Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation

Yuki Sato [1,*] Ruho Kondo,[2] Satoshi Koide,[1] Hideki Takamatsu,[3] and Nobuyuki Imoto[4]

[1]*Toyota Central R&D Labs., Inc., Koraku Mori Building 10F, 1-4-14 Koraku, Bunkyo-ku, Tokyo 112-0004, Japan*
[2]*Toyota Central R&D Labs., Inc., 41-1 Yokomichi, Nagakute, Aichi 480-1192, Japan*
[3]*Toyota Motor Corporation, 1 Toyota-Cho, Toyota, Aichi 471-8571, Japan*
[4]*Institute for Photon Science and Technology, School of Science, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*

Computer-aided engineering techniques are indispensable in modern engineering developments. In particular, partial differential equations are commonly used to simulate the dynamics of physical phenomena, but very large systems are often intractable within a reasonable computation time, even when using supercomputers. To overcome the inherent limit of classical computing, we present a variational quantum algorithm for solving the Poisson equation that can be implemented in noisy intermediate-scale quantum devices. The proposed method defines the total potential energy of the Poisson equation as an expectation of certain observables, which are decomposed into a linear combination of Pauli operators and simple observables. The expectation value of these observables is then minimized with respect to a parameterized quantum state. Because the number of decomposed terms is independent of the size of the problem, this method requires relatively few quantum measurements. Numerical experiments demonstrate the faster computing speed of this method compared with classical computing methods and a previous variational quantum approach. We believe that our approach brings quantum computer-aided techniques closer to future applications in engineering developments.

I. INTRODUCTION

Partial differential equations (PDEs) are frequently used to describe the dynamics of physical phenomena such as heat conduction, fluid dynamics, and solid mechanics [1]. Solving these problems as quickly as possible is a key factor in accelerating engineering developments that combine computer simulations and real-world experiments. Although many advances have been made in terms of treating very large physical systems using classical computers [2,3], obtaining solutions within a reasonable computational time is increasingly intractable. State-of-the-art computational techniques can perform simulations of systems with up to tens of billions of degrees of freedom using Fugaku, which is one of the most powerful supercomputers; these simulations typically take several hours [4,5].

Another possible and attractive approach that would significantly reduce the computational costs of solving PDEs is the application of quantum computing. Quantum computing has attracted considerable attention over recent decades as a potential means of providing faster and more powerful computation than classical computing.

Quantum algorithms for solving linear systems have been developed [6,7] and provide an exponential speedup over classical algorithms when the coefficient matrix of the linear system is sparse. Cao *et al.* [8] developed an algorithm for solving the Poisson equation, which is one of the most important PDEs in various areas of engineering, such as electrostatics [9] and computational fluid dynamics [10,11]. Although it is widely believed that the above-mentioned quantum algorithms will demonstrate quantum supremacy once fault-tolerant quantum computers with sufficient qubits and error correction techniques are ready, it is expected to be a long time until such a quantum computer can be realized. Thus, recent interest in quantum computing has focused on the development of quantum algorithms that perform some meaningful computations with a small number of qubits and a shallow circuit. In other words, it is important to construct a quantum algorithm that can be implemented on so-called noisy intermediate-scale quantum (NISQ) devices [12].

Variational quantum algorithms (VQAs) [13] are possible candidates for use on NISQ devices, as they exhibit several advantages over classical algorithms. In VQAs, a certain cost function is written as a function of the expectation value of some observables, and this function is evaluated on a quantum computer using a trial quantum state prepared by a parameterized quantum circuit. This cost function evaluation is iteratively performed while the classical parameters are updated so as to minimize the cost function. Thus, VQAs are effectively hybrid quantum-classical algorithms. A well-known example of a VQA is a variational quantum eigensolver (VQE) [14] in which the cost function is the expectation of the system Hamiltonian. The VQE was originally developed to calculate the lowest eigenvalue of a system Hamiltonian via parameterized quantum circuits, also called ansatz [15], and has been widely studied in terms of the quantum circuits suitable for hardware architectures [16,17]

*yuki-sato@mosk.tytlabs.co.jp

and the construction of efficient optimizers [18–20]. Another popular VQA is the quantum approximate optimization algorithm (QAOA) [21], which can be used to solve combinational optimization problems. The QAOA was originally based on the concept of adiabatic quantum computing [22,23]; it has since been extended to more general families of operators and is thus widely known as the quantum alternating operator ansatz [24].

VQAs for solving linear systems have also been proposed. Bravo-Prieto *et al.* [25] proposed a VQA for a linear system in which the system matrix is written as a linear combination of unitary operators. This algorithm prepares a quantum state whose amplitude is proportional to the solution of a linear system output by the Harrow-Hassidim-Lloyd algorithm [6]. Liu *et al.* [26] presented a scheme for efficiently solving the Poisson equation by explicitly decomposing a system matrix derived from the finite difference discretization of the Poisson equation. This scheme requires $O(\log_2 N)$ expectation calculations at every iteration in the optimization procedure, where N is the size of the problem. However, these methods provide normalized solutions and do not explicitly provide the norm of the solutions. In the field of classical physics, the unknown quantities of PDEs directly describe physical quantities whose norms are not necessarily equal to 1. For example, the temperature field in steady-state heat conduction is governed by the Poisson equation, and we need to know the scale of the temperature when assessing whether the temperature in a certain device exceeds the operating temperature limit. In an engineering sense, therefore, the norms of these physical quantities provide important information.

This paper describes a method for solving the Poisson equation by formulating the cost function based on the concept of the minimum potential energy. This approach naturally leads to an expression for the norm of the solutions. The system matrix of the discretized Poisson equation is split into a linear combination of the tensor product of Pauli operators and simple observables. The number of decomposed terms is independent of the system size, i.e., $O(1)$, which significantly reduces the required number of expectation calculations on quantum computers.

The remainder of this paper is organized as follows. In Sec. II, the optimization problem for solving the Poisson equation is formulated based on the concept of VQAs. Section III describes the numerical implementation based on this formulation, and then Sec. IV provides several results from numerical experiments. Finally, the conclusions to this study are presented in Sec. V.

II. FORMULATION

A. Derivation of optimization problem

Consider the Poisson equation defined in an open bounded domain $\Omega \subset \mathbb{R}^d$, where d is the number of spatial dimensions. Let $u(\boldsymbol{x})$ denote the state field at the spatial coordinate $\boldsymbol{x} \in \mathbb{R}^d$. Consider a function $f(\boldsymbol{x}) : \Omega \to \mathbb{C}$. The Poisson equation is then defined as

$$-\nabla^2 u(\boldsymbol{x}) = f(\boldsymbol{x}) \ \text{in} \ \Omega, \tag{1}$$

where ∇^2 is the Laplace operator. As typical boundary conditions, the Neumann and Dirichlet boundary conditions are, respectively, defined as

$$-\boldsymbol{n} \cdot \nabla u(\boldsymbol{x}) = 0 \ \text{on} \ \Gamma_\text{N}, \tag{2}$$

$$u(\boldsymbol{x}) = 0 \ \text{on} \ \Gamma_\text{D}, \tag{3}$$

where Γ_N and Γ_D denote the boundaries on which the Neumann and Dirichlet boundary conditions are, respectively, imposed, $\boldsymbol{n}$ is the normal vector pointing outward from the boundary of the domain Ω, and $\Gamma_\text{N} \cup \Gamma_\text{D} = \partial\Omega$, which means that any point on $\partial\Omega$ is included in either Γ_N or Γ_D. In the following, the spatial coordinate $\boldsymbol{x}$ is omitted for readability.

Let us consider the total potential energy, defined as

$$E := \frac{1}{2} \int_\Omega \nabla v^* \cdot \nabla v \, d\Omega - \frac{1}{2} \int_\Omega v^* f \, d\Omega - \frac{1}{2} \int_\Omega f^* v \, d\Omega, \tag{4}$$

for $v \in \mathcal{V}$ with

$$\mathcal{V} := \{v \in H^1(\Omega) \mid v = 0 \ \text{on} \ \Gamma_\text{D}\}, \tag{5}$$

where $H^1(\Omega)$ is a Sobolev space. The stationary condition of the total potential energy with respect to a function v yields

$$
\begin{aligned}
0 &= dE(v; \delta v) \\
&= \frac{1}{2} \int_\Omega \nabla \delta v^* \cdot \nabla v \, d\Omega + \frac{1}{2} \int_\Omega \nabla v^* \cdot \nabla \delta v \, d\Omega \\
&\quad - \frac{1}{2} \int_\Omega \delta v^* f \, d\Omega - \frac{1}{2} \int_\Omega f^* \delta v \, d\Omega \\
&= \frac{1}{2} \int_{\Gamma_\text{N}} \delta v^* \boldsymbol{n} \cdot \nabla v \, d\Gamma - \frac{1}{2} \int_\Omega \delta v^* (\nabla^2 v + f) \, d\Omega \\
&\quad + \frac{1}{2} \int_{\Gamma_\text{N}} \delta v (\boldsymbol{n} \cdot \nabla v)^* \, d\Gamma - \frac{1}{2} \int_\Omega \delta v (\nabla^2 v + f)^* \, d\Omega,
\end{aligned}
\tag{6}
$$

where $dE(v; \delta v)$ represents the Gâteaux derivative of E with respect to v in the direction $\delta v \in \mathcal{V}$. In Eq. (6), Gauss's theorem was applied from the second line to the third, and the integration on Γ_D vanished because the trace of $\delta v = 0$ on the boundary where the Dirichlet boundary condition is imposed. Because δv is arbitrary, the above stationary condition recovers the Poisson equation from the second and fourth terms of the right-hand side of Eq. (6), which is known as the principle of minimum potential energy. This means that minimizing the total potential energy with respect to the function v yields the state field u, governed by the Poisson equation.

