Variational quantum algorithm for the Poisson equation

Hai-Ling Liu,[1,2] Yu-Sen Wu,[1] Lin-Chun Wan,[1] Shi-Jie Pan,[1] Su-Juan Qin,[1,*] Fei Gao,[1,3,†] and Qiao-Yan Wen[1,‡]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*

[2]*State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China*

[3]*Center for Quantum Computing, Peng Cheng Laboratory, Shenzhen 518055, China*

(Received 4 March 2021; accepted 3 August 2021; published 18 August 2021)

The Poisson equation has wide applications in many areas of science and engineering. Although there are some quantum algorithms that can efficiently solve the Poisson equation, they generally require a fault-tolerant quantum computer, which is beyond the current technology. We propose a variational quantum algorithm (VQA) to solve the Poisson equation, which can be executed on noisy intermediate-scale quantum devices. In detail, we first adopt the finite-difference method to transform the Poisson equation into a linear system. Then, according to the special structure of the linear system, we find an explicit tensor product decomposition, with only $(2\log_2 n + 1)$ items, of its coefficient matrix under a specific set of simple operators, where n is the dimension of the coefficient matrix. This implies that the proposed VQA needs fewer quantum measurements, which dramatically reduces the required quantum resources. Additionally, we design observables to efficiently evaluate the expectation values of the simple operators on a quantum computer. Numerical experiments demonstrate that our algorithm can solve the Poisson equation.

DOI: 10.1103/PhysRevA.104.022418

I. INTRODUCTION

Quantum computing has been shown to be more computationally powerful than classical computing in solving certain problems, such as factoring large numbers [1], unstructured database searching [2], solving equations [3,4], classification [5,6], linear regression [7,8], and dimensionality reduction [9–11].

The Poisson equation has wide applications in many areas, such as quantum mechanical continuum solvation [12] and Markov chains [13,14]. In general, the finite-difference and spectral methods [15–17] are used to discretize the Poisson equation to obtain a linear system, the solution of which can approximate the solution of the Poisson equation. Since the dimension of the linear system is generally very large, solving the linear system is quite time consuming. To solve the Poisson equation efficiently, some related quantum algorithms [18–22] have been proposed. These quantum algorithms have shown significant speedups over their classical counterparts.

However, the advantages of quantum algorithms mentioned above usually rely on a fault-tolerant quantum computer, which may take a long time horizon to implement. Recent developments in quantum hardware have motivated advances in algorithms to run in the so-called noisy intermediate-scale quantum (NISQ) devices [23], which only support a shallow quantum circuit, a restricted number of physical qubits, and limited gate fidelity. An important question is how to solve some practical and meaningful tasks on such NISQ devices.

Variational quantum algorithms (VQAs) are a class of hybrid quantum-classical algorithms, which are expected to realize quantum advantages on NISQ devices [23–25]. Specifically, VQAs employ a shallow-depth quantum circuit to efficiently evaluate a cost function which depends on the parameters of a quantum gate sequence on the quantum computer, and the classical computer uses this cost information to adjust the parameters of the gate sequence to minimize the cost function. VQAs have been successfully applied to calculate the ground state or the excited state of the physical Hamiltonians [26–29], diagnose a quantum state [30], solve combinatorial optimization problems [31,32], process classification tasks [33], solve linear systems [34–36], etc.

Here, our goal is to design a VQA to solve the Poisson equation. A straightforward idea is to first adopt the finite-difference method to discretize the Poisson equation to obtain a linear system, and then solve the linear system directly by using the existing technique of solving the linear system with VQAs [34–36]. However, the algorithms proposed in Refs. [34–36] always require a strategy to decompose the coefficient matrix A into a sum of tensor products of some operators, that is, $A = \sum_{j=1}^{d} \alpha_j O_j$, where α_j is a constant coefficient and O_j is an operator. It is worth noting that d and O_j should meet the following two requirements [24,25]:

(i) R1: The number of terms $d = O[\text{poly}(\log_2 n)]$, where n is the dimension of the coefficient matrix.

(ii) R2: Each O_j is a tensor product of some simple operators which can be efficiently measured on a quantum computer.

In fact, finding a strategy that satisfies the above requirements R1 and R2 is a nontrivial problem. Typically, one can decompose the coefficient matrix under the Pauli basis, but the number of decomposed items usually grows polynomially with the dimension of the coefficient matrix, which does not

*qsujuan@bupt.edu.cn
†gaof@bupt.edu.cn
‡wqy@bupt.edu.cn

meet the requirement R1. Thus, it is not feasible to use this intuitive decomposition strategy to design a VQA that solves the Poisson equation.

In this work, according to the special structure of the linear system, we find an explicit tensor product decomposition of the coefficient matrix A under a set of simple operators $\{I, \sigma_+ = |0\rangle\langle 1|, \sigma_- = |1\rangle\langle 0|\}$. It is worth emphasizing that the number of decomposition terms is only $(2\log_2 n + 1)$, which means that the proposed VQA needs fewer quantum measurement terms than the decomposition under the Pauli basis. Furthermore, we construct observables to efficiently evaluate the expectation values of the simple operators on a quantum computer. The above two requirements for the coefficient matrix A are satisfied. As a result, we design a VQA to solve the Poisson equation. Finally, we conduct numerical experiments to simulate our algorithm on PROJECTQ [37], and experimental results show that our algorithm can solve the Poisson equation.

The remainder of the paper is organized as follows. In Sec. II, we adopt the finite-difference method to discretize the Poisson equation to obtain a linear system. In Sec. III, we propose a VQA for the Poisson equation. To show the feasibility of our algorithm, we conduct numerical experiments in Sec. IV. We analyze the performance of our algorithm in Sec. V. Finally, we present our conclusion and discussion in Sec. VI.

II. DISCRETIZE THE POISSON EQUATION

The d-dimensional Poisson equation with Dirichlet boundary conditions is defined as follows:

$$\begin{aligned} -\Delta\mu(x) &= f(x), x \in D, \\ \mu(x) &= 0, x \in \partial D, \end{aligned} \tag{1}$$

where Δ is the Laplace operator, $D = (0,1)^d$ is the domain of $\mu(x)$, ∂D represents the boundary of D, and $f : D \to R$ is a sufficiently smooth function [17]. Here, we adopt the finite-difference method to discretize the Poisson equation to obtain a linear system [15], and then we gain the approximate solution of the Poisson equation by solving the linear system. The linear system generated by the discretization of the one-dimensional Poisson equation is

$$A\mathbf{x} = \mathbf{b}, \tag{2}$$

where

$$A = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \in R^{n\times n}. \tag{3}$$

Here, A is a positive definite matrix and satisfies $A^\dagger = A$, n comes from evenly dividing (0,1) into $n+1$ parts during discretization, and $\mathbf{b}$ is the vector obtained by sampling the function $f(x)$ on the interior grid points [38]. Similarly, we can also obtain the coefficient matrix generated by the discretization of the d-dimensional Poisson equation with Dirichlet boundary conditions,

$$\begin{aligned} A^{(d)} = &\underbrace{A \otimes I \otimes \cdots \otimes I}_{d} + I \otimes A \otimes I \otimes \cdots \otimes I \\ &+ \cdots + I \otimes \cdots \otimes I \otimes A, \end{aligned} \tag{4}$$

where $I \in R^{n\times n}$ and $A^{(d)} \in R^{n^d \times n^d}$.

III. A VQA FOR THE POISSON EQUATION

To design a VQA to solve the Poisson equation, we assume that there is an efficient unitary operator U that can prepare a quantum state $|\mathbf{b}\rangle \propto \mathbf{b}$ (the vector). This assumption is the same as Refs. [3,34–36]. Following the idea in Refs. [34–36], we transform the problem of solving the linear system in Eq. (2) to find the ground state of a Hamiltonian,

$$H = A^\dagger(I - |\mathbf{b}\rangle\langle\mathbf{b}|)A. \tag{5}$$

Here, A is the coefficient matrix of the linear system in Eq. (2). It can be verified that the solution $|\mathbf{x}\rangle$ is the unique eigenstate corresponding to the minimum eigenvalue 0 of H. The detailed proof is shown in Appendix B.

To obtain the ground state $|\mathbf{x}\rangle$, we first define the cost function $E(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$, where trial state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle$, with unitary gate sequence $U(\boldsymbol{\theta}) = U_L(\theta_L)\cdots U_1(\theta_1), \boldsymbol{\theta} = (\theta_L, \ldots, \theta_1)$. The cost function $E(\boldsymbol{\theta})$ represents the expectation value of H under the state $|\psi(\boldsymbol{\theta})\rangle$. Next we apply the classical optimizer (e.g., gradient descent) to adjust the parameters $\boldsymbol{\theta}$ to minimize $E(\boldsymbol{\theta})$, that is,

$$\begin{aligned} \min_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) &= \min_{\boldsymbol{\theta}} \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle \\ &= \min_{\boldsymbol{\theta}} [\langle\psi(\boldsymbol{\theta})|A^2|\psi(\boldsymbol{\theta})\rangle - |\langle\mathbf{b}|A|\psi(\boldsymbol{\theta})\rangle|^2]. \end{aligned} \tag{6}$$

Once we obtain $\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}} E(\boldsymbol{\theta})$, the solution $|\mathbf{x}\rangle \approx |\psi(\boldsymbol{\theta}_{\text{opt}})\rangle$ can be produced by $U(\boldsymbol{\theta}_{\text{opt}})$. The structure of the entire algorithm is shown in Fig. 1.

