}$ are not part of the universal set, but exact decompositions can also be derived for them (see the Appendixes for details). The resulting exact decomposition for the gate $e^{it\hat{X}_j^2\hat{X}_k\hat{X}_l}$ contains 281 gates from the universal set. It is important to contrast this with the commutator approximation method [38], which requires 28 gates to decompose $e^{it\hat{X}_j^2\hat{X}_k\hat{X}_l}$, but for a precision of 10^{-3} , it needs about 10^6 repetitions for a total of roughly 10^7 gates.

For any sparse Hamiltonian that is a polynomial of constant degree over the quadrature operators, universal simulation theorems [27,28] state that $\text{poly}(N)$ time is required to perform Hamiltonian simulation and therefore to run our quantum algorithm for partial differential equations. This is an exponential improvement over classical algorithms for solving PDEs, which scale exponentially with dimension [40–42]. The runtime of Hamiltonian simulation also scales polynomially on the operator norm $\|\hat{A}\|_\infty$ [3], so care must be taken to ensure that this norm is well behaved over the support of the input state $|f\rangle$.

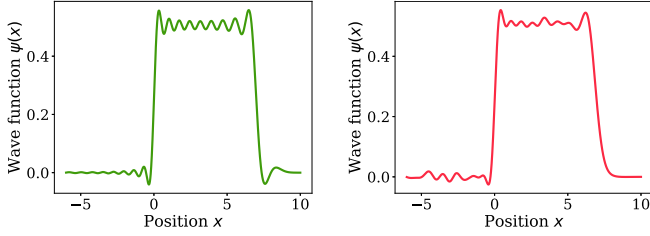


FIG. 3. (Top) Wave functions of the target step function state with cutoff $d = 41$ and width $L = 7$. (Bottom) The state prepared by the quantum neural network, with fidelity of 99.36% to the target state. The network consists of 30 layers for a total of 150 gates.

IV. RESOURCE STATE PREPARATION

We employ results from Refs. [29,30] to find circuits for preparing the single photon and step function resource states required in the algorithm. The strategy is to optimize a quantum neural network which takes a single-mode vacuum state as input and prepares a desired target state as output. A layer \mathcal{L} of the quantum neural network is composed of the sequence of gates [29]: $\mathcal{L} := K(\kappa)D(\alpha)R(\phi_2)S(r,\theta)R(\phi_1)$, where $R(\theta)$ is a rotation gate, $D(\alpha)$ is a displacement gate, $S(r)$ is a squeezing gate, and $K(\kappa)$ is a Kerr gate. The rotation, squeezing, and displacement gates are Gaussian and can be straightforwardly decomposed in terms of the universal set of Eq. (21). The Kerr gate can be decomposed using results from [38].

We perform optimization of the gate parameters by employing the TensorFlow [43] backend of the Strawberry Fields software platform for photonic quantum computing [33]. This approach has been pursued in Ref. [30], where it was shown that a single photon state can be prepared using a quantum neural network of eight layers, i.e., 40 gates, with fidelity 99.998%. For the target step function state, we consider the truncated state $|s_L\rangle = \frac{1}{\sqrt{L}} \int_0^L dx |x\rangle = \sum_{n=0}^{\infty} c_{n,L} |n\rangle$, where $|n\rangle$ is the Fock state of n photons and $c_{n,L} = \langle n | s_L \rangle$. For numerical simulations, we introduce a cutoff dimension d , yielding the truncated state $|s_{d,L}\rangle = \sum_{n=0}^d c_{n,L} |n\rangle$. As an example, we set a width of $L = 7$ and a cutoff $d = 41$, fixing a quantum neural network with 30 layers (150 gates) to prepare this state. The result is a network that can prepare a state with 99.36% fidelity to the target state $|s_{41,7}\rangle$. This is shown in Fig. 3, where we plot the wave function of both states.