Now, to derive the cost function, Eq. (4) is discretized using some technique such as the finite difference or finite element method. The discretized version of the total potential energy, E_h, can be written as

$$E_h := \frac{1}{2} \boldsymbol{v}^* \cdot A\boldsymbol{v} - \frac{1}{2} \boldsymbol{v}^* \cdot \boldsymbol{f} - \frac{1}{2} \boldsymbol{f}^* \cdot \boldsymbol{v}, \tag{7}$$

where $\boldsymbol{v}$ and $\boldsymbol{f}$ denote vectors with component values of v and f, respectively, at the nodes discretizing the domain Ω. A is a positive-definite symmetric matrix obtained from the discretization of the first term in Eq. (4) into the quadratic form. Let $|v\rangle$ and $|f\rangle$ be the state vectors encoding $\boldsymbol{v}$ and $\boldsymbol{f}$, respectively. Without loss of generality, the squared norm of

f can be assumed to be 1 from the linearity of the Poisson equation, which means that $|f\rangle$ can be regarded as a quantum state. However, it should be noted that $|v\rangle$ is not necessarily a quantum state because the squared norm of the solution vector is not necessarily 1. To represent the solution vector using a parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle$ with parameter vectors $\boldsymbol{\theta}$, we thus introduce a parameter $r \in \mathbb{R}$ and then represent $|v\rangle = r|\psi(\boldsymbol{\theta})\rangle$. Substituting $|v\rangle = r|\psi(\boldsymbol{\theta})\rangle$ and $|f\rangle$, respectively, for $\boldsymbol{v}$ and $\boldsymbol{f}$ in Eq. (7), we obtain

$$E_h = \tfrac{1}{2} r^2 \langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle$$
$$- \tfrac{1}{2} r \langle \psi(\boldsymbol{\theta}) | f \rangle - \tfrac{1}{2} r \langle f | \psi(\boldsymbol{\theta}) \rangle. \quad (8)$$

Introducing the quantum state $|f, \psi(\boldsymbol{\theta})\rangle := (|0\rangle|f\rangle + |1\rangle|\psi(\boldsymbol{\theta})\rangle)/\sqrt{2}$, the discretized version of the total potential energy E_h in Eq. (7) can be described as

$$E_h(r, \boldsymbol{\theta}) = \tfrac{1}{2} r^2 \langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle$$
$$- r \langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle, \quad (9)$$

where $I = |0\rangle\langle 0| + |1\rangle\langle 1|$, $X = |0\rangle\langle 1| + |1\rangle\langle 0|$, and $n = \log_2 N$, where N is the number of nodes. Because r and $\boldsymbol{\theta}$ parametrize $|v\rangle$, minimizing the total potential energy with respect to $|v\rangle$ corresponds to the minimization of $E_h(r, \boldsymbol{\theta})$ in Eq. (9) with respect to r and $\boldsymbol{\theta}$. The optimization problem that solves the Poisson equation can, therefore, be formulated as

$$\min_{r, \boldsymbol{\theta}} E_h(r, \boldsymbol{\theta}). \quad (10)$$

Letting r_{opt} and $\boldsymbol{\theta}_{\text{opt}}$ denote the solution of this optimization problem, the state vector that encodes the solution of the Poisson equation is $|u\rangle = r_{\text{opt}}|\psi(\boldsymbol{\theta}_{\text{opt}})\rangle$. Here, these two minimizations with respect to r and $\boldsymbol{\theta}$ can be performed sequentially, i.e., minimizing E_h with respect to r, followed by $\boldsymbol{\theta}$ as

$$\min_{r, \boldsymbol{\theta}} E_h(r, \boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} E_h[r_{\text{opt}}(\boldsymbol{\theta}), \boldsymbol{\theta}], \quad (11)$$

where $r_{\text{opt}}(\boldsymbol{\theta})$ is the optimal value of r for a fixed value of $\boldsymbol{\theta}$. As the cost function E_h is parabolic with respect to r, the optimal solution of r for a given $\boldsymbol{\theta}$ is analytically derived by the following necessary condition for optimality, which requires that the partial derivative of $E_h(r, \boldsymbol{\theta})$ with respect to r is equal to 0:

$$0 = \frac{\partial E_h(r, \boldsymbol{\theta})}{\partial r}$$
$$= r \langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle - \langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle. \quad (12)$$

Then, we have

$$r_{\text{opt}}(\boldsymbol{\theta}) = \frac{\langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle}{\langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle}. \quad (13)$$

Consequently, r can be deleted from the cost function, and the optimization problem is to minimize the following Eq. (14) with respect to $\boldsymbol{\theta}$:

$$E_h[r_{\text{opt}}(\boldsymbol{\theta}), \boldsymbol{\theta}] = -\frac{1}{2} \frac{[\langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle]^2}{\langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle}. \quad (14)$$

Note that the denominator $\langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle$ is positive for arbitrary quantum states $|\psi(\boldsymbol{\theta})\rangle$ owing to the positive definiteness of the operator A.

B. Evaluation of cost function

In this paper, we focus on the one-dimensional Poisson equation, discretized by the finite element method (FEM) [27], in which the mesh size of all finite elements is 1. For N nodes in one dimension, the matrix A is described using first-order elements, depending on the boundary conditions, as follows: With periodic boundary conditions,

$$A_{\text{periodic}} := \begin{bmatrix} 2 & -1 & 0 & & \ldots & & 0 & -1 \\ -1 & 2 & -1 & 0 & \ldots & & & 0 \\ 0 & -1 & 2 & -1 & \ldots & & & 0 \\ \vdots & & & \ddots & & & & \vdots \\ 0 & & & & \ldots & 0 & -1 & 2 & -1 \\ -1 & 0 & & & \ldots & & 0 & -1 & 2 \end{bmatrix}$$
$$\in \mathbb{R}^{N \times N}. \quad (15)$$

With Dirichlet boundary conditions,

$$A_{\text{Dirichlet}} := \begin{bmatrix} 2 & -1 & 0 & & \ldots & & 0 \\ -1 & 2 & -1 & 0 & \ldots & & 0 \\ 0 & -1 & 2 & -1 & \ldots & & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & & & & \ldots & 0 & -1 & 2 & -1 \\ 0 & & & & \ldots & & 0 & -1 & 2 \end{bmatrix}$$
$$\in \mathbb{R}^{N \times N}. \quad (16)$$

With Neumann boundary conditions,

$$A_{\text{Neumann}} := \begin{bmatrix} 1 & -1 & 0 & & \ldots & & 0 \\ -1 & 2 & -1 & 0 & \ldots & & 0 \\ 0 & -1 & 2 & -1 & \ldots & & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & & & & \ldots & 0 & -1 & 2 & -1 \\ 0 & & & & \ldots & & 0 & -1 & 1 \end{bmatrix}$$
$$\in \mathbb{R}^{N \times N}. \quad (17)$$

The matrix $A_{\text{Dirichlet}}$ is the same as that derived in previous research [26] using the finite difference method (FDM). In one dimension, the matrices derived from the FEM using first-order elements coincide with those from the central difference scheme in the FDM.

While previous research [26] has shown that the matrix $A_{\text{Dirichlet}}$ can be decomposed into $O(n)$ terms consisting of Pauli and simple operators, the present study provides the decomposition of the above matrices A_{periodic}, $A_{\text{Dirichlet}}$, and A_{Neumann} into $O(1)$ terms consisting of Pauli operators and simple observables. The matrix A_{periodic} can be split into two matrices as follows:

$$A_{\text{periodic}} = A_{\mathcal{T}_{\text{even}}} + A_{\mathcal{T}_{\text{odd}}}, \quad (18)$$

where

$$
A_{\mathcal{T}_{\text{even}}} := \begin{bmatrix} 1 & -1 & 0 & 0 & & \cdots & & 0 \\ -1 & 1 & 0 & 0 & & \cdots & & 0 \\ 0 & 0 & 1 & -1 & & \cdots & & 0 \\ 0 & 0 & -1 & 1 & & \cdots & & 0 \\ \vdots & & & \ddots & & & & \vdots \\ 0 & & & & \cdots & 0 & 0 & 1 & -1 \\ 0 & & & & \cdots & 0 & 0 & -1 & 1 \end{bmatrix},
$$

(19)

$$
A_{\mathcal{T}_{\text{odd}}} := \begin{bmatrix} 1 & 0 & 0 & & \cdots & & 0 & -1 \\ 0 & 1 & -1 & 0 & \cdots & & & 0 \\ 0 & -1 & 1 & 0 & \cdots & & & 0 \\ \vdots & & & \ddots & & & & \vdots \\ 0 & & & & 0 & 1 & -1 & 0 \\ 0 & & & \cdots & 0 & -1 & 1 & 0 \\ -1 & 0 & & \cdots & & 0 & 0 & 1 \end{bmatrix}.
$$

(20)

Because $A_{\mathcal{T}_{\text{even}}}$ can be written in a simple form using Pauli operators as

$$
A_{\mathcal{T}_{\text{even}}} = I^{\otimes n-1} \otimes (I - X),
$$

(21)

and $A_{\mathcal{T}_{\text{even}}}$ and $A_{\mathcal{T}_{\text{odd}}}$ are interchanged by shifting the node number towards $+1$, $A_{\mathcal{T}_{\text{odd}}}$ can also be written using Pauli operators as

$$
A_{\mathcal{T}_{\text{odd}}} = P^{-1}[I^{\otimes n-1} \otimes (I - X)]P,
$$

(22)

where P is a shift operator defined as

$$
P := \sum_{i \in [0, 2^n - 1]} |(i + 1) \bmod 2^n\rangle\langle i|.
$$

(23)

Considering the differences between matrices depending on the three types of boundary conditions in Eqs. (15)–(17), these matrices can now be described as follows using the tensor products of the Pauli operators, a shift operator, and a simple Hermitian $I_0 = |0\rangle\langle 0|$:

$$
A_{\text{periodic}} = I^{\otimes n-1} \otimes (I - X) + P^{-1}[I^{\otimes n-1} \otimes (I - X)]P, \quad (24)
$$

$$
A_{\text{Dirichlet}} = A_{\text{periodic}} + P^{-1}\big(I_0^{\otimes n-1} \otimes X\big)P, \quad (25)
$$

$$
A_{\text{Neumann}} = A_{\text{periodic}} - P^{-1}\big[I_0^{\otimes n-1} \otimes (I - X)\big]P. \quad (26)
$$

The expectation values of observables including a shift operator can be evaluated as follows by applying the shift operator, which is unitary, to the quantum state:

$$
\langle \phi | P^{-1}HP | \phi \rangle = \langle \phi' | H | \phi' \rangle, \quad (27)
$$

where $|\phi\rangle$ is an arbitrary n-qubit state and $|\phi'\rangle = P|\phi\rangle$. Now, the denominator of Eq. (14) is described using a linear combination of the Pauli operators and simple Hermitians I_0, I_1, and the numerator of Eq. (14) is originally expressed by the Pauli operators. Because the expectation values of the identity operator are 1, i.e.,

$$
\langle \phi | I^{\otimes n} | \phi \rangle = 1, \quad (28)
$$

$$
\langle \phi | P^{-1}I^{\otimes n}P | \phi \rangle = \langle \phi' | I^{\otimes n} | \phi' \rangle = 1, \quad (29)
$$

the number of terms to be evaluated by a quantum computer is three for the periodic boundary condition, four for the Dirichlet boundary condition, and five for the Neumann boundary condition; of these, one term is for the numerator of Eq. (14) and the others are for the linear combination in the denominator of Eq. (14).