Note that the output of VQA is a quantum state $|\mathbf{x}\rangle$ corresponding to the solution vector $\mathbf{x}$, and this kind of output $|\mathbf{x}\rangle$ is ubiquitous in previous works for solving a linear system [3,34–36,39]. The solution $|\mathbf{x}\rangle$ has several applications in some scenarios. For example, the solution $|\mathbf{x}\rangle$ of Harrow-Hassidim-Lloyd (HHL) [3] can be used to obtain an estimate of the expectation value $\langle\mathbf{x}|M|\mathbf{x}\rangle$ of the linear operator M. Moreover, the existing quantum algorithms for solving differential equations have obtained the quantum state of the solution $|\mathbf{x}\rangle$ [18,39]. They pointed out that the solution $|\mathbf{x}\rangle$ can be used to obtain the inner product $\langle\mathbf{x}|\mathbf{z}\rangle$ with the state $|\mathbf{z}\rangle$.

In order to design a VQA to solve the linear system in Eq. (2), we give a strategy to decompose A and A^2 satisfying the requirements R1 and R2 in Sec. I. In the following sections, for convenience, we first consider the coefficient matrix A generated by the discretization of the one-dimensional Poisson equation and assume that $n = 2^m$, where m is a positive integer.

A. An explicit decomposition of A and A^2

We will show the process of acquiring a sum of the tensor products of A and A^2 under a specific set of simple operators. According to the special structures of A and A^2, we first

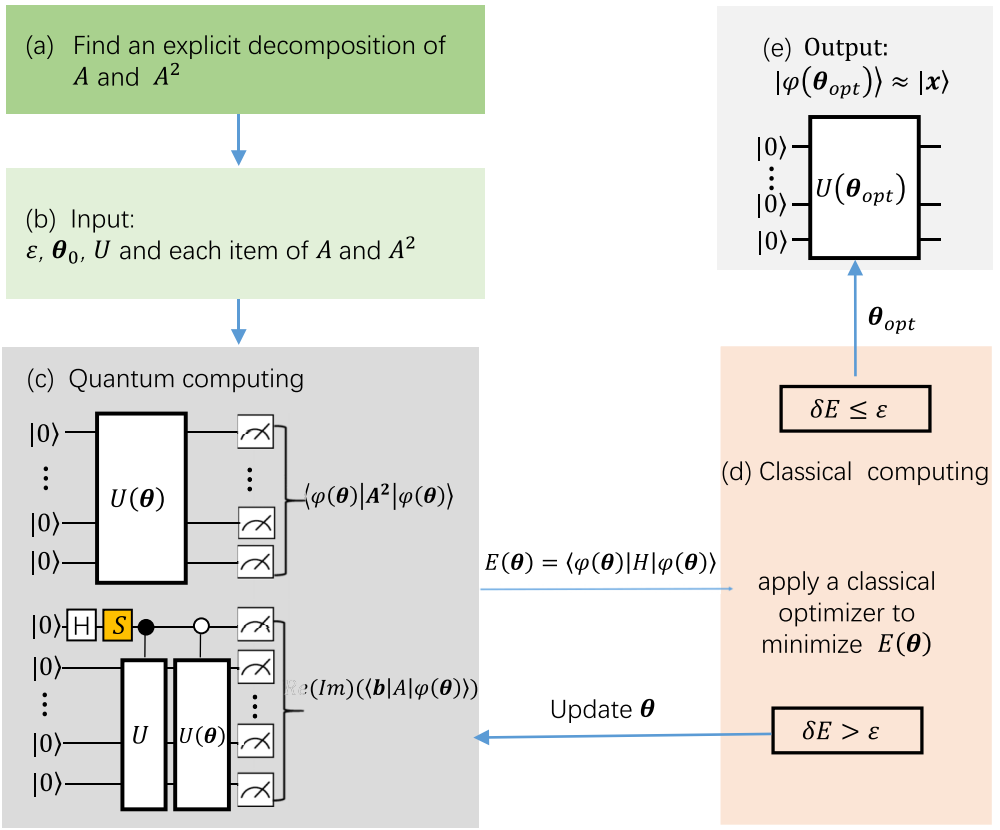

FIG. 1. Schematic diagram of the steps of the entire algorithm. (a) An explicit decomposition of A and A^2 under a specific set of simple operators is found. (b) The inputs of our algorithm are the precision ε, the initial parameters $\boldsymbol{\theta}_0$, the unitary operator U such that $U|\mathbf{0}\rangle = |\mathbf{b}\rangle$, and every item of a sum of tensor products of A and A^2. (c) To evaluate $E(\boldsymbol{\theta})$ on a quantum computer, we perform the unitary gate sequence $U(\boldsymbol{\theta})$ to evaluate the expected value of each decomposition term of A^2 under the state $|\psi(\boldsymbol{\theta})\rangle$ to gain $\langle\psi(\boldsymbol{\theta})|A^2|\psi(\boldsymbol{\theta})\rangle$. And we perform U and $U(\boldsymbol{\theta})$ to evaluate the value of each decomposition term of A under the state $|\psi(\boldsymbol{\theta})\rangle$ and $|\mathbf{b}\rangle$ to gain $|\langle\mathbf{b}|A|\psi(\boldsymbol{\theta})\rangle|^2$. The phase gate S is excluded when calculating the real part of $\langle\mathbf{b}|A|\psi(\boldsymbol{\theta})\rangle$ and included when calculating its imaginary part. (d) We apply a classical optimizer (e.g., gradient descent) to minimize $E(\boldsymbol{\theta})$. If $\delta E > \varepsilon$, where δE denotes the variation value of E, then update $\boldsymbol{\theta}$ to execute a new round of the quantum algorithm, otherwise output $\boldsymbol{\theta}_{\text{opt}} = \boldsymbol{\theta}$. (e) The output of VQA is a quantum state, $|\psi(\boldsymbol{\theta}_{\text{opt}}\rangle \approx |\mathbf{x}\rangle$, generated by $U(\boldsymbol{\theta}_{\text{opt}})$.

write A and A^2 into block matrices, respectively, then apply a recursive algorithm to find their explicit decomposition. For convenience, the matrices A and A^2 corresponding to m qubits are denoted as A_m and A_m^2, respectively.

Let us write A_m as a block matrix,

$$A_m = \left[\begin{array}{c|c} A_{m-1} & D_{m-1} \\ \hline D_{m-1}^{\mathrm{T}} & A_{m-1} \end{array}\right], \tag{7}$$

where

$$D_{m-1} = \begin{bmatrix} 0 & & 0 \\ & \ddots & \\ 0 & & \\ -1 & 0 & 0 \end{bmatrix}. \tag{8}$$

Then we provide a recursive decomposition strategy to find the explicit decomposition of A_m as follows:

$$\begin{aligned}
A_1 &= \left[\begin{array}{cc|cc} 2 & -1 & & \\ -1 & 2 & & \end{array}\right] = 2I - \sigma_+ - \sigma_-, \sigma_+ = |0\rangle\langle 1|, \sigma_- = |1\rangle\langle 0|; \\
A_2 &= \left[\begin{array}{cc|cc} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ \hline 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{array}\right] = I \otimes A_1 - \sigma_- \otimes \sigma_+ - \sigma_+ \otimes \sigma_-; \\
A_3 &= I \otimes A_2 - \sigma_- \otimes \sigma_+ \otimes \sigma_+ - \sigma_+ \otimes \sigma_- \otimes \sigma_- \\
&= I \otimes I \otimes (2I - \sigma_+ - \sigma_-) - I \otimes \sigma_- \otimes \sigma_+ \\
&\quad - I \otimes \sigma_+ \otimes \sigma_- - \sigma_- \otimes \sigma_+ \otimes \sigma_+ - \sigma_+ \otimes \sigma_- \otimes \sigma_-.
\end{aligned} \tag{9}$$

Then we have

$$
\begin{aligned}
A_m &= I \otimes A_{m-1} - \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-1} - \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-1} \\
&= \underbrace{I \otimes \cdots \otimes I}_{m-1} \otimes (2I - \sigma_+ - \sigma_-) \\
&\quad - \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_- \otimes \sigma_+ - \cdots - I \otimes \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-1} \\
&\quad - \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_+ \otimes \sigma_- - \cdots - I \otimes \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-2} \\
&\quad - \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-1} - \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-1}. \qquad (10)
\end{aligned}
$$