As in Sec. II, we evaluate the resulting approximation to the inverse operator \hat{A}^{-1} by computing the function

$$G(a) = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx \sum_{n=0}^d \gamma_n \psi_n(x) \int_{-\infty}^{\infty} dy y e^{-y^2/2} e^{-iaxy}, \tag{25}$$

where $\Psi_p(x) = \sum_{n=0}^d \gamma_n \psi_n(x)$ is the wave function of the state $|s_p\rangle = \sum_{n=0}^d \gamma_n |n\rangle$ prepared by the network. As shown in Fig. 4, $G(a)$ is also an approximation to the ideal case a^{-1} . However, here the approximation deviates more significantly from the ideal case for large values of a . As discussed previously, this can be understood from the fact that the wave function of the output state is itself an approximation of an ideal step function state of finite width.

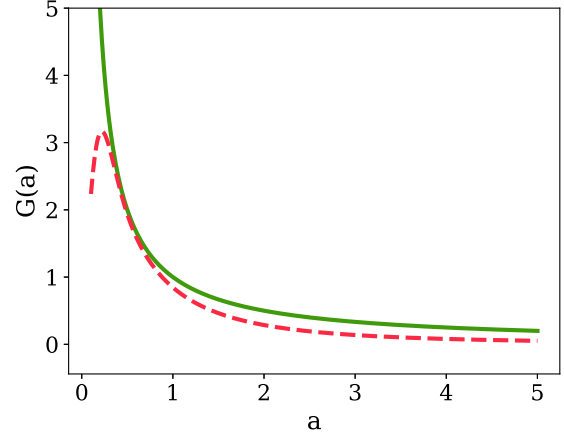


FIG. 4. Function a^{-1} (solid green) and the approximation $G(a)$ (dashed red) arising from the use of the step function state prepared by the quantum neural network.

V. EXAMPLES

In this section, we test the correctness and applicability of the algorithm by analyzing its performance on two low-dimensional problems where full classical simulation is possible.

A. Poisson equation

Poisson’s equation is the nonhomogeneous partial differential equation

$$\nabla^2 \psi(\mathbf{x}) = \sum_{i=1}^n \frac{\partial^2 \psi(\mathbf{x})}{\partial x_i^2} = f(\mathbf{x}), \tag{26}$$

which has applications across several areas of physics and engineering. Here we consider its relevance to electrostatics, where it establishes a relationship between a charge distribution $\rho(\mathbf{x})$ and the electric potential $\phi(\mathbf{x})$, namely $\nabla^2 \phi(\mathbf{x}) = -\frac{\rho(\mathbf{x})}{\epsilon}$, where ϵ is the permittivity of the medium, whose value we fix to $\epsilon = 1$. To apply our quantum algorithm to this problem, note that under the convention $\hbar = 1/2$, it holds that $\frac{\partial^2}{\partial x_i^2} = -4\hat{p}_i^2$ and thus we can set $\hat{A} = -4 \sum_{i=1}^n \hat{p}_i^2$. We consider a two-dimensional problem where the charge distribution is given by $\rho(x, y) = xy e^{-\frac{x^2}{2}} e^{-\frac{y^2}{2}}$, as shown in Fig. 5. This charge distribution is equivalent, up to normalization, to the wave function of the two-mode input state $|f\rangle = |1\rangle|1\rangle$ consisting of a single photon in each mode. We compute the output state by constructing the operator $\hat{A}_{\text{approx}}^{-1}$ as in Eq. (13) with $L = 20$ and $\Delta = 0.01$, then applying it to the input state $|f\rangle$. The wave function of the output state is proportional to the electric potential $\phi(x, y)$, which can be used to compute the electric field $\vec{E}(x, y) = -\nabla\phi(x, y)$. The fidelity between the output state of the algorithm and the ideal solution state is 99.9%. To illustrate the validity of the solution, the charge distribution and the electric field are shown in Fig. 5. In a physical implementation of the algorithm, repeated quadrature measurements of the output state would reveal regions of large electrostatic potential.