The above discussion can easily be extended to d-dimensional problems using the FDM and FEM. When using the FDM, the cost function is defined by the d-dimensional system matrix A_d, that is,

$$
A_d := A \otimes I^{\otimes(d-1)} + I \otimes A \otimes I^{\otimes(d-2)} + \cdots + I^{\otimes(d-1)} \otimes A, \quad (30)
$$

instead of A [8,26]. This increases the number of terms to be measured by a factor of d compared with the one-dimensional case, i.e., the number of terms to be measured is $O(d)$. When using the FEM, the decomposition of A_d into a linear combination of Pauli operators is derived by defining a graph corresponding to the finite elements, which is explained in Appendix A.

III. IMPLEMENTATION

A. Overview of the algorithm

Here, we briefly describe the proposed algorithm.

Step 1. Initialize a set of parameters $\boldsymbol{\theta}$ in a classical computer.

Step 2. Evaluate the cost function E_h in Eq. (14) using a quantum computer.

Step 3. If a certain terminal condition is satisfied, the optimization procedure halts; otherwise, proceed to Step 4.

Step 4. Update the set of parameters using some classical optimization scheme, then return to Step 2.

We use several kinds of terminal conditions in the numerical experiments. These will be specified in Sec. IV.

B. State preparation

In the proposed method, the quantum state $|\psi(\boldsymbol{\theta})\rangle$ is prepared by applying a sequence of parameterized quantum gates $U(\boldsymbol{\theta})$, the so-called ansatz, to the $|0\rangle^{\otimes n}$ state. We use a hardware-efficient ansatz [16], specifically, an alternating layered ansatz consisting of R_Y gates and controlled Z gates [25], to constrain the amplitude in the real space. This is valid in the solution of the Poisson equation with $f(\boldsymbol{x}) \in \mathbb{R}$. In the state-preparation stage, the state vector $|f\rangle$, which corresponds to the source term of the Poisson equation, must also be prepared. To encode arbitrary state vectors, amplitude encoding techniques are required [28–30]. These encode classical data into the amplitudes of a quantum state. In the current study, for simplicity, we assume that there is a unitary U_f that efficiently prepares $|f\rangle$ from $|0\rangle^{\otimes n}$ (i.e., $|f\rangle = U_f|0\rangle^{\otimes n}$). The quantum state $|f, \psi(\boldsymbol{\theta})\rangle$ for evaluating the numerator of the cost function in Eq. (14) is prepared using an auxiliary qubit and controlled versions of the parameterized quantum circuit $U(\boldsymbol{\theta})$ and the unitary U_f of the quantum circuit shown in Fig. 1, which is more expensive than circuits preparing $|\psi(\boldsymbol{\theta})\rangle$ and $|f\rangle$. Note that when we constrain the amplitudes of $|\psi(\boldsymbol{\theta})\rangle$ and $|f\rangle$ in the real space, the numerator of the cost function can also be evaluated without the quantum state $|f, \psi(\boldsymbol{\theta})\rangle$ as
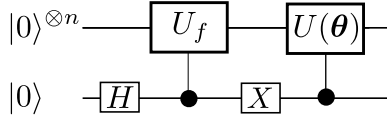
FIG. 1. Quantum circuit for preparing $|f, \psi(\boldsymbol{\theta})\rangle := [|0\rangle|f\rangle + |1\rangle|\psi(\boldsymbol{\theta})\rangle]/\sqrt{2}$. The leftmost qubit is shown in the bottom line of the circuit.

follows:

$$
\begin{aligned}
&[\langle f, \psi(\boldsymbol{\theta}) \,|\, X \otimes I^{\otimes n} \,|\, f, \psi(\boldsymbol{\theta})\rangle]^2 \\
&= \langle \psi(\boldsymbol{\theta}) \,|\, f\rangle \langle f \,|\, \psi(\boldsymbol{\theta})\rangle \; \{\text{if Im}[\langle \psi(\boldsymbol{\theta}) \,|\, f\rangle] = 0\} \\
&= \langle \psi(\boldsymbol{\theta})|U_f|0\rangle \langle 0|U_f^{\dagger}|\psi(\boldsymbol{\theta})\rangle,
\end{aligned} \tag{31}
$$

where the first equality holds when the amplitudes of $|\psi(\boldsymbol{\theta})\rangle$ and $|f\rangle$ are in the real space.

C. Quantum circuit for the shift operator

The shift operator P is represented by a sequence of multicontrolled Toffoli gates with, at most, $n - 1$ control lines, as shown in Fig. 2. With $k \geqslant 3$ denoting the number of control lines, a k-controlled Toffoli gate can be decomposed into $2k - 4$ relative-phase Toffoli gates and a Toffoli gate using $k - 2$ auxiliary qubits [31]. Because a shift operator has k-controlled Toffoli gates for $3 \leqslant k \leqslant n - 1$, a CNOT gate, and an X gate, it can be expressed by $(n - 2)(n - 3)$ relative-phase Toffoli gates, $n - 3$ Toffoli gates, a CNOT gate, and an X gate. Consequently, a quantum circuit for a shift operator requires $O(n^2)$ depth and $2n - 3$ qubits, including auxiliary qubits. Note that this implementation of the shift operator P is the major bottleneck for making a quantum circuit shallow, and so it is challenging to perform the proposed method directly on NISQ devices. Thus, an efficient implementation of the shift operator is a crucial aspect that we will address in future research. For example, Oomura *et al.* [32] proposed an efficient implementation of the Toffoli gate employing the IBM Q Open Pulse Systems, which halves the gate time and improves the fidelity. We believe that such pulse design approaches are promising.

The decomposition of the matrix A introduced in a previous approach [26] can also be used in our proposed method. Though this decomposition yields $O(n)$ terms to be measured, the fact that the shift operator is not used means that the quantum circuit remains shallow. With the use of this decomposition, the previous method requires the expectations of $X \otimes A$ and A^2, which yield $2n + 1$ and $4n + 1$ terms to
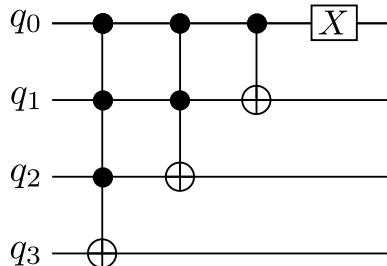
be measured, respectively, while our formulation requires the expectations of $X \otimes I$ and A, which yield 1 and $2n + 1$ terms to be measured, respectively. Therefore, our proposed method has the advantage that the number of terms to be measured is roughly one-third of that in the previous method.

D. Number of shots for expectation estimation

To estimate the expectation value of a certain observable using a quantum computer, a quantum circuit with state preparation and measurement is run many times to sample an observable value and apply the Monte Carlo approach. Each run of a quantum circuit to obtain a sample is referred to as a "shot." In this section, we estimate the number of shots required to estimate the expectations.

In the proposed method, the expectations to be estimated are $\langle f, \psi(\boldsymbol{\theta})\,|\, X \otimes I^{\otimes n} \,|\, f, \psi(\boldsymbol{\theta})\rangle$ and $\langle \psi(\boldsymbol{\theta})\,|\, A \,|\, \psi(\boldsymbol{\theta})\rangle$, where the latter consists of several terms depending on the boundary conditions, as mentioned in Sec. II B. Let $q_i^{(j)}$ denote the jth sample value for estimating the ith expectation value. Here, $q_1^{(j)}$ denotes the value of the jth sample of $\langle f, \psi(\boldsymbol{\theta})\,|\, X \otimes I^{\otimes n} \,|\, f, \psi(\boldsymbol{\theta})\rangle$, whereas $q_{\geqslant 2}^{(j)}$ denotes the value of the jth sample of $\langle \psi(\boldsymbol{\theta})\,|\, A \,|\, \psi(\boldsymbol{\theta})\rangle$. In the following discussion, the parameter values $\boldsymbol{\theta}$ are assumed to be fixed and are omitted to simplify the notation. Using S_i shots, the ith expectation value $\bar{q}_i$ is estimated as follows:

$$
\bar{q}_i := \frac{\sum_{j=1}^{S_i} q_i^{(j)}}{S_i}. \tag{32}
$$

Regarding $q_i^{(j)}$ as a random variable with a mean value of μ_i and a variance of σ_i^2, the mean and variance of $\bar{q}_i$ are written as

$$
\mathbb{E}[\bar{q}_i] = \mu_i, \tag{33}
$$

$$
\text{Var}[\bar{q}_i] = \frac{\sum_{j=1}^{S_i} \text{Var}[q_i^{(j)}]}{S_i^2} = \frac{\sum_{j=1}^{S_i} \sigma_i^2}{S_i^2} = \frac{\sigma_i^2}{S_i}. \tag{34}
$$

The variance of $\bar{q}_i$ corresponds to a squared standard error of $q_i^{(j)}$, the standard error of which is $\sigma_i/\sqrt{S_i}$.

Now, the cost function E_h in Eq. (14) is written using μ_i, and is assumed to be estimated as follows using the approximated expectation values:

$$
E_h = -\frac{1}{2}\frac{\mu_1^2}{\sum_{i=2}^{m} \mu_i} \approx -\frac{1}{2}\frac{\bar{q}_1^2}{\sum_{i=2}^{m} \bar{q}_i} =: g(\bar{q}_1, \ldots, \bar{q}_m), \tag{35}
$$

with $g(\bar{q}_1, \ldots, \bar{q}_m)$ denoting the approximated cost function, where $m = 3$ for the periodic boundary condition, $m = 4$ for the Dirichlet boundary condition, and $m = 5$ for the Neumann boundary condition, which corresponds to the number of terms to be evaluated derived in Sec. II B.

Note that sampling using quantum computers is performed independently for each term of the expectation, leading to $\text{Cov}(\bar{q}_i, \bar{q}_{i'}) = 0$ for $i \neq i'$, where $\text{Cov}(\bar{q}_i, \bar{q}_{i'})$ is the covariance of $\bar{q}_i$ and $\bar{q}_{i'}$. Using the first-order Taylor series expansion of $g(\bar{q}_1, \ldots, \bar{q}_m)$ around μ_i for $i \in [1, m]$, the mean squared error ε^2 between E_h and $g(\bar{q}_1, \ldots, \bar{q}_m)$ in Eq. (35) is then written as

FIG. 2. Quantum circuit for the shift operator.

follows:

$$\varepsilon^2 = r_{\mathrm{opt}}^2 \left(\frac{\sigma_1^2}{S_1} + \frac{1}{4} r_{\mathrm{opt}}^2 \sum_{i=2}^{m} \frac{\sigma_i^2}{S_i} \right). \tag{36}$$

A detailed derivation of the above equation is provided in Appendix B. In the proposed method, for simplicity, we use the same number of shots, denoted by S, for all sampling processes, which yields

$$\varepsilon^2 \approx r_{\mathrm{opt}}^2 \left(\sigma_1^2 + \frac{1}{4} r_{\mathrm{opt}}^2 \sum_{i=2}^{m} \sigma_i^2 \right) \frac{1}{S}. \tag{37}$$

This equation implies that the mean square error is inversely proportional to the number of shots.