It is shown that A_m can be written as a linear combination of tensor products of simple operators $\{I, \sigma_+, \sigma_-\}$ and the total number of items of A_m is $2m + 1$, which is linear with respect to the logarithm of the dimension of the matrix. It means that our algorithm requires fewer quantum measurement terms than the decomposition under the Pauli basis, which will dramatically reduce the required quantum resources. Although the decomposition of A_m^2 can be obtained by A_m, the number of terms is $(2m + 1)^2$. In order to further reduce the number of decomposition items of A_m^2, next we use the similar method as A_m to show the decomposition process of A_m^2:

$$
\begin{aligned}
A_m^2 &= \begin{bmatrix}
5 & -4 & 1 & & & 0 \\
-4 & 6 & -4 & 1 & & \\
1 & \ddots & \ddots & \ddots & \ddots & \\
 & \ddots & \ddots & \ddots & \ddots & 1 \\
 & & 1 & -4 & 6 & -4 \\
0 & & & 1 & -4 & 5
\end{bmatrix} \\
&= \begin{bmatrix}
6 & -4 & 1 & & & 0 \\
-4 & 6 & -4 & 1 & & \\
1 & \ddots & \ddots & \ddots & \ddots & \\
 & \ddots & \ddots & \ddots & \ddots & 1 \\
 & & 1 & -4 & 6 & -4 \\
0 & & & 1 & -4 & 6
\end{bmatrix} - \begin{bmatrix}
1 & & & & & 0 \\
 & 0 & & & & \\
 & & \ddots & & & \\
 & & & 0 & & \\
 & & & & 0 & \\
0 & & & & & 1
\end{bmatrix} \\
&\equiv B_m - C_m. \qquad (11)
\end{aligned}
$$

According to Eq. (11), we only need to obtain the decomposition of B_m and C_m. We write B_m into a block matrix,

$$
B_m = \left[\begin{array}{c|c} B_{m-1} & M_{m-1} \\ \hline M_{m-1}^{\mathrm{T}} & B_{m-1} \end{array}\right], \qquad (12)
$$

where

$$
M_{m-1} = \begin{bmatrix}
0 & & & 0 \\
 & & & \\
0 & & \ddots & \\
1 & 0 & & \\
-4 & 1 & 0 & 0
\end{bmatrix}. \qquad (13)
$$

Next, we apply the recursive decomposition strategy to obtain the decomposition of B_m,

$$
\begin{aligned}
B_1 &= \left[\begin{array}{c|c} 6 & -4 \\ \hline -4 & 6 \end{array}\right] = 6I - 4\sigma_+ - 4\sigma_-; \\
B_2 &= \left[\begin{array}{cc|cc} 6 & -4 & 1 & 0 \\ -4 & 6 & -4 & 1 \\ \hline 1 & -4 & 6 & -4 \\ 0 & 1 & -4 & 6 \end{array}\right] \\
&= I \otimes B_1 + \sigma_- \otimes (I - 4\sigma_+) + \sigma_+ \otimes (I - 4\sigma_-); \\
B_3 &= I \otimes B_2 + \sigma_- \otimes \sigma_+ \otimes (I - 4\sigma_+) + \sigma_+ \otimes \sigma_- \otimes (I - 4\sigma_-)
\end{aligned}
$$

$$
\begin{aligned}
&= I \otimes I \otimes (6I - 4\sigma_+ - 4\sigma_-) \\
&\quad + I \otimes \sigma_- \otimes (I - 4\sigma_+) + \sigma_- \otimes \sigma_+ \otimes (I - 4\sigma_+) \\
&\quad + I \otimes \sigma_+ \otimes (I - 4\sigma_-) + \sigma_+ \otimes \sigma_- \otimes (I - 4\sigma_-).
\end{aligned} \tag{14}
$$

And then we can obtain

$$
\begin{aligned}
B_m &= I \otimes B_{m-1} + \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-2} \otimes (I - 4\sigma_+) \\
&\quad + \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-2} \otimes (I - 4\sigma_-) \\
&= \underbrace{I \otimes \cdots \otimes I}_{m-1} \otimes (6I - 4\sigma_+ - 4\sigma_-) \\
&\quad + \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_+ \otimes (I - 4\sigma_-) + \cdots \\
&\quad \cdots + I \otimes \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-3} \otimes (I - 4\sigma_-) \\
&\quad + \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_- \otimes (I - 4\sigma_+) + \cdots \\
&\quad \cdots + I \otimes \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-3} \otimes (I - 4\sigma_+) \\
&\quad + \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-2} \otimes (I - 4\sigma_+) \\
&\quad + \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-2} \otimes (I - 4\sigma_-).
\end{aligned} \tag{15}
$$

Thus, B_m can be expressed as a sum of tensor products of operators $\{I, \sigma_+, \sigma_-\}$ and the number of items is $4m - 1$.

Finally, we obtain the decomposition of C_m via the recursive decomposition strategy,

$$
\begin{aligned}
C_1 &= \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}\right] = \sigma_+\sigma_- + \sigma_-\sigma_+; \\
C_2 &= \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] = \sigma_+\sigma_- \otimes \sigma_+\sigma_- + \sigma_-\sigma_+ \otimes \sigma_-\sigma_+; \\
C_3 &= \sigma_+\sigma_- \otimes \sigma_+\sigma_- \otimes \sigma_+\sigma_- + \sigma_-\sigma_+ \otimes \sigma_-\sigma_+ \otimes \sigma_-\sigma_+.
\end{aligned} \tag{16}
$$

And we can obtain

$$
C_m = \underbrace{\sigma_+\sigma_- \otimes \cdots \otimes \sigma_+\sigma_-}_{m} + \underbrace{\sigma_-\sigma_+ \otimes \cdots \otimes \sigma_-\sigma_+}_{m}. \tag{17}
$$

It shows that C_m is presented in the form of a sum of the tensor product of $\{\sigma_+\sigma_- = |0\rangle\langle 0|, \sigma_-\sigma_+ = |1\rangle\langle 1|\}$. Thus the explicit decomposition form of A_m^2 can be obtained and the total number of terms is $4m + 1$.

In short, the decomposition strategy meets requirement R1. Next we will show that the decomposition strategy also satisfies requirement R2.

B. Evaluation of $E(\boldsymbol{\theta})$

To evaluate $E(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|(B - C)|\psi(\boldsymbol{\theta})\rangle - |\langle \mathbf{b}|A|\psi(\boldsymbol{\theta})\rangle|^2$, we need to evaluate $\langle\psi(\boldsymbol{\theta})|B|\psi(\boldsymbol{\theta})\rangle$, $|\langle \mathbf{b}|A|\psi(\boldsymbol{\theta})\rangle|^2$, and $\langle\psi(\boldsymbol{\theta})|C|\psi(\boldsymbol{\theta})\rangle$, respectively.

(1) Evaluation of $\langle\psi(\boldsymbol{\theta})|B|\psi(\boldsymbol{\theta})\rangle$ and $|\langle \mathbf{b}|A|\psi(\boldsymbol{\theta})\rangle|^2$.

Since the decomposition terms of A and B are not Hermitian operators, we need to design special observables to calculate

$$
\begin{aligned}
&\langle\psi(\boldsymbol{\theta})|\sigma_+ \otimes \sigma_- \otimes \cdots \otimes \sigma_-|\psi(\boldsymbol{\theta})\rangle, \\
&\langle\psi(\boldsymbol{\theta})|\sigma_- \otimes \sigma_+ \otimes \cdots \otimes \sigma_+|\psi(\boldsymbol{\theta})\rangle, \\
&|\langle \mathbf{b}|\sigma_+ \otimes \sigma_- \otimes \cdots \otimes \sigma_-|\psi(\boldsymbol{\theta})\rangle|^2, \\
&|\langle \mathbf{b}|\sigma_- \otimes \sigma_+ \otimes \cdots \otimes \sigma_+|\psi(\boldsymbol{\theta})\rangle|^2.
\end{aligned} \tag{18}
$$

First consider the simplest case, that is, to construct observables to obtain $\langle\psi(\boldsymbol{\theta})|\sigma_+|\psi(\boldsymbol{\theta})\rangle$, $\langle\psi(\boldsymbol{\theta})|\sigma_-|\psi(\boldsymbol{\theta})\rangle$, $|\langle \mathbf{b}|\sigma_+|\psi(\boldsymbol{\theta})\rangle|^2$, and $|\langle \mathbf{b}|\sigma_-|\psi(\boldsymbol{\theta})\rangle|^2$. Specifically, the

observables can be designed as follows:

$$\begin{aligned}\begin{bmatrix}\mathbf{0} & \sigma_+\\ \sigma_+^\dagger & \mathbf{0}\end{bmatrix} &= |\phi_{11}^+\rangle\langle\phi_{11}^+| - |\phi_{11}^-\rangle\langle\phi_{11}^-| \equiv O_{11},\\ \begin{bmatrix}\mathbf{0} & \sigma_-\\ \sigma_-^\dagger & \mathbf{0}\end{bmatrix} &= |\phi_{12}^+\rangle\langle\phi_{12}^+| - |\phi_{12}^-\rangle\langle\phi_{12}^-| \equiv O_{12},\end{aligned} \tag{19}$$

where $|\phi_{11}^\pm\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$ and $|\phi_{12}^\pm\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$ are Bell states. Then we can construct quantum states,

$$\begin{aligned}&|0,1\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\ |0,i1\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle),\\ &|\mathbf{b},\psi(\boldsymbol{\theta})\rangle \equiv \frac{1}{\sqrt{2}}[|0\rangle|\mathbf{b}\rangle + |1\rangle|\psi(\boldsymbol{\theta})\rangle],\\ &|\mathbf{b},i\psi(\boldsymbol{\theta})\rangle \equiv \frac{1}{\sqrt{2}}[|0\rangle|\mathbf{b}\rangle + i|1\rangle|\psi(\boldsymbol{\theta})\rangle].\end{aligned} \tag{20}$$