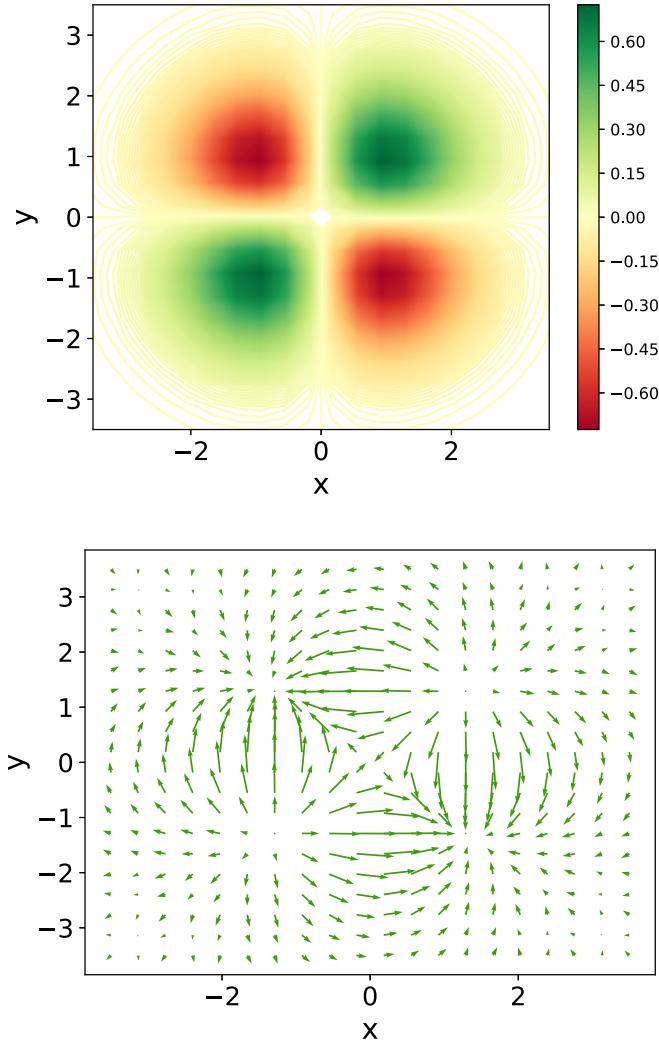


FIG. 5. (Top) Charge distribution $\rho(x, y) = xy e^{-\frac{x^2}{2}} e^{-\frac{y^2}{2}}$. The top-right and bottom-left quadrants are regions of positive charge, while the remaining quadrants are negatively charged. (Bottom) Electric-field lines reconstructed from the output state of the quantum algorithm. There are regions of zero electric field in each quadrant that arise from interfering contributions of the charge clouds surrounding these points. The electric field is also zero at the origin, as expected from the symmetry of the charge distribution.

B. One-dimensional integration

The simplest nonhomogeneous differential equation is the one-dimensional equation $A\psi(x) := \frac{d\psi(x)}{dx} = f(x)$. A solution to the equation is

$$A^{-1}f(x) = \int dx f(x), \tag{27}$$

i.e., the solution is the indefinite integral of $f(x)$. To apply the quantum algorithm to this problem we set $\hat{A} = \hat{P}$, in which case the output of the algorithm is the state $|\psi\rangle = \hat{P}^{-1}|f\rangle$ whose wave function is proportional to

$$\psi(x) = 2i \int dx f(x). \tag{28}$$

This is equal to the desired solution up to a global phase. In short, the quantum algorithm performs one-dimensional integration. To calculate the output state of the algorithm, we numerically compute the operator

$$\hat{A}_{\text{approx}}^{-1} = \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^{\infty} dy y e^{-y^2/2} e^{-iAxy} g(x, y, \Delta) \tag{29}$$

as in Eq. (13) and consequently calculate $\hat{A}_{\text{approx}}^{-1}|f\rangle$.