E. Time complexity

Here, we analyze the time complexity of the proposed method in terms of state preparation, number of quantum circuits, number of shots, and the number of iterations required to optimize the parameter set $\boldsymbol{\theta}$. Note that the time complexity of the classical computing parts, such as parameter initialization and update, depends on the classical implementation and optimizers; thus, it is omitted from this analysis.

The time complexity of state preparation represents the time required for setting up the quantum circuit before performing the measurements to estimate a certain expectation value. Therefore, the time complexity of state preparation, denoted by T_P, can be estimated by the depth of the quantum circuit of state preparation:

$$T_P := O(D_{\mathrm{ansatz}} + D_{\mathrm{enc}} + D_{\mathrm{shift}}) \tag{38}$$

$$= O(D_{\mathrm{ansatz}} + D_{\mathrm{enc}} + n^2), \tag{39}$$

where D_{ansatz}, D_{enc}, and D_{shift} denote the circuit depths of the ansatz, amplitude encoding, and shift operator, respectively.

To estimate the cost function value using a quantum computer, several quantum circuits are required, corresponding to the numerator of Eq. (14) and each term in Eqs. (24)–(26) in the denominator of Eq. (14). The required number of quantum circuits, T_C, is

$$T_C := O(1) \tag{40}$$

because $T_C = 3$ for periodic boundary conditions, $T_C = 4$ for Dirichlet boundary conditions, and $T_C = 5$ for Neumann boundary conditions, independent of the scale of the problem, n. Furthermore, when we use gradient-based optimizers, we need several quantum circuits, the number of which is proportional to the number of parameters in the parameter set $\boldsymbol{\theta}$, to evaluate the gradient of the cost function. As the number of parameters is $O(nD_{\mathrm{ansatz}})$, the number of quantum circuits required, denoted by T_G, is

$$T_G := O(nD_{\mathrm{ansatz}}). \tag{41}$$

To evaluate the cost function and its gradient, each quantum circuit must be run many times for the sampling required to estimate the expectation values. Based on the discussion in Sec. III D, the required number of shots is

$$T_S := O\left(\frac{1}{\varepsilon^2} \right). \tag{42}$$

In the proposed method, the above-mentioned evaluations of the cost function and its gradient through quantum circuits are repeated while the parameter set $\boldsymbol{\theta}$ is updated, as discussed in Sec. III A. The number of iterations is strongly dependent on the classical optimization solver and the terminal condition setting. As the discussion of classical optimization solvers is beyond the scope of this paper, let T_{it} denote the number of iterations, for simplicity.

Consequently, the total time complexity can be derived as

$$T := T_{\mathrm{it}} T_P (T_C + T_G) T_S$$

$$= O\left[\frac{T_{\mathrm{it}}(D_{\mathrm{ansatz}} + D_{\mathrm{enc}} + n^2) n D_{\mathrm{ansatz}}}{\varepsilon^2} \right]. \tag{43}$$

The time complexity for solving the Poisson equation by classical computing is $O(N \log_2 N)$ [33], where N is the size of the matrix A, i.e., $N = 2^n$. Thus, the proposed method has reduced time complexity compared with classical algorithms, as long as the number of optimization iterations T_{it}, the depth of the ansatz D_{ansatz}, and the depth of the quantum circuit for the amplitude encoding D_{enc} are relatively small, i.e., $O[\mathrm{poly}(n)]$.

The time complexity of the previous method [26] can also be evaluated through a similar procedure. Here, only the result is provided:

$$T = O\left[\frac{T_{\mathrm{it}}(D_{\mathrm{ansatz}} + D_{\mathrm{enc}}) n^2 D_{\mathrm{ansatz}}}{\varepsilon^2} \right]. \tag{44}$$

Hence, if either the depth of the ansatz or that of the amplitude encoding is greater than $O(n)$, the proposed method has reduced time complexity compared with the previous method. As for amplitude encoding, a depth of $O(n^2)$ is required to encode arbitrary real-valued inputs [30], which implies that the proposed method has reduced time complexity compared with the previous method.

F. Barren plateaus

Note that the proposed cost function will suffer from the problem of exponentially vanishing gradients, i.e., barren plateaus, because of the definition of the cost function and the use of an alternating layered ansatz [34]. Because the operator $A_{\mathcal{T}_{\mathrm{even}}} = I^{\otimes n-1} \otimes (I - X)$ in the denominator of the cost function acts nontrivially on only one qubit, i.e., the operator is local, the expectation of the operator is resilient to the problem of barren plateaus. However, the operator $A_{\mathcal{T}_{\mathrm{odd}}} = P^{-1}[I^{\otimes n-1} \otimes (I - X)]P$ acts nontrivially on all qubits because of the shift operator, i.e., it is global, and so the expectation of the operator will be affected by barren plateaus. Moreover, although the operator $X \otimes I$ in the numerator of the cost function is local, the expectation of the operator will be affected by barren plateaus because the numerator inherently evaluates the inner product of states $|\psi(\boldsymbol{\theta})\rangle$ and $|f\rangle$ by definition, which is a global quantity. Consequently, the cost function as a whole will be affected by barren plateaus, even though one term is resilient to the problem. Therefore, the required order of shots needed to evaluate the gradients will be exponential with respect to a certain error because of the exponentially vanishing gradients. We will attempt to alleviate

this problem in future research. A more detailed discussion is provided in Appendix D 1.

IV. NUMERICAL EXPERIMENTS

This section describes the results of several numerical experiments that demonstrate the validity of the proposed method. The proposed method was implemented in QISKIT version 0.23 [35], an open-source framework for working with quantum computers. The Statevector Simulator backend in AER, which is a high-performance quantum computing simulator operating with QISKIT, was used for the calculations, except for those reported in Sec. IV B 3, where the QASM Simulator backend was used. As an optimizer for updating $\boldsymbol{\theta}$, we employed the Broyden-Fletcher-Goldfarb-Shanno method [36–39], with the gradient of the cost function evaluated by quantum computing. An analytic derivation of the expression for the gradient is provided in Appendix C.

In the experiments reported below, the unitary U_{b} for preparing $|f\rangle$ was set as

$$U_{\mathrm{b}} = H^{\otimes n} X \otimes I^{\otimes(n-1)}, \qquad (45)$$

where H represents a Hadamard gate. This unitary U_{b} prepares $|f\rangle$ in the form of a step function from $1/2^{n/2}$ to $-1/2^{n/2}$. The number of layers of the ansatz was fixed at 5.

A. Solutions for three types of boundary conditions

First, we provide solutions obtained by the proposed method for the three types of boundary conditions to show the applicability of the proposed method to these basic types. Here, the Statevector Simulator backend in AER was used to evaluate the proposed method in an ideal environment without any noise or sampling errors.

In this experiment, the optimization procedure was terminated when the norm of the gradients became less than the predetermined threshold value. The optimization was performed 10 times from randomly set initial parameters $\boldsymbol{\theta}$ for each condition. When imposing the periodic or Neumann boundary conditions on both edges, a regularization term ϵI with $\epsilon = 10^{-3}$ was added to the system matrix A to prevent the matrix from becoming singular.

Figures 3(a)–3(c) show the components of the solution vectors on each node from 10 trials when $n = 5$. As shown in these figures, the solutions for the periodic and Dirichlet boundary conditions are similar to each other, whereas the Neumann boundary condition gives a totally different solution when the input vector $|f\rangle$ is prepared using the unitary operator in Eq. (45). Figures 3(a)–3(c) also indicate that the proposed algorithm underestimates the norms of the solution vectors, although the directions of the solution vectors given by the proposed method are in good agreement with those from classical computing.

Figure 4 shows the trace distance $\varepsilon_{\mathrm{tr}}$ between the solutions obtained by the proposed method and the classical computation when calculating $A^{-1}\boldsymbol{f}$ with respect to the number of qubits. The trace distance $\varepsilon_{\mathrm{tr}}$ between the trial state $|\psi(\boldsymbol{\theta})\rangle$ and the normalized ground truth $|\bar{u}\rangle := |u\rangle/\sqrt{\langle u \mid u\rangle}$ is defined as

$$\varepsilon_{\mathrm{tr}} := \tfrac{1}{2}\mathrm{Tr}(\sqrt{(|\psi\rangle\langle\psi| - |\bar{u}\rangle\langle\bar{u}|)^2}) = \sqrt{1 - |\langle\psi \mid \bar{u}\rangle|^2}. \quad (46)$$
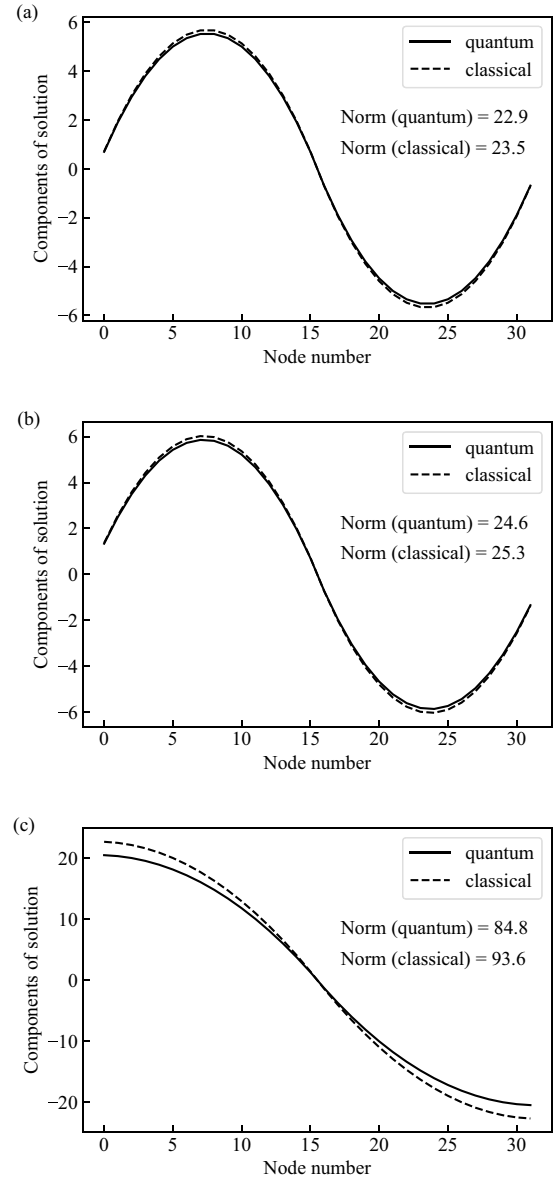
FIG. 3. Distribution of solutions. The components of the solution vectors on each node are plotted. (a) Periodic boundary conditions. (b) Dirichlet boundary conditions. (c) Neumann boundary conditions.