Note that

$$\begin{aligned}\langle\psi(\boldsymbol{\theta})|\sigma_+|\psi(\boldsymbol{\theta})\rangle &= \langle 0,1|\langle\psi(\boldsymbol{\theta})|O_{11}|0,1\rangle|\psi(\boldsymbol{\theta})\rangle\\ &\quad - i\langle 0,i1|\langle\psi(\boldsymbol{\theta})|O_{11}|0,i1\rangle|\psi(\boldsymbol{\theta})\rangle,\\ \langle\psi(\boldsymbol{\theta})|\sigma_-|\psi(\boldsymbol{\theta})\rangle &= \langle 0,1|\langle\psi(\boldsymbol{\theta})|O_{12}|0,1\rangle|\psi(\boldsymbol{\theta})\rangle\\ &\quad - i\langle 0,i1|\langle\psi(\boldsymbol{\theta})|O_{12}|0,i1\rangle|\psi(\boldsymbol{\theta})\rangle,\\ |\langle\mathbf{b}|\sigma_+|\psi(\boldsymbol{\theta})\rangle|^2 &= [\langle\mathbf{b},\psi(\boldsymbol{\theta})|O_{11}|\mathbf{b},\psi(\boldsymbol{\theta})\rangle]^2\\ &\quad + [\langle\mathbf{b},i\psi(\boldsymbol{\theta})|O_{11}|\mathbf{b},i\psi(\boldsymbol{\theta})\rangle]^2,\\ |\langle\mathbf{b}|\sigma_-|\psi(\boldsymbol{\theta})\rangle|^2 &= [\langle\mathbf{b},\psi(\boldsymbol{\theta})|O_{12}|\mathbf{b},\psi(\boldsymbol{\theta})\rangle]^2\\ &\quad + [\langle\mathbf{b},i\psi(\boldsymbol{\theta})|O_{12}|\mathbf{b},i\psi(\boldsymbol{\theta})\rangle]^2.\end{aligned} \tag{21}$$

Thus we can directly perform measurements in the Bell basis and then calculate the relevant probabilities to obtain $\langle\psi(\boldsymbol{\theta})|\sigma_+|\psi(\boldsymbol{\theta})\rangle$, $\langle\psi(\boldsymbol{\theta})|\sigma_-|\psi(\boldsymbol{\theta})\rangle$, $|\langle\mathbf{b}|\sigma_+|\psi(\boldsymbol{\theta})\rangle|^2$, and $|\langle\mathbf{b}|\sigma_-|\psi(\boldsymbol{\theta})\rangle|^2$. We mention that the method we used is similar to the Hadamard test [34–36,40,41].

For the case that the number of σ_+ and σ_- in each row of Eq. (18) is greater than 1, we show that observables can be designed in a similar way as the simplest case. The details can be found in Appendix C. By using Hadamard and CNOT gates, the measurements required to calculate the value of Eq. (18) can be transformed to the measurements in the computational basis. The quantum circuit is shown in Fig. 2. In short, we can evaluate each term of $|\langle\mathbf{b}|A|\psi(\boldsymbol{\theta})\rangle|^2$ and $\langle\psi(\boldsymbol{\theta})|B|\psi(\boldsymbol{\theta})\rangle$.

(2) Evaluation of $\langle\psi(\boldsymbol{\theta})|C|\psi(\boldsymbol{\theta})\rangle$.

Since $\sigma_-\sigma_+ = |1\rangle\langle 1|$ and $\sigma_+\sigma_- = |0\rangle\langle 0|$ are Hermitian operators, we can directly perform the measurements in the computational basis to obtain the expectation value of C under the state $|\psi(\boldsymbol{\theta})\rangle$. Due to the linearity property of operators, we can evaluate $E(\boldsymbol{\theta})$ efficiently on the quantum computer.

Herein, A and A^2 meet the requirement R2 in Sec. I. In short, we can design VQA to solve the one-dimensional Poisson equation with Dirichlet boundary conditions.

Similarly, we can obtain the decompositions of $A^{(d)}$ and $(A^{(d)})^2$ of the d-dimensional Poisson equation with the number of terms $d(2m+1)$ and $[d(2m+1)]^2$, respectively. When $d = O(\text{poly m})$, we can design VQA to solve the d-dimensional Poisson equation with Dirichlet boundary conditions.

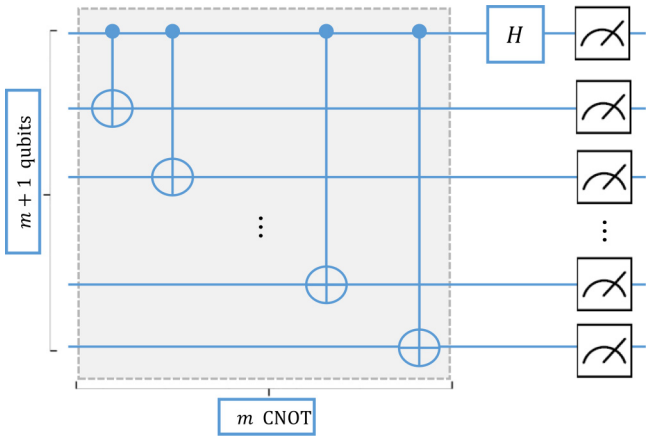

FIG. 2. The quantum circuit of the measurement required in the calculation of Eq. (18).

We extend our algorithm to the one-dimensional Poisson equation with the common boundary conditions of Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin [42–45]. However, for the d-dimensional Poisson equation, our algorithm cannot currently be extended to Neumann and Robin and the mixed boundary conditions of Dirichlet, Neumann, and Robin, except for the Dirichlet boundary conditions. See Appendix A for the specific analysis.

In addition, our algorithm can also solve the general tridiagonal and pentadiagonal Toeplitz systems, which are often utilized in solving partial differential equations [15–17,38]. See the detailed analysis in Appendix A.

IV. NUMERICAL EXPERIMENTS

We conduct numerical experiments to simulate our algorithm by using the PROJECTQ package [37], which is a high-performance simulator with emulation capabilities. In the experiments, the size of the coefficient matrix A obtained by the discretization of the one-dimensional Poisson equation with Dirichlet boundary conditions is $2^m \times 2^m$, with the number of qubits $m = 2, \ldots, 6$. The vector $\mathbf{b}$ is generated by sampling from the function $f(x) = x$ on the interior grid points.

Next we design a parameterized circuit $U(\boldsymbol{\theta})$ to generate a quantum state $|\psi(\boldsymbol{\theta})\rangle$ to approximate the solution $|\mathbf{x}\rangle$. In VQAs, the popular selections of $U(\boldsymbol{\theta})$ include the hardware efficient ansatz (HEA) [46], unitary coupled cluster (UCC) ansatz [47], quantum alternating operator ansatz (QAOA) [31,48], etc. Among them, the QAOA is known to be universal as the number of circuit layers tends to infinity [31,49], and the QAOA has obtained good results for several problems with a finite number of circuit layers [50,51]. One of the strengths of the QAOA is the fact that it reduces the size of the feasible solution space to obtain a better performance algorithm. Therefore, we choose the QAOA to construct $U(\boldsymbol{\theta})$ in this problem.