This computation is performed by expressing \hat{P} in the Fock basis, truncating to a finite photon number, and approximating the integral by a Riemann sum. The operator $\hat{A}_{\text{approx}}^{-1}$ includes the effects of a finite-width step function state and limited precision measurement, but not of approximations to the step function state. We choose $f(x) = \sin(\omega x)e^{-x^2/(2\sigma^2)}$ as the function to integrate. The corresponding input state $|f\rangle = \sum_n c_n |n\rangle$ can be obtained by computing the coefficients $c_n = \int_{-\infty}^{\infty} dx f(x)\psi_n(x)$, where $\psi_n(x)$ is the wave function of the Fock state with n photons.

The results are shown in Fig. 6 where we plot $f(x) = \sin(\omega x)e^{-x^2/(2\sigma^2)}$ and the wave function of the output state. The output wave function closely reproduces the integral of $f(x)$, even when considering a finite-width step function state with parameter $L = 7$ and measurements with finite precision $\Delta = 0.1$.

VI. CONCLUSION

We have presented a quantum algorithm for preparing quantum states that encode the solution to nonhomogeneous linear partial differential equations. The algorithm is a continuous-variable version of the quantum algorithm for linear systems of equations. For differential operators of fixed degree, the runtime is polynomial in the dimension N : an exponential improvement compared to classical algorithms that compute the full solution of partial differential equations. However, there are important differences between this quantum algorithm and classical approaches: the quantum algorithm assumes that the input state $|f\rangle$ encoding the nonhomogeneous term of the equation can be efficiently prepared and the output is not an explicit specification of the solution, but instead a state whose wave function is proportional to the solution. It is crucial to identify applications where input states can be efficiently prepared and where sampling from the output state—for example to compute expectation values—is enough for the task at hand. Note also that recent breakthroughs in classical algorithms for linear systems of equations [44,45] apply only to low-rank operators, which do not appear in practice in solving partial differential equations.

Finally, the quantum algorithm provides a specific solution to the nonhomogeneous equation, but to solve general boundary problems it is necessary to also incorporate solutions to the homogeneous equation. The solution $\psi(\mathbf{x})$ of a nonhomogeneous partial differential equation can always be written as $\psi(\mathbf{x}) = \psi_H(\mathbf{x}) + \psi_{NH}(\mathbf{x})$, where $\psi_H(\mathbf{x})$ is a general solution to the homogeneous equation and $\psi_{NH}(\mathbf{x})$ is a specific solution to the nonhomogeneous equation. The corresponding boundary problem can be solved by first finding an arbitrary solution $\psi_{NH}(\mathbf{x})$ to the nonhomogeneous differential equation, then finding a solution to the homogeneous problem