The plots and error bars represent the mean values and the standard deviations of 10 experiments. This figure shows that the trace distance $\varepsilon_{\mathrm{tr}}$ is less than 0.01 for all cases, which results in the fidelity $\langle\psi \mid \bar{u}\rangle^2$ being greater than 0.9999. These trace distance and fidelity values compare favorably with those reported in previous research [25,26]. Therefore, the errors between the solution vectors obtained by the proposed algorithm and a classical approach are mainly caused by errors in the norms, and the directions of the solution vectors given by the proposed method actually agree with those from classical computing. Because the errors of the solution depend on the Hilbert space expressed by the ansatz, we hope to conduct future research on the ansatz that is most suitable for expressing the solutions of certain PDEs.
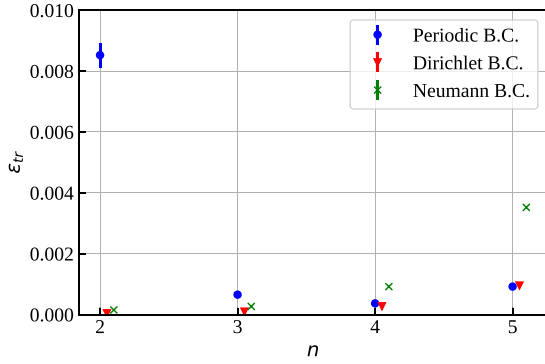
FIG. 4. Trace distances $\varepsilon_{\mathrm{tr}}$ vs the number of qubits, n. The mean values of 10 trials are plotted, with the error bars representing the standard deviations. The x axis is slightly shifted for different legends for visibility.

B. Comparison of the proposed method with the previous method

We now compare the proposed method with the previous approach [26] to evaluate the performance of the proposed method. In the previous method, the cost function to be minimized is formulated as

$$E(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | A(I - |f\rangle\langle f|)A | \psi(\boldsymbol{\theta})\rangle, \qquad (47)$$

which corresponds to the maximization of the cosine similarity of $A|\psi(\boldsymbol{\theta})\rangle$ and $|f\rangle$. Because the cosine similarity does not take into account the norm information, the previous method does not explicitly provide the norm. Note that the norm can be calculated once the normalized solution has been obtained. Letting r denote the norm of the solution, the equation $Ar|\psi(\boldsymbol{\theta})\rangle = |f\rangle$, which holds at the optimal point of $\boldsymbol{\theta}$ if the expressibility of the ansatz is sufficiently high, yields the following expression:

$$r = \frac{1}{\sqrt{\langle \psi(\boldsymbol{\theta}) | A^2 | \psi(\boldsymbol{\theta})\rangle}}. \qquad (48)$$

We also implemented the previous method using QISKIT by referring to the original paper.

In the following experiments, except those reported in Sec. IV B 1, the proposed method used the Dirichlet boundary conditions imposed on both edges. The results are compared with those of the previous approach using the Dirichlet boundary conditions.

1. Dependency of the number of circuit executions per cost function evaluation on the number of qubits

First, we examined the dependency of the number of circuit executions on the number of qubits to evaluate the scalability of the proposed method. For randomly set parameters $\boldsymbol{\theta} \in [0, 4\pi]$, the number of circuit executions for evaluating a cost function value was recorded for both the proposed and previous methods.

Figure 5 illustrates the number of circuit executions per cost function evaluation with respect to the number of qubits for both the proposed and previous methods [26]. This figure clearly shows that the proposed method only requires $O(1)$ measurements per cost function evaluation, whereas the previ-
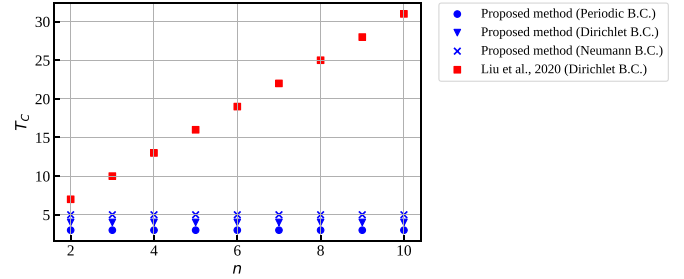

FIG. 5. Number of circuit executions per cost function evaluation, T_C, vs the number of qubits for both the proposed and previous methods [26].

ous method requires $O(n)$ measurements. This result confirms the validity of the proposed formulation given in Sec. II B, implying that our proposed method significantly reduces the required number of expectation calculations on quantum computers.

2. Dependency of the number of iterations on the number of qubits

Next, we examined the dependency of the number of optimization iterations on the number of qubits to evaluate the scalability of the classical computing part of the proposed method.

In this experiment, the terminal condition was based on the tolerance of the trace distance $\varepsilon_{\mathrm{tr}}$ between the trial state $|\psi(\boldsymbol{\theta})\rangle$ and the normalized ground truth $|\bar{u}\rangle := |u\rangle/\sqrt{\langle u | u\rangle}$, which is the same criterion used in previous research [25]. Note that this metric only evaluates the difference in direction between the trial state and the true solution; the difference in the norm cannot be determined. In spite of this defect, we used the metric to compare the proposed method with the previous approach according to the same criterion used in the previous method. The tolerance of the trace distance was successively set to $\varepsilon_{\mathrm{tr}} = 0.1, 0.03, 0.01$.

For each condition, the optimization was run 10 times, using randomly set initial parameters in $[0, 4\pi]$. Figure 6 shows the decadic logarithm of the number of optimization iterations with respect to the decadic logarithm of the number of qubits for both the proposed method and the previous method [26]. The plots and error bars represent the mean values and the standard deviations of 10 experiments, respectively. This figure clearly shows that the number of iterations is positively correlated with the number of qubits. Although it is difficult to assert the time complexity of the number of iterations numerically because of the error bars, these plots were fitted to the lines that minimize the squared errors of fitting. The lines for the proposed method have slopes of 2.6 for $\varepsilon_{\mathrm{tr}} = 0.1$, 2.3 for $\varepsilon_{\mathrm{tr}} = 0.03$, and 2.5 for $\varepsilon_{\mathrm{tr}} = 0.01$. This implies that T_{it} in Eq. (43) is, at most, $O(n^{2.6})$ in this experiment. For the previous method, the lines have slopes of 3.5 for $\varepsilon_{\mathrm{tr}} = 0.1$, 3.6 for $\varepsilon_{\mathrm{tr}} = 0.03$, and 4.2 for $\varepsilon_{\mathrm{tr}} = 0.01$. Therefore, it seems that the number of iterations with respect to the number of qubits is of similar order in both the proposed method and the previous method.
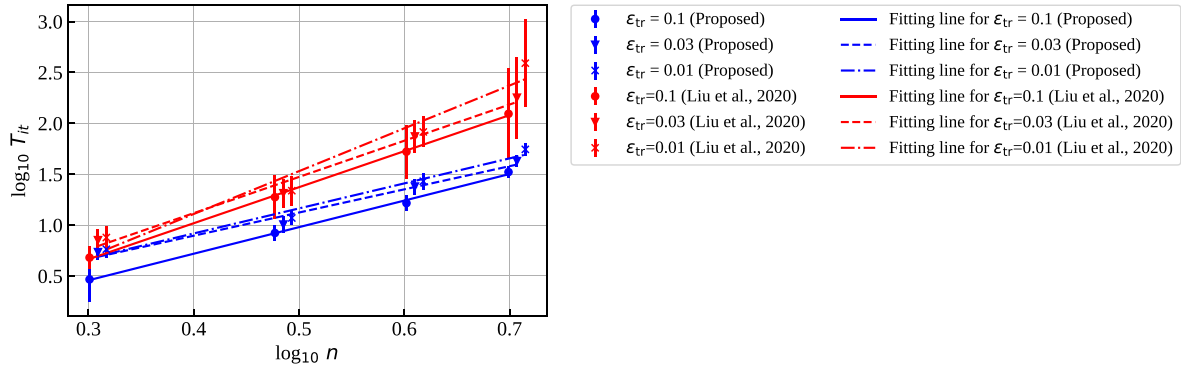
FIG. 6. Decadic logarithm of the number of optimization iterations, T_{it}, vs the number of qubits, n, for both the proposed and previous methods [26]. The points show the mean values of 10 experiments and the fitted lines minimize the squared errors of fitting. The error bars represent the standard deviation of 10 experiments. The x axis is slightly shifted for different legends for visibility.

3. Dependency of the sampling errors on the number of shots

Next, we examined the effect of the number of shots on the expectation estimations. The QASM Simulator backend was used to evaluate the sampling errors in the environment without any noise.

For randomly set parameters $\boldsymbol{\theta} \in [0, 4\pi]$, the squared error between the ground truth and the estimation value of the cost function was evaluated. The former was calculated by the Statevector Simulator and the latter by the QASM Simulator (i.e., sampling). The cost function was evaluated 10 times by sampling for each number of shots.

Figure 7 shows the decadic logarithm of the squared error of the cost function with respect to the decadic logarithm of the number of shots for the proposed method. The points show the mean values of the 10 experiments and the error bars represent their standard deviations. Fitting lines are also provided in the plots. This figure clearly indicates that the squared error decreases as the number of shots increases. The mean and standard deviation of the slopes are -1.11 and 0.12, respectively, while the theoretical slope is -1, as derived in Eq. (37). Note that it does not make sense to compare the magnitude of errors between different numbers of qubits. Because this figure is plotted for a fixed parameter, which is set randomly for each number of qubits, the plots necessarily exhibit monotonicity with respect to n. We also examined the

effect of the number of shots on the expectation estimations for the previous method, and the results are illustrated in Appendix D 2.

In addition to the cost function evaluation, it is important to precisely evaluate the gradients when using a gradient-based optimization method. Therefore, we also evaluated the errors between the gradient evaluated by the sampling obtained using the QASM Simulator and that computed by the Statevector Simulator with fixed parameters.