In general, the QAOA is composed of a driver Hamiltonian H_D and a mixer Hamiltonian H_M evolving the $|+\rangle^{\otimes n}$ state at

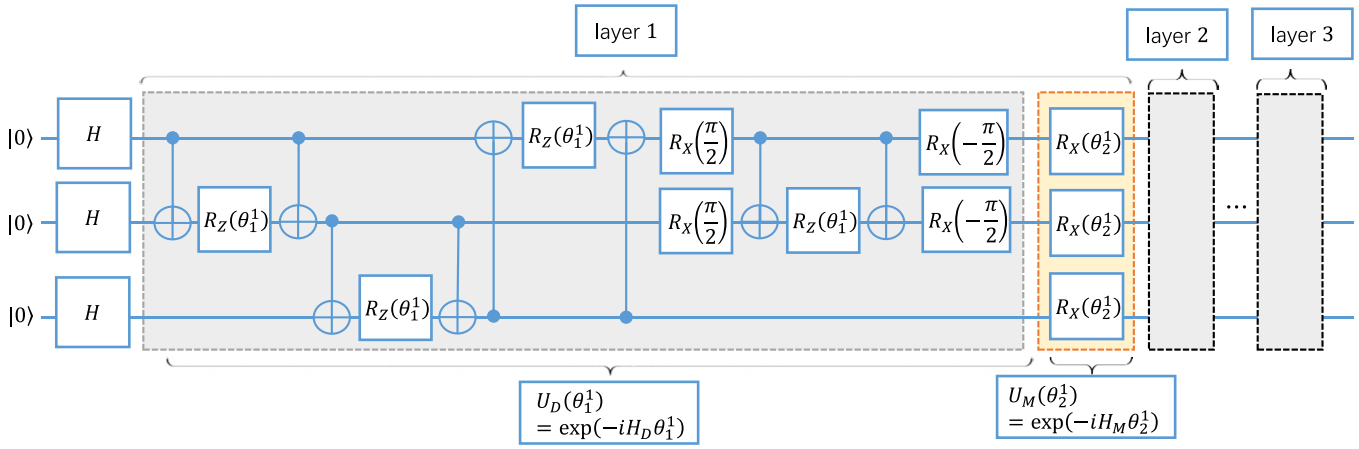

FIG. 3. A QAOA circuit with $m = 3$ qubits. R_X and R_Z are single-qubit rotation operators, respectively. The gray dashed box indicates the repeated block.

specified layers. The form of $U(\boldsymbol{\theta})$ can be written as follows:

$$U(\boldsymbol{\theta}) = \underbrace{U_M(\theta_l^L)U_D(\theta_{l-1}^L)}_{layerL} \cdots \underbrace{U_M(\theta_2^1)U_D(\theta_1^1)}_{layer1}, \tag{22}$$

where $U_D(\theta_p^q) = \exp(-iH_D\theta_p^q)$ and $U_M(\theta_p^q) = \exp(-iH_M\theta_p^q)$, with $p = 1, \ldots, l, q = 1, \ldots, L$.

In our experiments, $H_M = \sum_{j=0}^{m-1} X_j$, $H_D = \sum_{j=0}^{m-1} Z_jZ_{j+1} + Z_{m-1}Z_0 + Y_0Y_1$, $m = 2, \ldots, 6$, and each initial parameter $\theta_p^q \in [0, 2\pi)$ of $\boldsymbol{\theta}$ is chosen randomly, where X_j, Y_j, Z_j are Pauli operators. Specifically, the operators $U_D(\theta_p^q)$ and $U_M(\theta_p^q)$ can be decomposed into single- and double-qubit gates,

$$\begin{aligned}
\exp\left(-iZ_jZ_k\theta_p^q\right) =& \text{CNOT}(j,k)R_Z\left(k,\theta_p^q\right)\text{CNOT}(j,k),\\
\exp\left(-iY_jY_k\theta_p^q\right) =& R_X\left(j,\frac{\pi}{2}\right)R_X\left(k,\frac{\pi}{2}\right)\text{CNOT}(j,k)R_Z\left(k,\theta_p^q\right)\\
&\times \text{CNOT}(j,k)R_X\left(j,-\frac{\pi}{2}\right)R_X\left(k,-\frac{\pi}{2}\right),
\end{aligned} \tag{23}$$

where $\text{CNOT}(j,k)$ is controlled by the jth qubit and targeted on the kth qubit. For m-qubit circuits, the number of variable parameters required is $2L$ and the depth of single- and double-qubit gates required is $1 + (3m + 6)L$. The number of the variable parameters increases linearly with the increase of L, and the depth of the single- and double-qubit gates increases linearly with the increase of mL. Therefore, our algorithm can employ a relatively shallow quantum circuit to approximate $|\mathbf{x}\rangle$, which is expected to be implemented on a NISQ device.

According to Eqs. (22) and (23), we provide the QAOA quantum circuits with $m = 2, \ldots, 6$. In Fig. 3, we plot details of the QAOA circuit in the scenario of $m = 3$. In each iteration step, we perform 1000 quantum measurements for each term in the operators A and A^2. For example, to get $\langle\psi(\boldsymbol{\theta})|\sigma_+|\psi(\boldsymbol{\theta})\rangle$, we need to perform the measurements in Fig. 2 1000 times to calculate the correlation probabilities. According to the measurement results, each expectation value can be approximated and we can calculate the value of $E(\theta)$ in the current stage. For the classical optimizer, we adopt the Broyden-Fletcher-Goldfarb-Shanno (BFGS) to adjust the parameters $\boldsymbol{\theta}$. Figure 4 shows the experimental results of the proposed algorithm, that is, the fidelity $|\langle\mathbf{x}|\psi(\boldsymbol{\theta}_{\text{opt}})\rangle|$ gradually increases with the increasing of the layers of circuits. We also obtain the minimum number of layers that is required to guarantee the 0.99 fidelity, plotted as an inset.

V. ANALYSIS OF ALGORITHM PERFORMANCE

Here we make a detailed analysis of the performance of the algorithm, including the solution precision and the running time of the algorithm. We conclude that the lower the dimension of $|\mathbf{x}\rangle$, the lower the precision and the less running time of the algorithm. The specific analysis is as follows.

We first define three types of solutions:

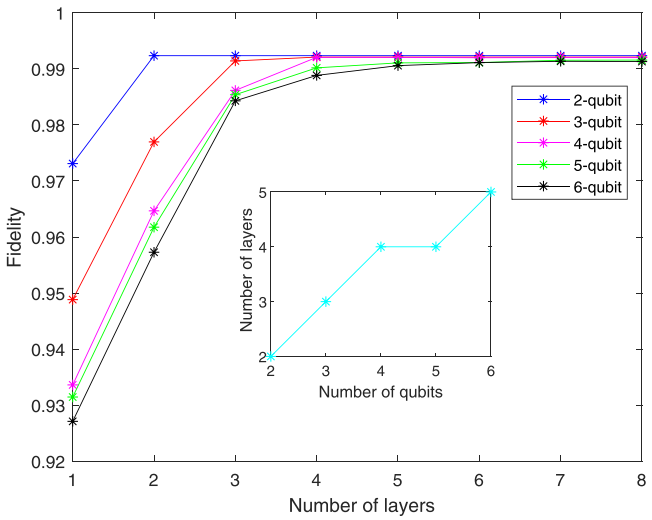

FIG. 4. Experimental results of our algorithm. The fidelity $|\langle\mathbf{x}|\psi(\boldsymbol{\theta}_{\text{opt}})\rangle|$ increases with the increase of layers and qubits. For a given number of qubits, the number of circuit layers is increases gradually until the fidelity reaches 0.99. The inset graph shows the minimum number of layers in the simulation when the fidelity reaches 0.99.

(a) Exact solution: the exact solution of the Poisson equation.

(b) Numerical solution: the exact solution of the linear system generated by the discretization of the Poisson equation.

(c) Quantum solution: the solution obtained by our quantum algorithm.

(1) Analysis of the solution precision.

The error ϵ_1 between the exact solution and the numerical solution has been analyzed by the classical algorithm [52], i.e., $\epsilon_1 = O(1/n^2)$, where n is the dimension of the solution $\mathbf{x}$. In our algorithm, the dimension n of $|\mathbf{x}\rangle$ grows exponentially with the increasing of m, i.e., $n = 2^m$. Thus the error $\epsilon_1 = O(\frac{1}{2^{2m}})$, that is, ϵ_1, decreases exponentially when the number of qubits m increases.

The error ϵ_2 between the numerical solution and the quantum solution is characterized by fidelity. Given a numerical solution, our quantum algorithm can provide a quantum solution with the fidelity of 0.99 by implementing 1000 measurements to each term. As the number of measurements increases, our algorithm will output a quantum solution with higher fidelity.

We define ϵ as the error between the exact solution and the quantum solution. Since $\epsilon \leqslant \epsilon_1 + \epsilon_2$ and $\epsilon_1 = O(\frac{1}{2^{2m}})$, ϵ is dominated by ϵ_2 when m is large. Our experimental results shown that our algorithm could output a quantum solution with high fidelity. Therefore, when m is large, our algorithm is expected to output a quantum solution that is closer to the exact solution.

(2) Analysis of algorithm running time.

In our algorithm, the depth of single- and double-qubit gates increases linearly with the increase of mL, which means that the quantum circuit can be implemented in polynomial time. Thus the running time to find the optimal parameters θ_{opt} is mainly determined by the iteration steps. For small-scale m, the classical optimization algorithm can converge to θ_{opt} quickly. When m takes a large value (50–100 qubits), the iteration steps of finding θ_{opt} cannot be accurately characterized, which is related to the initial parameters, classical optimization algorithms, and other factors. In addition, how to find θ_{opt} quickly on NISQ devices is still an open question. Nevertheless, with the development of quantum technology, it is expected to efficiently solve this optimization problem in the near future.

VI. CONCLUSION AND DISCUSSION

To summarize, we designed a VQA to solve the Poisson equation with Dirichlet boundary conditions. In particular, we found an explicit decomposition of the coefficient matrix of the linear system that approximates the Poisson equation. It is noteworthy that the number of decomposition items is only $2\log_2 n + 1$, where n is the dimension of the coefficient matrix, which greatly reduces the number of measurements in the VQAs. In addition, we construct observables to efficiently evaluate the expectation values of the simple operators on a quantum computer.