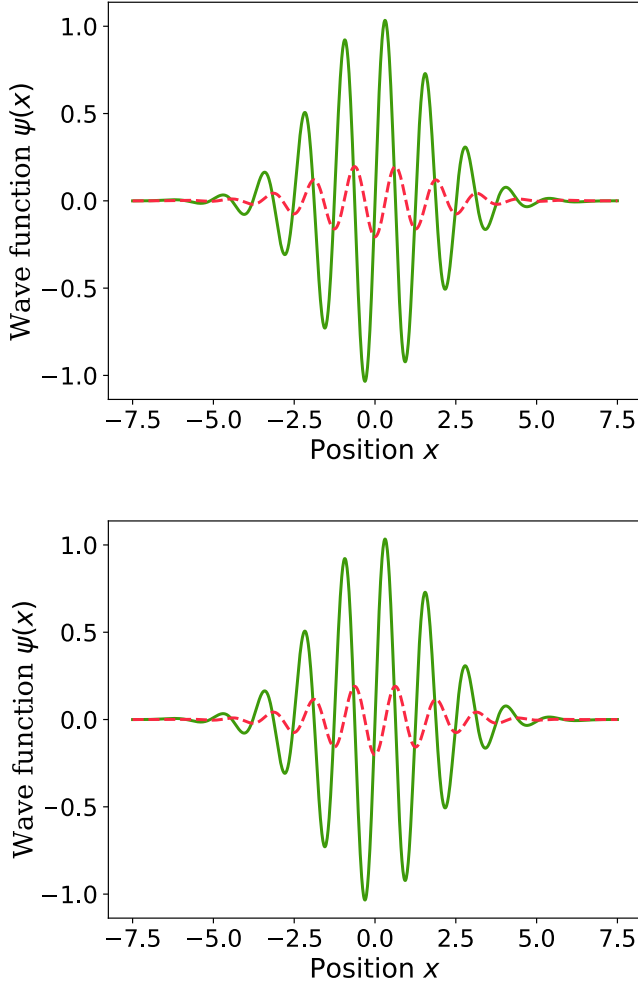


FIG. 6. (Top) Function $f(x) = \sin(\omega x)e^{-x^2/(2\sigma^2)}$ (solid green) with $\omega = 5$ and $\sigma = 1.8$. The wave function of the output state of the algorithm is also shown (dashed red). (Bottom) Function $f(x) = \sin(\omega x)e^{-x^2/(2\sigma^2)}$ with $\omega = 5$ (solid green) and its integral (dashed red), which can be computed analytically. The wave function of the output state of the algorithm is almost identical to integral, as desired. In both cases we consider an approximate inverse operator $\hat{A}_{\text{approx}}^{-1}$ with parameters $L = 7$ and $\Delta = 0.1$, demonstrating that the effect of a finite-width step function state and finite measurement precision does not significantly affect the correctness of the algorithm.

satisfying the boundary conditions. Through sampling from the solution state, the quantum algorithm can provide information that aids in solving the nonhomogeneous problem, which is usually the most challenging part. Once this has been achieved, the full solution can be obtained by solving the corresponding homogeneous boundary problem.

ACKNOWLEDGMENTS

We thank A. Ignjatovic, N. Killoran, T. Bromley, and N. Quesada for helpful discussions. S.L. was funded by AFOSR under a MURI on Optimal Quantum Measurements and State Verification, by IARPA under the QEO program, by ARO, and by NSF.

APPENDIX A: EXACT DECOMPOSITIONS

We use the convention $[\hat{X}, \hat{P}] = i/2$ and begin with the decomposition below:

$$\begin{aligned} e^{i2\delta\hat{X}_j\hat{X}_k\hat{X}_l} &= e^{i2\hat{P}_j\hat{X}_k} e^{i2\hat{P}_j\hat{X}_l} e^{\frac{i\delta}{3}\hat{X}_j^3} e^{-i2\hat{P}_j\hat{X}_l} e^{-i2\hat{P}_j\hat{X}_k} e^{i2\hat{P}_j\hat{X}_l} e^{-\frac{i\delta}{3}\hat{X}_k^3} e^{-i2\hat{P}_k\hat{X}_l} \\ &\times e^{i2\hat{P}_l\hat{X}_j} e^{-\frac{i\delta}{3}\hat{X}_l^3} e^{-i2\hat{P}_l\hat{X}_k} e^{i2\hat{P}_j\hat{X}_k} e^{-\frac{i\delta}{3}\hat{X}_j^3} e^{-i2\hat{P}_j\hat{X}_k} e^{\frac{i\delta}{3}\hat{X}_j^3} \\ &\times e^{\frac{i\delta}{3}\hat{X}_k^3} e^{\frac{i\delta}{3}\hat{X}_l^3}. \end{aligned} \quad (\text{A1})$$

This equation can be best understood by looking at the right hand and building each term in sequence. To begin, note that the first four cubic operators in the decomposition are surrounded by operators of the form $e^{i2\hat{P}_j\hat{X}_k}$ which can be expanded with unitary conjugation:

$$e^{i2\hat{P}_j\hat{X}_k} e^{i\delta\hat{X}_j^3} e^{-i2\hat{P}_j\hat{X}_k} = e^{i\delta(\hat{X}_j + \hat{X}_k)^3}. \quad (\text{A2})$$

The first one in mode j is translated by the k and l modes, leading to an exponent $(\hat{X}_j + \hat{X}_k + \hat{X}_l)^3$. Similarly, the other three cubic gates lead to the exponents $(\hat{X}_k + \hat{X}_l)^3$, $(\hat{X}_l + \hat{X}_j)^3$, and $(\hat{X}_j + \hat{X}_k)^3$. Expanding these polynomials gives a series of operators which can be simplified to give the cubic gate on the left-hand side.