As a metric to evaluate the errors, we used the cosine similarity, which measures the similarity of directions, as the directions of the gradients are more important to optimizers than their norms. Figure 8 shows the decadic logarithm of "1 − cosine similarity" with respect to the number of shots for both the proposed and previous methods. This figure clearly shows that the cosine similarity increases as the number of shots increases. When $n = 10$, the slope of the fitting line increases. This is caused by the existence of barren plateaus, whereby more shots are required to evaluate small gradients precisely. The means and standard deviations of the slopes of the fitted lines for $n = 2, \ldots, 10$ (which includes cases that do not appear in Fig. 8) are -0.99 and 0.14 for our proposed method and -0.98 and 0.03 for the previous method, respectively. Therefore, it seems that the gradient estimation for a given number of shots is of similar order in both the proposed and previous methods.
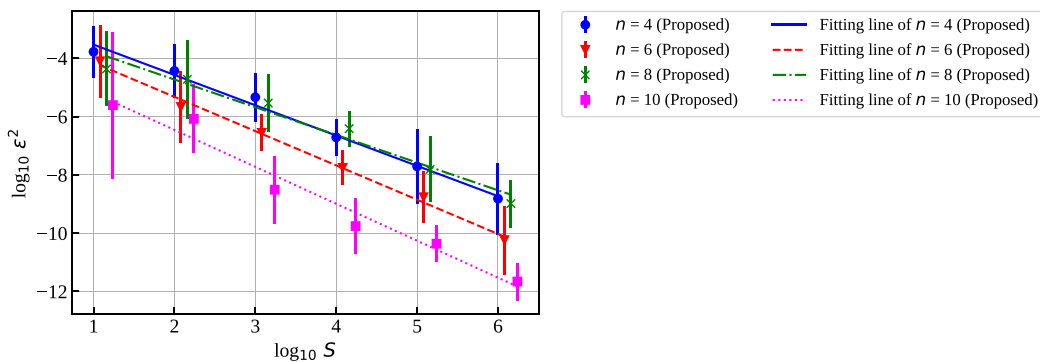

FIG. 7. Decadic logarithms of cost function error ε vs the number of shots S for the proposed method. The optimization was run 10 times with fixed parameters for each number of shots. The mean values are plotted, with the error bars representing the standard deviations. The mean and standard deviation of the slopes for $n = 2, \ldots, 10$ (which includes cases that are not shown in this figure) are -1.11 and 0.12, respectively. The x axis is slightly shifted for different legends for visibility.
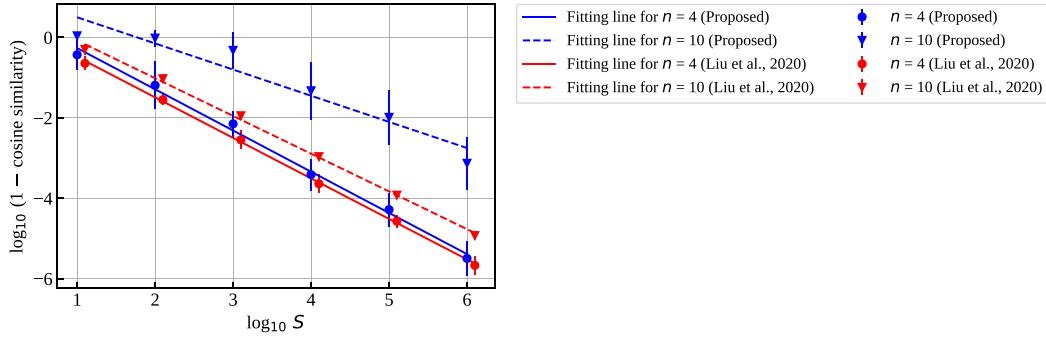
FIG. 8. Decadic logarithms of "1 − cosine similarity" vs the number of shots S for the proposed and previous methods [26] based on our implementation. The optimization was run 10 times with fixed parameters for each number of shots. The mean values are plotted, with the error bars representing the standard deviations. The means and standard deviations of the slopes of the fitted lines for $n = 2, \dots, 10$ (which includes cases that are not shown in this figure) are -0.99 and 0.14 for our proposed method and -0.98 and 0.03 for the previous method, respectively. The x axis is slightly shifted for different legends for visibility.

However, we can clearly observe that the proposed method has longer error bars, i.e., a larger standard deviation of sampling, than the previous method. This comes from the difference in the definition of the cost function, with the proposed method considering the norm of the solution vector as well as its direction in the optimization procedure. This implies that more shots are required to estimate the norm of the solution in addition to its direction.

V. CONCLUSIONS

This paper has presented a VQA for solving the Poisson equation based on the minimum potential energy. The main contributions of the present study are as follows: (1) we have provided an explicit decomposition of the system matrix for the Poisson equation into $O(1)$ terms consisting of simple observables, (2) the proposed method provides information about the norm of the solution vectors in addition to the direction of the vectors, and (3) the time complexity of the proposed algorithm has been derived and verified. The first contribution implies that the proposed method only requires a small number of quantum measurements, compared with conventional approaches, at every iteration of the optimization procedure. The second contribution enhances the ability of VQAs to solve PDEs because the norm information is essential when using the calculation results for engineering developments. As for the third contribution, we estimated the time complexity of the proposed method and demonstrated that it has significant potential for reducing the computation time of classical computing algorithms. The number of optimization iterations and the depth of the ansatz depend on the classical optimization and the architecture of the ansatz, respectively. To derive the theoretical total time complexity, these aspects will be discussed in future work.

We believe the present study elevates the application of quantum computing to the field of computer-aided engineering and, moreover, design optimization.

ACKNOWLEDGMENTS

APPENDIX A: DECOMPOSITION OF STIFFNESS MATRIX IN FINITE ELEMENT METHODS USING GRAPH THEORY

First, a one-dimensional problem is considered. Let us consider an N-node cycle graph $G = (V, E)$, where V and E represent node and edge sets, respectively. This N-node cycle graph corresponds to the one-dimensional finite element with a periodic boundary condition. A clique is defined as a subset of several nodes that form a complete subgraph. A tessellation $\mathcal{T}$ is then defined as a set of cliques such that all nodes belong to one clique. The tessellation $\mathcal{T}$ includes edges whose endpoints belong to the tessellation. Generally, several tessellations can be defined in a graph, and there exists a set of tessellations such that all edges of the graph are included in at least one tessellation. Such a set of tessellations is called a tessellation cover [40]. Assuming that $N = 2^n$, two tessellations of the tessellation cover of the N-node cycle graph can be defined as

$$\mathcal{T}_{\text{even}} := \{C_{2i} \mid i \in [0, 2^{n-1} - 1]\}, \tag{A1}$$

$$\mathcal{T}_{\text{odd}} := \{C_{2i+1} \mid i \in [0, 2^{n-1} - 1]\}, \tag{A2}$$

where $C_i := \{v_i, v_{i+1}\}$ is a clique consisting of the i-th node v_i and the $(i + 1)$th node v_{i+1}. Note that we define $v_N := v_0$ to simplify the notation.
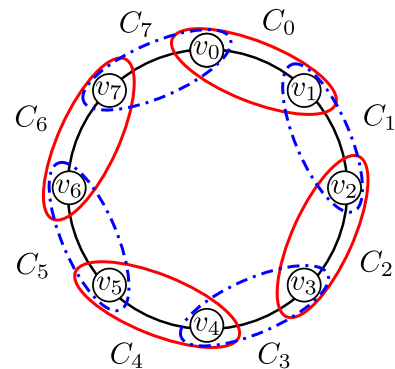

FIG. 9. Example of tessellations for an 8-node cycle graph. Red and blue ellipses indicate the tessellations $\mathcal{T}_{\text{even}}$ and $\mathcal{T}_{\text{odd}}$, respectively.
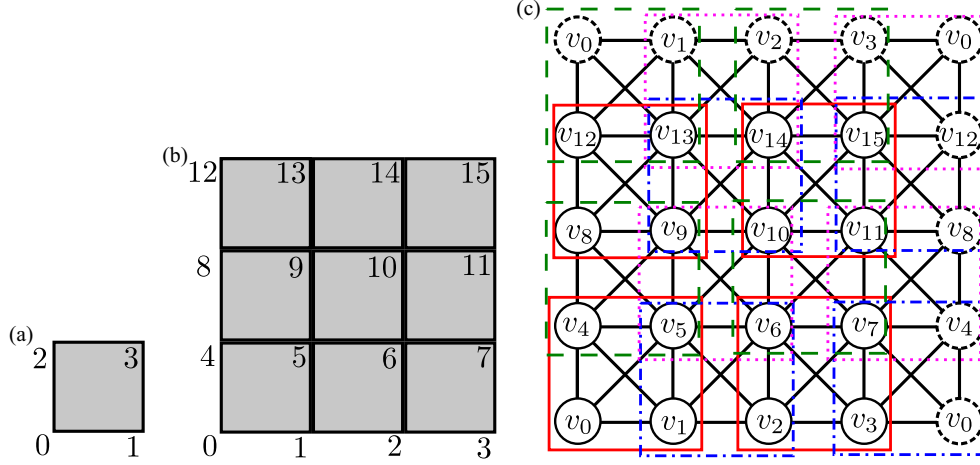
FIG. 10. Example of two-dimensional finite elements and the corresponding graph: (a) an element with node numbers, (b) a 3×3 two-dimensional square mesh, (c) a graph corresponding to the two-dimensional mesh. Rectangles drawn by red lines, blue dash-dotted lines, green dashed lines, and magenta dotted lines indicate separate tessellations.

Figure 9 illustrates an example of these tessellations for an 8-node cycle graph, where the red line represents a tessellation $\mathcal{T}_{\text{even}}$ and the blue dash-dotted line represents the other tessellation $\mathcal{T}_{\text{odd}}$.

The decomposed matrices $A_{\mathcal{T}_{\text{even}}}$ and $A_{\mathcal{T}_{\text{odd}}}$ in Eqs. (19) and (20) can then be expressed as the sums of the element stiffness matrices related to elements in each tessellation $\mathcal{T}_{\text{even}}$ and $\mathcal{T}_{\text{odd}}$, respectively.

The above discussion can easily be extended to two-dimensional problems. Let us consider the finite element method in the two-dimensional Poisson equation. For a first-order quadrilateral element of length 1 in Fig. 10(a), the element stiffness matrix is described as

$$
A_{\text{e}} := \frac{1}{6}\begin{bmatrix} 4 & -1 & -1 & -2 \\ -1 & 4 & -2 & -1 \\ -1 & -2 & 4 & -1 \\ -2 & -1 & -1 & 4 \end{bmatrix}
$$

$$
= \frac{1}{6}(4I \otimes I - I \otimes X - X \otimes I - 2X \otimes X). \tag{A3}
$$

Now, let us describe the total stiffness matrix for the mesh in Fig. 10(b) using the Pauli operators. Here, for simplicity, we assume that periodic boundary conditions are imposed on all edges of the mesh. Let N_x and N_y denote the numbers of columns and rows of nodes, respectively. That is, the number of nodes is $N = N_x \times N_y$. For example, in Fig. 10(b), $N_x = 4$ and $N_y = 4$.