Our algorithm can be used to solve the one-dimensional Poisson equation with the common boundary conditions of Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin [42–45]. However, for the d-dimensional Poisson equation, except for the Dirichlet boundary conditions, our algorithm cannot currently be extended to the d-dimensional Poisson equation with Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin. How to design the corresponding VQA to solve the d-dimensional Poisson equation with other boundary conditions is still an interesting open problem. In addition, our algorithm can also solve the general tridiagonal and pentadiagonal Toeplitz systems, which are often utilized in solving partial differential equations [15–17,38]. Our algorithm is also expected to be extended to address the banded Toeplitz systems that have wide applications in many fields, such as Markov chains [53,54] and signal processing [55].

For the variational quantum algorithm applied in various fields, one often needs to find a strategy that decomposes the data matrix to meet the two requirements R1 and R2 mentioned in Sec. I. The most intuitive strategy that decomposes the data matrix into the Pauli operators naturally meets these two requirements for some problems in combinatorial optimization and quantum chemistry, but it is not always feasible for other problems such as a linear system, dimensionality reduction, and classification. Our algorithm indicates that the data matrix can be decomposed into a sum of tensor products of a class of simple operators with a small number of decomposition terms. And these simple operators may not be Pauli operators or even Hermitian operators, as long as a quantum circuit can be designed to efficiently evaluate the expected value of them. This idea may stimulate more VQAs for solving problems of practical interest.

Recently, the authors of Ref. [56] proposed a general framework for solving general nonlinear differential equations using differentiable quantum circuits on gate-based quantum hardware. But when using this framework to deal with Poisson equations, many challenges will be faced. For example, it is difficult to design an effective unitary operation to implement quantum feature mapping of high-dimensional variables. Designing VQAs based on other frameworks to solve differential equations is a goal worth considering.

ACKNOWLEDGMENTS

We thank J. Liu for useful discussions on the subject. This work is supported by the Fundamental Research Funds for the Central Universities (Grant No. 2019XD-A01) and NSFC (Grants No. 61976024 and No. 61972048).

APPENDIX A: APPLICATION OF ALGORITHM

In this Appendix, we extend our algorithm to the one-dimensional Poisson equation with the common boundary conditions of Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin [42–45]. However, for the d-dimensional Poisson equation, our algorithm cannot currently be extended to Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin, except for the Dirichlet boundary conditions. In addition, our algorithm can also

solve the general tridiagonal and pentadiagonal Toeplitz systems, which often appear in solving partial differential equations [15–17,38]. The specific analysis is as follows.

(1) The one-dimensional Poisson equation:

For convenience, the boundary conditions can be written uniformly as

$$\alpha_1\mu'(0)-\alpha_2\mu(0)=0,\quad \beta_1\mu'(1)+\beta_2\mu(1)=0, \tag{A1}$$

where $\alpha_1,\alpha_2,\beta_1,\beta_2$ are all positive constants. The types of boundary conditions are given by the nature of the quadruplet $(\alpha_1,\alpha_2,\beta_1,\beta_2)$. For example, $(\alpha_1=0,\alpha_2=1,\beta_1=0,\beta_2=1)$, $(\alpha_1=1,\alpha_2=0,\beta_1=1,\beta_2=0)$, and $(\alpha_1=1,\alpha_2=1,\beta_1=1,\beta_2=1)$ correspond to the Dirichlet, Neumann, and Robin boundary conditions, respectively. We can also consider mixing the Dirichlet, Neumann, and Robin boundary conditions, such as $(\alpha_1=0,\alpha_2=1,\beta_1=1,\beta_2=0)$ and $(\alpha_1=0,\alpha_2=1,\beta_1=1,\beta_2=1)$, which correspond to the Dirichlet-Neumann and Dirichlet-Robin boundary conditions, respectively.

Under the above unified boundary conditions, the coefficient matrix $\tilde{A}$ obtained by finite-difference discretization is defined as

$$\tilde{A}=\begin{bmatrix} 2-c & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 2-d & \end{bmatrix}=A-\begin{bmatrix} c & & & & & & 0 \\ & 0 & & & & & \\ & & \ddots & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ 0 & & & & & & d \end{bmatrix}\equiv A-\tilde{C}, \tag{A2}$$

where $c=\frac{\alpha_1}{\alpha_1+\alpha_2 h}$, $d=\frac{\beta_1}{\beta_1+\beta_2 h}$, $h=1/(n+1)$, with n derived from evenly dividing (0,1) into $n+1$ parts during discretization. $\tilde{A}^2$ can be calculated as follows:

$$\tilde{A}^2=B+\begin{bmatrix} 0 & c & & & & \\ c & 0 & 0 & & & \\ & 0 & 0 & 0 & & \\ & & \ddots & \ddots & & \\ & & & 0 & 0 & d \\ & & & & d & 0 \end{bmatrix}-\begin{bmatrix} 4c+1-c^2 & & & & & 0 \\ & 0 & & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & & & & 0 & \\ 0 & & & & & 4d+1-d^2 \end{bmatrix}\equiv B+D-E. \tag{A3}$$

We adopt the similar method as A to find the decompositions of $\tilde{C}$, D, and E as follows:

$$\begin{aligned} \tilde{C}&=c\times(\sigma_+\sigma_-\otimes\cdots\otimes\sigma_+\sigma_-)+d\times(\sigma_-\sigma_+\otimes\cdots\otimes\sigma_-\sigma_+),\\ D&=c\times(\sigma_+\sigma_-\otimes\cdots\otimes\sigma_+\sigma_-\otimes X)+d\times(\sigma_-\sigma_+\otimes\cdots\otimes\sigma_-\sigma_+\otimes X),\\ E&=(4c+1-c^2)\times(\sigma_+\sigma_-\otimes\cdots\otimes\sigma_+\sigma_-)+(4d+1-d^2)\times(\sigma_-\sigma_+\otimes\cdots\otimes\sigma_-\sigma_+). \end{aligned} \tag{A4}$$

The number of decomposition terms of $\tilde{A}$ and $\tilde{A}^2$ is $2\log_2 n+3$ and $4\log_2 n+3$, respectively. Therefore, our algorithm can also be used to solve the one-dimensional Poisson equation satisfying the common boundary conditions of Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin.

(2) The d-dimensional Poisson equation:

Except for the Dirichlet boundary conditions, the coefficient matrices obtained by discretization with the common boundary conditions of Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin, are complex. It is difficult for our method to find an effective decomposition that satisfies the requirements R1 and R2 (see Sec. I) of the variational quantum algorithm to solve a linear system. Thus our algorithm cannot currently be extended to the d-dimensional Poisson equation with Neumann and Robin, and the mixed boundary conditions of Dirichlet, Neumann, and Robin. How to design the corresponding variational quantum algorithm to solve the d-dimensional Poisson equation with other boundary conditions is still an interesting open problem.

(3) The general tridiagonal and pentadiagonal Toeplitz matrices:

The general tridiagonal and pentadiagonal Toeplitz matrices are defined as

$$W=\begin{bmatrix} t_0 & t_1 & & 0 \\ t_{-1} & \ddots & \ddots & \\ & \ddots & \ddots & t_1 \\ 0 & & t_{-1} & t_0 \end{bmatrix}\in R^{n\times n},\quad V=\begin{bmatrix} t_0 & t_1 & t_2 & & & 0 \\ t_{-1} & t_0 & t_1 & t_2 & & \\ t_{-2} & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & t_2 \\ & & t_{-2} & t_{-1} & t_0 & t_1 \\ 0 & & & t_{-2} & t_{-1} & t_0 \end{bmatrix}\in R^{n\times n}, \tag{A5}$$

where $t_i \in R$, $i = 0, \pm1, \pm2$. The explicit decompositions of W and V can be expressed as

$$\begin{aligned}
W =& \underbrace{I \otimes \cdots \otimes I}_{m-1} \otimes (t_0 I + t_{-1}\sigma_+ + t_1\sigma_-) + \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_- \otimes (t_1\sigma_+) + \cdots + I \otimes \sigma_- \otimes \underbrace{\sigma_+ \cdots \otimes \sigma_+}_{m-3} \otimes (t_1\sigma_+) \\
&+ \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_+ \otimes (t_{-1}\sigma_-) + \cdots + I \otimes \sigma_+ \otimes \underbrace{\sigma_- \cdots \otimes \sigma_-}_{m-3} \otimes (t_{-1}\sigma_-) \\
&+ \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-2} \otimes (t_1\sigma_+) + \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-2} \otimes (t_{-1}\sigma_-), \\
V =& \underbrace{I \otimes \cdots \otimes I}_{m-1} \otimes (t_0 I + t_{-1}\sigma_+ + t_1\sigma_-) + \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_- \otimes (t_2 I + t_1\sigma_+) + \cdots + I \otimes \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-3} \otimes (t_2 I + t_1\sigma_+) \\
&+ \underbrace{I \otimes \cdots \otimes I}_{m-2} \otimes \sigma_+ \otimes (t_{-2} I + t_{-1}\sigma_-) + \cdots + I \otimes \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-3} \otimes (t_{-2} I + t_{-1}\sigma_-) \\
&+ \sigma_- \otimes \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{m-2} \otimes (t_2 I + t_1\sigma_+) + \sigma_+ \otimes \underbrace{\sigma_- \otimes \cdots \otimes \sigma_-}_{m-2} \otimes (t_{-2} I + t_{-1}\sigma_-).
\end{aligned} \tag{A6}$$

The total number of decomposed items of W and V are $2\log_2 n + 1$ and $4\log_2 n - 1$, respectively. Similarly, we can obtain the decompositions of W^2 and V^2 with the number of terms $(2\log_2 n + 1)^2$ and $(4\log_2 n - 1)^2$. Therefore, our algorithm can also be used to solve the general tridiagonal and pentadiagonal Toeplitz systems.