The second decomposition we use is

$$e^{2i\alpha^2\hat{X}_j^2\hat{P}_k\hat{P}_l} = e^{2i\alpha\hat{X}_j\hat{X}_k\hat{X}_l} e^{-i\alpha\hat{X}_j^2\hat{P}_k} e^{-2i\alpha\hat{X}_j\hat{X}_k\hat{X}_l} e^{i\alpha\hat{X}_j^2\hat{P}_k} e^{-2i\alpha\hat{X}_j^2\hat{P}_l}. \quad (\text{A3})$$

For this decomposition we use operations $e^{i\alpha\hat{X}_j^2\hat{P}_k}$ and $e^{-2i\alpha\hat{X}_j^2\hat{P}_l}$, which are not in the universal set. For these, we require the decompositions

$$\begin{aligned} e^{i3\alpha^2k\hat{P}_k\hat{X}_j^2} &= e^{i2\alpha\hat{X}_j\hat{X}_k} e^{ik\hat{P}_k^3} e^{-i\alpha\hat{X}_j\hat{X}_k} e^{-ik\hat{P}_k^3} e^{-i2\alpha\hat{X}_j\hat{X}_k} \\ &\times e^{ik\hat{P}_k^3} e^{i\alpha\hat{X}_j\hat{X}_k} e^{-ik\hat{P}_k^3} e^{i\alpha^3k\frac{3}{4}\hat{X}_j^3} \end{aligned} \quad (\text{A4})$$

and

$$e^{i\alpha\hat{X}_j^2\hat{X}_k^2} = e^{i2\hat{P}_j\hat{X}_k} e^{i\frac{\alpha}{12}\hat{X}_j^4} e^{-i4\hat{P}_j\hat{X}_k} e^{i\frac{\alpha}{12}\hat{X}_j^4} e^{i2\hat{P}_j\hat{X}_k} e^{-i\frac{\alpha}{6}\hat{X}_j^4} e^{-i\frac{\alpha}{6}\hat{X}_k^4}, \quad (\text{A5})$$

which requires a decomposition for the terms of the form $e^{i\alpha\hat{X}_j^4}$. This can be achieved via the expression

$$e^{i\alpha\hat{X}_k^4} = e^{2i\hat{P}_j\hat{X}_k^2} e^{i\alpha\hat{X}_j^2} e^{-2i\hat{P}_j\hat{X}_k^2} e^{-i\alpha\hat{X}_j^2} e^{-2i\alpha\hat{X}_j\hat{X}_k^2}. \quad (\text{A6})$$

The total gate count for Eq. (A3) in terms of universal gates is 281, but it is exact. If we wish to express all operations in terms of \hat{X} operators only, we can use unitary conjugation with the Fourier transform gate. This will bring the total number of gates from the universal set to 373. On the other hand, the standard commutator approximation method requires about 28 gates, but for a precision of 10^{-3} will need 10^6 repetitions for a total of roughly 2.8×10^7 gates to decompose the original operation.

APPENDIX B: COARSE-GRAINING MEASUREMENT OUTPUTS

Here we consider the effect of postselecting on states other than $p = 0$ momentum eigenstates. Starting with the output

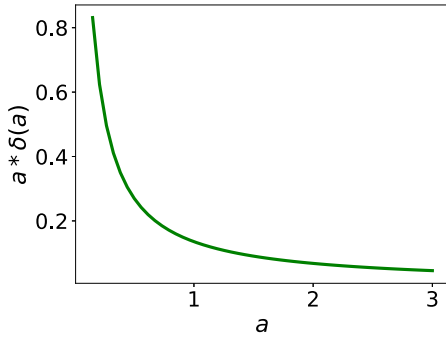


FIG. 7. Relative error $a\delta(a)$ due to coarse graining of measurement outputs. Here we have selected $p_1 = -p_2 = \Delta = 0.1$. As a approaches zero, the error becomes significant but decreases for larger a .