We now define a graph corresponding to the mesh, as shown in Fig. 10(c). Each node v_i of the graph corresponds to node i of the mesh, and the graph has edges between nodes within the same elements. Because of the periodic boundary conditions, nodes corresponding to the edges of the mesh are also connected, e.g., v_0 and v_3, v_0 and v_{12}, and so on. For clear visibility, nodes with dashed circles are added on the upper and right sides.

Assuming that $N_x = 2^{n_x}$ and $N_y = 2^{n_y}$, four tessellations in the tessellation cover of the N-node graph can be defined as

$$
\mathcal{T}_0 := \left\{ C_{2i_x + 2i_y N_x} \mid i_x \in [0, 2^{n_x - 1} - 1], i_y \in [0, 2^{n_y - 1} - 1] \right\}, \tag{A4}
$$

$$
\mathcal{T}_1 := \left\{ C_{2i_x + 1 + 2i_y N_x} \mid i_x \in [0, 2^{n_x - 1} - 1], i_y \in [0, 2^{n_y - 1} - 1] \right\}, \tag{A5}
$$

$$
\mathcal{T}_2 := \left\{ C_{2i_x + (2i_y + 1)N_x} \mid i_x \in [0, 2^{n_x - 1} - 1], i_y \in [0, 2^{n_y - 1} - 1] \right\}, \tag{A6}
$$

$$
\mathcal{T}_3 := \left\{ C_{2i_x + 1 + (2i_y + 1)N_x} \mid i_x \in [0, 2^{n_x - 1} - 1], i_y \in [0, 2^{n_y - 1} - 1] \right\}, \tag{A7}
$$

where $C_i := \{v_i, v_{i+1}, v_{N_x}, v_{N_x+1}\}$ is the ith clique defined on the graph. Note that we define $v_{i \geqslant N} := v_{i \bmod N}$ to simplify the notation. In Fig. 10(c), rectangles with red lines, blue dash-dotted lines, green dashed lines, and magenta dotted lines represent $\mathcal{T}_0$, $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$, respectively.

Let $|i\rangle$ be the quantum state corresponding to node v_i of the graph. The quantum state $|i\rangle$ consists of two quantum registers, $|i_x\rangle$ and $|i_y\rangle$, for each direction, as $|i\rangle := |i_y\rangle|i_x\rangle$, where $|i_x\rangle$ and $|i_y\rangle$ consist of n_x and n_y qubits, respectively. The sum of the element stiffness matrix for the elements related to the tessellation $\mathcal{T}_0$, denoted as $A_{\mathcal{T}_0}$, can then be expressed as

$$
A_{\mathcal{T}_0} = \frac{1}{6}[4(I^{\otimes n_y - 1} \otimes I) \otimes (I^{\otimes n_x - 1} \otimes I) - (I^{\otimes n_y - 1} \otimes I) \otimes (I^{\otimes n_x - 1} \otimes X)
$$
$$
- (I^{\otimes n_y - 1} \otimes X) \otimes (I^{\otimes n_x - 1} \otimes I) - 2(I^{\otimes n_y - 1} \otimes X) \otimes (I^{\otimes n_x - 1} \otimes X)]
$$
$$
= \frac{1}{6}[4I^{\otimes (n_x + n_y)} - I^{\otimes n_y} \otimes (I^{\otimes n_x - 1} \otimes X) - (I^{\otimes n_y - 1} \otimes X) \otimes I^{\otimes n_x} - 2(I^{\otimes n_y - 1} \otimes X) \otimes (I^{\otimes n_x - 1} \otimes X)], \tag{A8}
$$

where the first n_y tensor products are for the y direction (row direction) and the latter n_x tensor products are for the x direction (column direction). As the nodes of the cliques in the tessellation $\mathcal{T}_1$ can be expressed by adding 1 to the node numbers of the nodes of cliques in the tessellation $\mathcal{T}_0$, the sum of the element stiffness matrix for the elements related to the tessellation $\mathcal{T}_1$, denoted as $A_{\mathcal{T}_1}$, can be described as follows:

$$A_{\mathcal{T}_1} = P_x^{-1} A_{\mathcal{T}_0} P_x, \tag{A9}$$

where P_x is a shift operator in the x direction defined as

$$P_x := \sum_{\substack{i_x \in [0, 2^{n_x} - 1] \\ i_y \in [0, 2^{n_y} - 1]}} |i_y\rangle |(i_x + 1) \bmod 2^{n_x}\rangle \langle i_y| \langle i_x|. \tag{A10}$$

Similarly, the sum of the element stiffness matrix for the elements related to the tessellations $\mathcal{T}_2$ and $\mathcal{T}_3$, denoted as $A_{\mathcal{T}_2}$ and $A_{\mathcal{T}_3}$, respectively, can be described as

$$A_{\mathcal{T}_2} = P_y^{-1} A_{\mathcal{T}_0} P_y, \quad A_{\mathcal{T}_3} = P_x^{-1} P_y^{-1} A_{\mathcal{T}_0} P_x P_y, \tag{A11}$$

where P_y is a shift operator in the y direction defined as

$$P_y := \sum_{\substack{i_x \in [0, 2^{n_x} - 1] \\ i_y \in [0, 2^{n_y} - 1]}} |(i_y + 1) \bmod 2^{n_y}\rangle |i_x\rangle \langle i_y| \langle i_x|. \tag{A12}$$

Consequently, the total stiffness matrix A can be described as the sum of the stiffness matrices related to each tessellation:

$$A = A_{\mathcal{T}_0} + A_{\mathcal{T}_1} + A_{\mathcal{T}_2} + A_{\mathcal{T}_3}. \tag{A13}$$

For Dirichlet and Neumann boundary conditions, we just have to add terms to adjust the stiffness matrices of the edge elements.

APPENDIX B: DERIVATION OF THE MEAN SQUARED ERROR BETWEEN THE EXACT COST FUNCTION AND THAT ESTIMATED BY SAMPLING

Here, the mean squared error between the exact cost function value and that estimated by sampling is derived using the Taylor series expansion. The first-order Taylor series expansion of $g(\bar{q}_1, \ldots, \bar{q}_m)$ around μ_i for $i \in [1, m]$ is given as

$$g(\bar{q}_1, \ldots, \bar{q}_m) = E_h + \sum_{i=1}^{m} \frac{\partial g}{\partial \bar{q}_i}\bigg|_{\bar{q}_i = \mu_i} (\bar{q}_i - \mu_i) + o[(\bar{q}_i - \mu_i)^2], \tag{B1}$$

where

$$g(\bar{q}_1, \ldots, \bar{q}_m) = -\frac{1}{2} \frac{\bar{q}_1^2}{\sum_{i=2}^{m} \bar{q}_i}. \tag{B2}$$

Assuming that $\mathrm{Cov}(\bar{q}_i, \bar{q}_{i'}) = 0$ for $i \neq i'$, the mean squared error between the exact cost function value and that estimated by sampling can be evaluated as follows:

$$\varepsilon^2 = \mathbb{E}[(g - E_h)^2] \approx \mathbb{E}\left\{ \left[\sum_{i=1}^{m} \frac{\partial g}{\partial \bar{q}_i}\bigg|_{\bar{q}_i = \mu_i} (\bar{q}_i - \mu_i) \right]^2 \right\}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{\partial g}{\partial \bar{q}_i}\bigg|_{\bar{q}_i = \mu_i} \frac{\partial g}{\partial \bar{q}_j}\bigg|_{\bar{q}_j = \mu_j} \mathbb{E}[(\bar{q}_i - \mu_i)(\bar{q}_j - \mu_j)] = \sum_{i=1}^{m} \left(\frac{\partial g}{\partial \bar{q}_i}\bigg|_{\bar{q}_i = \mu_i} \right)^2 \frac{\sigma_i^2}{S_i} \quad [\because \mathrm{Cov}(\bar{q}_i, \bar{q}_j) = 0]$$

$$= \frac{\mu_1^2}{\left(\sum_{i=2}^{m} \mu_i \right)^2} \frac{\sigma_1^2}{S_1} + \frac{1}{4} \frac{\mu_1^4}{\left(\sum_{i=2}^{m} \mu_i \right)^4} \sum_{i=2}^{m} \frac{\sigma_i^2}{S_i} = \frac{\mu_1^2}{\left(\sum_{i=2}^{m} \mu_i \right)^2} \left[\frac{\sigma_1^2}{S_1} + \frac{1}{4} \frac{\mu_1^2}{\left(\sum_{i=2}^{m} \mu_i \right)^2} \sum_{i=2}^{m} \frac{\sigma_i^2}{S_i} \right] = r_{\mathrm{opt}}^2 \left(\frac{\sigma_1^2}{S_1} + \frac{1}{4} r_{\mathrm{opt}}^2 \sum_{i=2}^{m} \frac{\sigma_i^2}{S_i} \right), \tag{B3}$$

where δ_{ij} is Kronecker's delta. The assumption that $\mathrm{Cov}(\bar{q}_i, \bar{q}_{i'}) = 0$ for $i \neq i'$ is based on the assumption that in quantum computers, each shot is mutually independent. In the last transformation, we have used the following equation:

$$r_{\mathrm{opt}} = \frac{\langle f, \psi | X \otimes I^{\otimes n} | f, \psi \rangle}{\langle \psi | A | \psi \rangle} = \frac{\mu_1}{\sum_{i=2}^{m} \mu_i}. \tag{B4}$$

APPENDIX C: DERIVATIVE OF THE COST FUNCTION

The gradient of the cost function in Eq. (14) is now derived. The partial derivative of the cost function with respect to the parameters $\boldsymbol{\theta}$ yields

$$\frac{\partial E_h}{\partial \boldsymbol{\theta}} = -\frac{[\langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle] \frac{\partial}{\partial \boldsymbol{\theta}} \langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle}{\langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle}$$
$$+ \frac{1}{2} \frac{[\langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle]^2 \frac{\partial}{\partial \boldsymbol{\theta}} \langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle}{\langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle^2}. \tag{C1}$$

Recalling that $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes n}$, where $U(\boldsymbol{\theta})$ is a sequence of parameterized quantum gates, the following holds for the ith parameter θ_i:

$$\frac{\partial}{\partial \theta_i} |\psi(\boldsymbol{\theta})\rangle = \frac{\partial}{\partial \theta_i} U(\boldsymbol{\theta})|0\rangle^{\otimes n} = \frac{1}{2} U(\theta_1, \theta_2, \dots, \theta_i + \pi, \dots)|0\rangle^{\otimes n}, \tag{C2}$$

under the assumption that the parameterized gates consist of either R_X, R_Y, or R_Z gates. Note that

$$|\psi(\boldsymbol{\theta})_{,i}\rangle := U(\theta_1, \theta_2, \dots, \theta_i + \pi, \dots)|0\rangle^{\otimes n} \tag{C3}$$

is a quantum state because $U(\theta_1, \theta_2, \dots, \theta_i + \pi, \dots)$ is a unitary operator.