APPENDIX B: PROOF THAT $|\mathbf{x}\rangle$ IS THE UNIQUE EIGENSTATE CORRESPONDING TO THE MINIMUM EIGENVALUE 0 OF H

We prove in detail that the minimum eigenvalue of H is 0, and its corresponding unique eigenstate is $|\mathbf{x}\rangle = A^{-1}|\mathbf{b}\rangle / \|A^{-1}|\mathbf{b}\rangle\|$. The proof process, which is similar to Refs. [34–36,57], is as follows:

(1) Let $G = (I - |\mathbf{b}\rangle\langle\mathbf{b}|)A$. Then we have

$$G^\dagger G = A^\dagger (I - |\mathbf{b}\rangle\langle\mathbf{b}|)(I - |\mathbf{b}\rangle\langle\mathbf{b}|)A = A^\dagger (I - |\mathbf{b}\rangle\langle\mathbf{b}|)A = H. \tag{B1}$$

It implies that H is a semidefinite matrix.

(2) Note that

$$H|\mathbf{x}\rangle = A^\dagger (I - |\mathbf{b}\rangle\langle\mathbf{b}|)A \frac{A^{-1}|\mathbf{b}\rangle}{\|A^{-1}|\mathbf{b}\rangle\|} = \frac{1}{\|A^{-1}|\mathbf{b}\rangle\|} A^\dagger (|\mathbf{b}\rangle - |\mathbf{b}\rangle) = 0|\mathbf{x}\rangle. \tag{B2}$$

Herein, the minimum eigenvalue of H is 0, and its corresponding eigenstate is $|\mathbf{x}\rangle$.

(3) Assume that $|\mathbf{y}\rangle = A^{-1}|\mathbf{c}\rangle / \|A^{-1}|\mathbf{c}\rangle\|$ is another eigenstate corresponding to the 0 eigenvalue of H. Then,

$$H|\mathbf{y}\rangle = A^\dagger (I - |\mathbf{b}\rangle\langle\mathbf{b}|)A \frac{A^{-1}|\mathbf{c}\rangle}{\|A^{-1}|\mathbf{c}\rangle\|} = \frac{1}{\|A^{-1}|\mathbf{c}\rangle\|} A^\dagger (I - |\mathbf{b}\rangle\langle\mathbf{b}|)|\mathbf{c}\rangle = 0|\mathbf{y}\rangle. \tag{B3}$$

Since A is invertible and the only eigenstate of the zero eigenvalue of $I - |\mathbf{b}\rangle\langle\mathbf{b}|$ is $|\mathbf{b}\rangle$, we have $|\mathbf{c}\rangle = |\mathbf{b}\rangle$, which means $|\mathbf{y}\rangle = |\mathbf{x}\rangle$.

In short, $|\mathbf{x}\rangle$ is the unique eigenstate corresponding to the minimum eigenvalue 0 of H.

APPENDIX C: EVALUATION OF THE CORRESPONDING VALUE OF EQ. (18)

For the case where the number of σ_+ and σ_- in each row of Eq. (18) is greater than 1, we show that observables can be designed in a similar way as the simplest case.

To obtain $\langle\psi(\boldsymbol{\theta})|\sigma_+ \otimes \sigma_-|\psi(\boldsymbol{\theta})\rangle$, $\langle\psi(\boldsymbol{\theta})|\sigma_- \otimes \sigma_+|\psi(\boldsymbol{\theta})\rangle$, $|\langle\mathbf{b}|\sigma_+ \otimes \sigma_-|\psi(\boldsymbol{\theta})\rangle|^2$, and $|\langle\mathbf{b}|\sigma_- \otimes \sigma_+|\psi(\boldsymbol{\theta})\rangle|^2$, observables can be designed as follows:

$$\begin{aligned}
&\begin{bmatrix} \mathbf{0} & \sigma_+ \otimes \sigma_+ \\ (\sigma_+ \otimes \sigma_+)^\dagger & \mathbf{0} \end{bmatrix} = |\phi_{21}^+\rangle\langle\phi_{21}^+| - |\phi_{21}^-\rangle\langle\phi_{21}^-| \equiv O_{21},\ |\phi_{21}^\pm\rangle = \frac{1}{\sqrt{2}}(|000\rangle \pm |111\rangle), \\
&\begin{bmatrix} \mathbf{0} & \sigma_+ \otimes \sigma_- \\ (\sigma_+ \otimes \sigma_-)^\dagger & \mathbf{0} \end{bmatrix} = |\phi_{22}^+\rangle\langle\phi_{22}^+| - |\phi_{22}^-\rangle\langle\phi_{22}^-| \equiv O_{22},\ |\phi_{22}^\pm\rangle = \frac{1}{\sqrt{2}}(|001\rangle \pm |110\rangle), \\
&\begin{bmatrix} \mathbf{0} & \sigma_- \otimes \sigma_+ \\ (\sigma_- \otimes \sigma_+)^\dagger & \mathbf{0} \end{bmatrix} = |\phi_{23}^+\rangle\langle\phi_{23}^+| - |\phi_{23}^-\rangle\langle\phi_{23}^-| \equiv O_{23},\ |\phi_{23}^\pm\rangle = \frac{1}{\sqrt{2}}(|010\rangle \pm |101\rangle), \\
&\begin{bmatrix} \mathbf{0} & \sigma_- \otimes \sigma_- \\ (\sigma_- \otimes \sigma_-)^\dagger & \mathbf{0} \end{bmatrix} = |\phi_{24}^+\rangle\langle\phi_{24}^+| - |\phi_{24}^-\rangle\langle\phi_{24}^-| \equiv O_{24},\ |\phi_{24}^\pm\rangle = \frac{1}{\sqrt{2}}(|011\rangle \pm |100\rangle).
\end{aligned} \tag{C1}$$

Then we also need to construct the quantum states of Eq. (20). Note that

$$\begin{aligned}
\langle\psi(\boldsymbol{\theta})|\sigma_+\otimes\sigma_-|\psi(\boldsymbol{\theta})\rangle &= \langle 0,1|\langle\psi(\boldsymbol{\theta})|O_{22}|0,1\rangle|\psi(\boldsymbol{\theta})\rangle - i\langle 0,i1|\langle\psi(\boldsymbol{\theta})|O_{22}|0,i1\rangle|\psi(\boldsymbol{\theta})\rangle,\\
\langle\psi(\boldsymbol{\theta})|\sigma_-\otimes\sigma_+|\psi(\boldsymbol{\theta})\rangle &= \langle 0,1|\langle\psi(\boldsymbol{\theta})|O_{23}|0,1\rangle|\psi(\boldsymbol{\theta})\rangle - i\langle 0,i1|\langle\psi(\boldsymbol{\theta})|O_{23}|0,i1\rangle|\psi(\boldsymbol{\theta})\rangle,\\
|\langle\mathbf{b}|\sigma_+\otimes\sigma_-|\psi(\boldsymbol{\theta})\rangle|^2 &= [\langle\mathbf{b},\psi(\boldsymbol{\theta})|O_{22}|\mathbf{b},\psi(\boldsymbol{\theta})\rangle]^2 + [\langle\mathbf{b},i\psi(\boldsymbol{\theta})|O_{22}|\mathbf{b},i\psi(\boldsymbol{\theta})\rangle]^2,\\
|\langle\mathbf{b}|\sigma_-\otimes\sigma_+|\psi(\boldsymbol{\theta})\rangle|^2 &= [\langle\mathbf{b},\psi(\boldsymbol{\theta})|O_{23}|\mathbf{b},\psi(\boldsymbol{\theta})\rangle]^2 + [\langle\mathbf{b},i\psi(\boldsymbol{\theta})|O_{23}|\mathbf{b},i\psi(\boldsymbol{\theta})\rangle]^2.
\end{aligned} \tag{C2}$$

Thus, $\langle\psi(\boldsymbol{\theta})|\sigma_+\otimes\sigma_-|\psi(\boldsymbol{\theta})\rangle$, $\langle\psi(\boldsymbol{\theta})|\sigma_-\otimes\sigma_+|\psi(\boldsymbol{\theta})\rangle$, $|\langle\mathbf{b}|\sigma_+\otimes\sigma_-|\psi(\boldsymbol{\theta})\rangle|^2$, and $|\langle\mathbf{b}|\sigma_-\otimes\sigma_+|\psi(\boldsymbol{\theta})\rangle|^2$ can be obtained by directly performing measurements in the $|\phi_{2j}^{\pm}\rangle$ basis and calculating the relevant probabilities, $j=1,\ldots,4$. The required measurements can be transformed to the measurements in the computational basis, which is shown in Fig. 2.