state as in Eq. (4) and performing a momentum homodyne measurement, projecting onto the general state $|p_1\rangle|p_2\rangle$ yields $(\mathbb{1} \otimes |p_1\rangle\langle p_2| \otimes |p_1\rangle\langle p_2|)|\Psi\rangle$

$$F(p_1, p_2, a) = e^{-p_1(p_1 - 2iap_2)/2a^2} \frac{\left[\sqrt{2}a e^{(p_1 - iap_2)^2/2a^2} - ip_1\sqrt{\pi} - p_1\sqrt{\pi} \operatorname{Erfi}\left(\frac{p_1 - iap_2}{\sqrt{2}a}\right) \right]}{\sqrt{2}a^2}, \tag{B4}$$

where $\operatorname{Erfi}(\cdot)$ is the imaginary error function. Outputs can be coarse grained by selecting a parameter Δp that when p_1 and p_2 are both in the interval $[-\Delta p, \Delta p]$ the algorithm has succeeded. Let $\delta(a) = \left| \frac{1}{a} - F(p_1, p_2, a) \right|$. We study the

$$\begin{aligned} &= \left(\frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx dy \Theta(x) y e^{-y^2/2} e^{-i\hat{A}xy} g(x, y) |f\rangle \right) |p_1\rangle |p_2\rangle \\ &= \left(\hat{A}_{\text{approx}}^{-1} |f\rangle \right) |p_1\rangle |p_2\rangle, \end{aligned} \tag{B1}$$

where

$$g(x, y) = e^{-ip_1x} e^{-ip_2y}. \tag{B2}$$

Expressing the inverse operator in terms of its action on the eigenstate $|a\rangle$

$$\begin{aligned} &\hat{A}_{\text{approx}}^{-1} |a\rangle \\ &= \frac{i}{\sqrt{2\pi}} \int_0^{\infty} dx \int_{-\infty}^{\infty} dy y e^{-y^2/2} e^{-i\hat{A}xy} g(x, y) |a\rangle. \end{aligned} \tag{B3}$$

Solving this integral gives the function $F(p_1, p_2, a)$, which deviates away from the desired function $1/a$ and is given by

error $\delta(a)$ for the worst case values of p_1 and p_2 , which occur when $p_1 = -p_2 = \Delta p$. Figure 7 shows the relative error $\delta(a)/(1/a) = a\delta(a)$ as a function of a for fixed Δp . As before, the error is largest for small a .

[1] P. W. Shor, in *Foundations of Computer Science, 1994 Proceedings, 35th Annual Symposium* (IEEE, New York, 1994), pp. 124–134.

[2] L. K. Grover, in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing* (ACM, Providence, RI, 1996), pp. 212–219.

[3] S. Lloyd, *Science* **273**, 1073 (1996).

[4] S. Hallgren, *J. ACM (JACM)* **54**, 4 (2007).

[5] M. H. Freedman, A. Kitaev, and Z. Wang, *Commun. Math. Phys.* **227**, 587 (2002).

[6] D. Aharonov, V. Jones, and Z. Landau, *Algorithmica* **55**, 395 (2009).

[7] A. M. Childs and W. van Dam, *Rev. Mod. Phys.* **82**, 1 (2010).

[8] F. G. Brandao and K. M. Svore, in *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium* (IEEE, New York, 2017), pp. 415–426.

[9] Fernando G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, X. Wu *et al.*, *Leibniz Int. Proc. Info.* (2019), <https://par.nsf.gov/biblio/10106371>.

[10] S. Lloyd, M. Mohseni, and P. Reberstrost, *Nat. Phys.* **10**, 631 (2014).

[11] P. Reberstrost, M. Mohseni, and S. Lloyd, *Phys. Rev. Lett.* **113**, 130503 (2014).