Similarly, we can also design observables to obtain $\langle\psi(\boldsymbol{\theta})|\sigma_+\otimes\sigma_-\otimes\cdots\otimes\sigma_-|\psi(\boldsymbol{\theta})\rangle$, $\langle\psi(\boldsymbol{\theta})|\sigma_-\otimes\sigma_+\otimes\cdots\otimes\sigma_+|\psi(\boldsymbol{\theta})\rangle$, $|\langle\mathbf{b}|\sigma_+\otimes\sigma_-\otimes\cdots\otimes\sigma_-|\psi(\boldsymbol{\theta})\rangle|^2$, and $|\langle\mathbf{b}|\sigma_-\otimes\sigma_+\otimes\cdots\otimes\sigma_+|\psi(\boldsymbol{\theta})\rangle|^2$. For example, the observables corresponding to $\sigma_+\otimes\underbrace{\sigma_-\otimes\cdots\otimes\sigma_-}_{m-1}$ and $\sigma_-\otimes\underbrace{\sigma_+\otimes\cdots\otimes\sigma_+}_{m-1}$ are $|\phi_{m2}^+\rangle\langle\phi_{m2}^+| - |\phi_{m2}^-\rangle\langle\phi_{m2}^-|$, $|\phi_{m2}^{\pm}\rangle = \frac{1}{\sqrt{2}}(|00\underbrace{1\cdots1}_{m-1}\rangle \pm |11\underbrace{0\cdots0}_{m-1}\rangle)$ and $|\phi_{m3}^+\rangle\langle\phi_{m3}^+| - |\phi_{m3}^-\rangle\langle\phi_{m3}^-|$, $|\phi_{m3}^{\pm}\rangle = \frac{1}{\sqrt{2}}(|01\underbrace{1\cdots1}_{m-1}\rangle \pm |01\underbrace{0\cdots0}_{m-1}\rangle)$, respectively.

[1] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 1994), pp. 124–134.

[2] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Phys. Rev. Lett. **79**, 325 (1997).

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[4] L.-C. Wan, C.-H. Yu, S.-J. Pan, F. Gao, Q.-Y. Wen, and S.-J. Qin, Asymptotic quantum algorithm for the Toeplitz systems, Phys. Rev. A **97**, 062322 (2018).

[5] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).

[6] B. Duan, J. Yuan, Y. Liu, and D. Li, Quantum algorithm for support matrix machines, Phys. Rev. A **96**, 032301 (2017).

[7] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[8] C.-H. Yu, F. Gao, and Q. Wen, An improved quantum algorithm for ridge regression, IEEE Trans. Know. Data Eng. **33**, 858 (2021).

[9] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[10] I. Cong and L. Duan, Quantum discriminant analysis for dimensionality reduction and classification, New J. Phys. **18**, 073011 (2016).

[11] S.-J. Pan, L.-C. Wan, H.-L. Liu, Q.-L. Wang, S.-J. Qin, Q.-Y. Wen, and F. Gao, Improved quantum algorithm for A-optimal projection, Phys. Rev. A **102**, 052402 (2020).

[12] J. Tomasi, B. Mennucci, and R. Cammi, Quantum mechanical continuum solvation models, Chem. Rev. **105**, 2999 (2005).

[13] S. Meyn, *Control Techniques for Complex Networks* (Cambridge University Press, Cambridge, 2008).

[14] S. Asmussen and P. W. Glynn, *Stochastic Simulation: Algorithms and Analysis* (Springer Science & Business Media, New York, 2007), Vol. 57.

[15] G. E. Forsythe, W. R. Wasow, and W. Nachbar, Finite-difference methods for partial differential equations, Phys. Today **14**(4), 58 (1961).

[16] C. I. Gheorghiu, *Spectral Methods for Differential Problems* (Casa Cărţii de Ştiinţă Cluj-Napoca, Cluj-Napoca, Romania, 2007).

[17] G. B. Folland, *Introduction to Partial Differential Equations* (Princeton University Press, Princeton, NJ, 1995).

[18] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, Quantum algorithm and circuit design solving the Poisson equation, New J. Phys. **15**, 013021 (2013).

[19] A. M. Childs, J.-P. Liu, and A. Ostrander, High-recision quantum algorithms for partial differential equations, arXiv:2002.07868.

[20] A. C. Vazquez, R. Hiptmair, and S. Woerner, Enhancing the quantum linear systems algorithm using Richardson extrapolation, arXiv:2009.04484.

[21] 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).

[22] S. Lloyd, G. De Palma, C. Gokler, B. Kiani, Z.-W. Liu, M. Marvian, F. Tennie, and T. Palmer, Quantum algorithm for nonlinear differential equations, arXiv:2011.06571.

[23] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[24] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

[25] M. Cerezo, A. Arrasmith, R. Babbush *et al.*, Variational quantum algorithm, arXiv:2012.09265.

[26] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[27] K. M. Nakanishi, K. Mitarai, and K. Fujii, Subspace-search variational quantum eigensolver for excited states, Phys. Rev. Research **1**, 033062 (2019).

[28] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, Variational quantum algorithms for discovering Hamiltonian spectra, Phys. Rev. A **99**, 062304 (2019).

[29] O. Higgott, D. Wang, and S. Brierley, Variational quantum computation of excited states, Quantum **3**, 156 (2019).

[30] R. LaRose, A. Tikku, É. O' Neel-Judy, L. Cincio, and P. J. Coles, Variational quantum state diagonalization, npj Quantum Inf. **5**, 57 (2019).

[31] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[32] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, arXiv:1602.07674.

[33] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).

[34] X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational algorithms for linear algebra, arXiv:1909.03898.

[35] H.-Y. Huang, K. Bharti, and P. Rebentrost, Near-term quantum algorithms for linear systems of equations, arXiv:1909.07344.

[36] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. Coles, Variational quantum linear solver: A hybrid algorithm for linear systems, Bull. Am. Phys. Soc. **65** (2020).

[37] D. S. Steiger, T. Häner, and M. Troyer, ProjectQ: An open source software framework for quantum computing, Quantum **2**, 49 (2018).

[38] J. W. Demmel, *Applied Numerical Linear Algebra* (SIAM, Philadelphia, PA, 1997).

[39] A. Montanaro and S. Pallister, Quantum algorithms and the finite element method, Phys. Rev. A **93**, 032324 (2016).

[40] D. Aharonov, V. Jones, and Z. Landau, A polynomial quantum algorithm for approximating the Jones polynomial, Algorithmica **55**, 395 (2009).

[41] B. Wu, M. Ray, L. Zhao, X. Sun, and P. Rebentrost, Quantum-classical algorithms for skewed linear systems with an optimized Hadamard test, Phys. Rev. A **103**, 042422 (2021).

[42] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-dependent Problems* (SIAM, Philadelphia, PA, 2007).

[43] C. Weitzel and A. Harsy, *Eigenvalues and Eigenvectors for General Boundary Conditions* (Citeseer, 2007).

[44] S. B. Gueye, K. Talla, C. Mbow *et al.*, Solution of 1d poisson equation with neumann-dirichlet and dirichlet-neumann boundary conditions, using the finite difference method, J. Electromag. Anal. Applic. **6**, 309 (2014).

[45] S. B. Gueye, K. Talla, C. Mbow *et al.*, Generalization of the exact solution of 1D Poisson equation with Robin boundary conditions, using the finite difference method, J. Electromag. Anal. Applic. **6**, 372 (2014).

[46] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[47] A. G. Taube and R. J. Bartlett, New perspectives on unitary coupled-cluster theory, Intl. J. Quantum Chem. **106**, 3393 (2006).

[48] S. Hadfield, Z. Wang, B. O' Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[49] S. Lloyd, Quantum approximate optimization is computationally universal, arXiv:1812.11075.

[50] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv:1811.08419.

[51] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices, Phys. Rev. X **10**, 021067 (2020).

[52] G. Yoon and C. Min, Analyses on the finite difference method by Gibou et al. for Poisson equation, J. Comput. Phys. **280**, 184 (2015).

[53] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach* (Courier Corporation, Massachusetts, 1994).

[54] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, 1st ed. (CRC Press, Boca Raton, FL, 1989).

[55] M. K. Ng and R. H. Chan, Scientific applications of iterative Toeplitz solvers, Calcolo **33**, 249 (1996).

[56] O. Kyriienko, A. E. Paine, and V. E. Elfving, Solving nonlinear differential equations with differentiable quantum circuits, Phys. Rev. A **103**, 052416 (2021).

[57] Y. Subaşı, R. D. Somma, and D. Orsucci, Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing, Phys. Rev. Lett. **122**, 060504 (2019).