[12] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Nature (London)* **549**, 195 (2017).

[13] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, *Commun. Math. Phys.* **270**, 359 (2007).

[14] D. W. Berry, A. M. Childs, and R. Kothari, in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium* (IEEE, New York, 2015), pp. 792–809.

[15] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Forum of Mathematics, Sigma* (Cambridge University Press, Cambridge, UK, 2017), Vol. 5.

[16] D. W. Berry, *J. Phys. A: Math. Theor.* **47**, 105301 (2014).

[17] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, *Commun. Math. Phys.* **356**, 1057 (2017).

[18] T. Xin, S. Wei, J. Cui, J. Xiao, I. Arrazola, L. Lamata, X. Kong, D. Lu, E. Solano, and G. Long, [arXiv:1807.04553](https://arxiv.org/abs/1807.04553).

[19] S. K. Leyton and T. J. Osborne, [arXiv:0812.4423](https://arxiv.org/abs/0812.4423).

[20] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, *New J. Phys.* **15**, 013021 (2013).

[21] A. Montanaro and S. Pallister, *Phys. Rev. A* **93**, 032324 (2016).

[22] A. W. Harrow, A. Hassidim, and S. Lloyd, *Phys. Rev. Lett.* **103**, 150502 (2009).

[23] N. Wiebe, D. Braun, and S. Lloyd, *Phys. Rev. Lett.* **109**, 050505 (2012).

- [24] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, *Phys. Rev. Lett.* **110**, 250504 (2013).
- [25] A. Childs, R. Kothari, and R. Somma, *SIAM J. Comput.* **46**, 1920 (2017).
- [26] L. Wossnig, Z. Zhao, and A. Prakash, *Phys. Rev. Lett.* **120**, 050502 (2018).
- [27] S. Lloyd and S. L. Braunstein, *Phys. Rev. Lett.* **82**, 1784 (1999).
- [28] S. L. Braunstein and P. van Loock, *Rev. Mod. Phys.* **77**, 513 (2005).
- [29] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, [arXiv:1806.06871](https://arxiv.org/abs/1806.06871).
- [30] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, *Quantum Science and Technology* **4** (2018).
- [31] L. O’Driscoll, R. Nichols, and P. A. Knott, *Quantum Mach. Intel.* **1**, 5 (2019).
- [32] H.-K. Lau, R. Pooser, G. Siopsis, and C. Weedbrook, *Phys. Rev. Lett.* **118**, 080501 (2017).
- [33] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, *Quantum* **3**, 129 (2019).
- [34] H. F. Trotter, *Proc. Am. Math. Soc.* **10**, 545 (1959).
- [35] M. Suzuki, *Proc. Jpn. Acad., Ser. B* **69**, 161 (1993).
- [36] I. Dhand and B. C. Sanders, *J. Phys. A* **47**, 265206 (2014).
- [37] T. Kalajdziewski, C. Weedbrook, and P. Reberntrost, *Phys. Rev. A* **97**, 062311 (2018).
- [38] S. Sefi and P. van Loock, *Phys. Rev. Lett.* **107**, 170501 (2011).
- [39] T. Kalajdziewski and J. M. Arrazola, *Phys. Rev. A* **99**, 022341 (2019).
- [40] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods* (Springer Science & Business Media, New York, 2013), Vol. 22.
- [41] A. G. Werschulz, *The Computational Complexity of Differential and Integral Equations: An Information-based Approach* (Oxford University Press, Oxford, 1991).
- [42] K. Ritter and G. W. Wasilkowski, *Lect. Appl. Math.* **32**, 677 (1996).
- [43] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, [arXiv:1603.04467](https://arxiv.org/abs/1603.04467).
- [44] A. Gilyén, S. Lloyd, and E. Tang, [arXiv:1811.04909](https://arxiv.org/abs/1811.04909).
- [45] N.-H. Chia, H.-H. Lin, and C. Wang, [arXiv:1811.04852](https://arxiv.org/abs/1811.04852).