Now, recall that $|f, \psi(\boldsymbol{\theta})\rangle := [|0\rangle|f\rangle + |1\rangle|\psi(\boldsymbol{\theta})\rangle]/\sqrt{2}$, which yields

$$\frac{\partial}{\partial \theta_i} \langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle = \frac{1}{4} [\langle \psi(\boldsymbol{\theta})_{,i} | f \rangle + \langle f | \psi(\boldsymbol{\theta})_{,i} \rangle] = \frac{1}{2} \langle f, \psi(\boldsymbol{\theta})_{,i} | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta})_{,i} \rangle, \tag{C4}$$

where $|f, \psi(\boldsymbol{\theta})_{,i}\rangle := [|0\rangle|f\rangle + |1\rangle|\psi(\boldsymbol{\theta})_{,i}\rangle]/\sqrt{2}$. The following equation also holds:

$$\frac{\partial}{\partial \theta_i} \langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle = \frac{1}{2} [\langle \psi(\boldsymbol{\theta})_{,i} | A | \psi(\boldsymbol{\theta}) \rangle + \langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta})_{,i} \rangle] = \langle \psi(\boldsymbol{\theta})_{,i}, \psi(\boldsymbol{\theta}) | X \otimes A | \psi(\boldsymbol{\theta})_{,i}, \psi(\boldsymbol{\theta}) \rangle, \tag{C5}$$

where $|\psi(\boldsymbol{\theta})_{,i}, \psi(\boldsymbol{\theta})\rangle := [|0\rangle|\psi(\boldsymbol{\theta})_{,i}\rangle + |1\rangle|\psi(\boldsymbol{\theta})\rangle]/\sqrt{2}$.
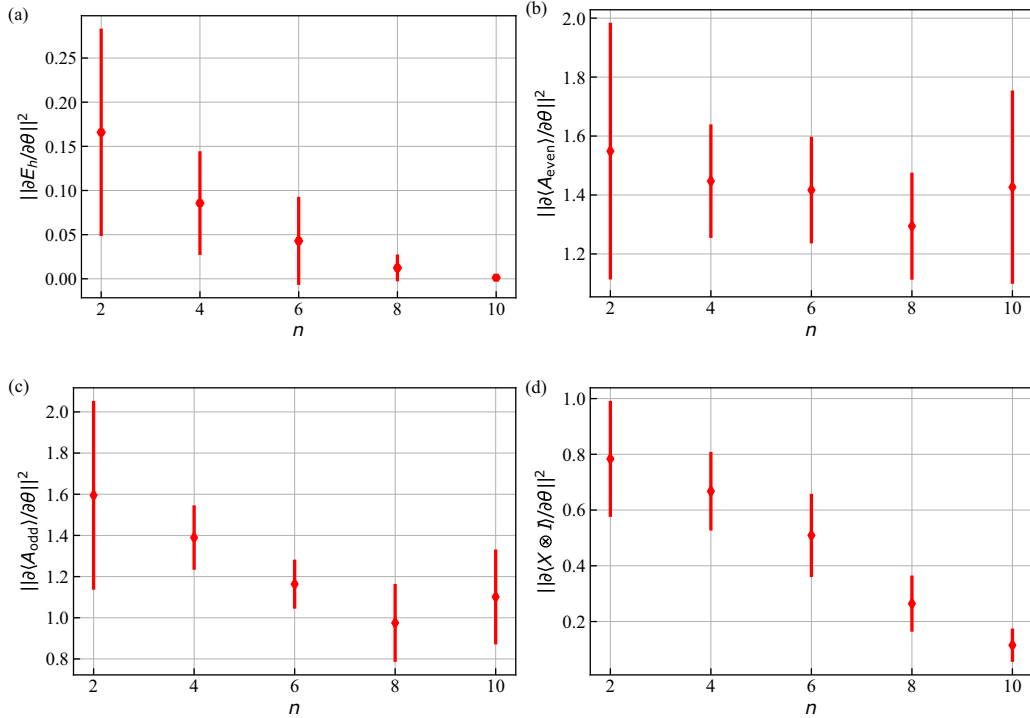

FIG. 11. Norm of gradients of the cost function and those of each term composing the cost function. The points show the mean values of 10 experiments from varying initial parameters and the error bars represent the standard deviation. (a) Norm of $\partial E_h / \partial \boldsymbol{\theta}$. (b) Norm of $\partial \langle A_{\text{even}} \rangle / \partial \boldsymbol{\theta}$, where $\langle A_{\text{even}} \rangle := \langle \psi(\boldsymbol{\theta}) | A_{\mathcal{T}_{\text{even}}} | \psi(\boldsymbol{\theta}) \rangle$. (c) Norm of $\partial \langle A_{\text{odd}} \rangle / \partial \boldsymbol{\theta}$, where $\langle A_{\text{odd}} \rangle := \langle \psi(\boldsymbol{\theta}) | A_{\mathcal{T}_{\text{odd}}} | \psi(\boldsymbol{\theta}) \rangle$. (d) Norm of $\partial \langle X \otimes I^{\otimes n} \rangle / \partial \boldsymbol{\theta}$, where $\langle X \otimes I \rangle := \langle f, \psi(\boldsymbol{\theta}) | X \otimes I | f, \psi(\boldsymbol{\theta}) \rangle$.
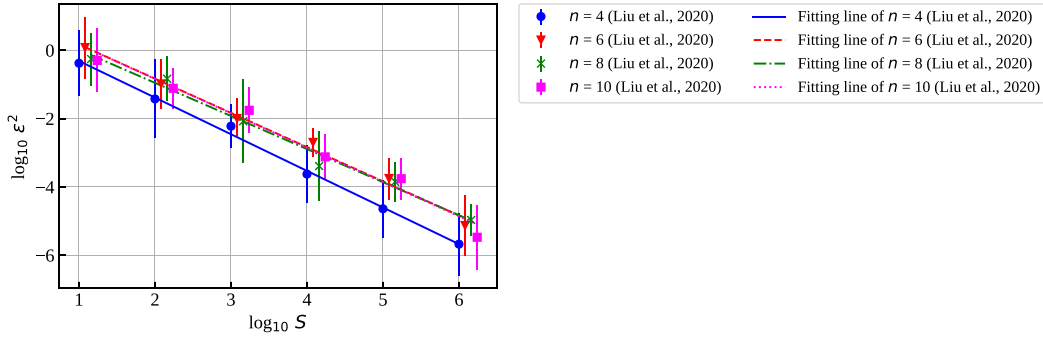
FIG. 12. Decadic logarithms of cost function error ε vs the number of shots S for the previous method [26] based on our implementation. The optimization was run 10 times with fixed parameters for each number of shots. The mean values are plotted, with the error bars representing the standard deviations. The mean and standard deviation of the slopes of the fitted dashed line for $n = 2, \ldots, 10$ (which includes cases that are not shown in this figure) are -1.0 and 0.05, respectively.

Substituting Eqs. (C4) and (C5) into Eq. (C1), the gradient of the cost function is derived as

$$
\frac{\partial E_h}{\partial \theta_i} = -\frac{1}{2} \frac{[\langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle] \langle f, \psi(\boldsymbol{\theta})_{,i} | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta})_{,i} \rangle}{\langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle}
$$
$$
+ \frac{1}{2} \frac{[\langle f, \psi(\boldsymbol{\theta}) | X \otimes I^{\otimes n} | f, \psi(\boldsymbol{\theta}) \rangle]^2 \langle \psi(\boldsymbol{\theta})_{,i}, \psi(\boldsymbol{\theta}) | X \otimes A | \psi(\boldsymbol{\theta})_{,i}, \psi(\boldsymbol{\theta}) \rangle}{\langle \psi(\boldsymbol{\theta}) | A | \psi(\boldsymbol{\theta}) \rangle^2}. \tag{C6}
$$

Now, as all components of the gradient can be described as the expectations of observables, the gradient can be evaluated by quantum computers.

APPENDIX D: SUPPLEMENTARY RESULTS OF NUMERICAL EXPERIMENTS

1. Barren plateaus

We examined the vanishing gradients of the cost function. Figure 11 illustrates the L2-norm of the gradients of the cost function and those of each term composing the cost function, i.e., $\langle \psi(\boldsymbol{\theta}) | A_{\mathcal{T}_{\mathrm{even}}} | \psi(\boldsymbol{\theta}) \rangle$, $\langle \psi(\boldsymbol{\theta}) | A_{\mathcal{T}_{\mathrm{odd}}} | \psi(\boldsymbol{\theta}) \rangle$, and $\langle f, \psi(\boldsymbol{\theta}) | X \otimes I | f, \psi(\boldsymbol{\theta}) \rangle$. These gradients were calculated using the Statevector Simulator. The number of layers of the ansatz was set to 5. The points show the mean values of 10 experiments with different randomly set parameters and the error bars represent the standard deviations. As shown in these figures, the gradient of the expectation of the operator $A_{\mathcal{T}_{\mathrm{even}}}$, which is local, does not vanish, while the gradients of the other terms, which are global, vanish. As a result, the gradient of the cost function as a whole vanishes. The alleviation of barren plateaus will be addressed in future research.

2. Dependency of the cost function error on the number of shots for the previous method

Figure 12 shows the decadic logarithm of the squared error of the cost function with respect to the decadic logarithm of the number of shots for the previous method [26]. In a similar fashion to our proposed method in Fig. 7, the squared error decreases as the number of shots increases. The mean and standard deviation of the slopes of the fitted lines are -1.01 and 0.04, respectively. Although a comparison of the proposed and previous methods from these figures is difficult because of the different definitions of the cost function, it can be deduced that the mean squared error in the cost function evaluation has a similar dependency on the number of shots in both methods.

[1] M. Renardy and R. C. Rogers, *An Introduction to Partial Differential Equations* (Springer-Verlag, New York, 2004), Vol. 13.

[2] A. Klawonn, M. Lanser, and O. Rheinbach, SIAM J. Sci. Comput. **37**, C667 (2015).

[3] J. Toivanen, P. Avery, and C. Farhat, Intl. J. Numer. Methods Eng. **116**, 661 (2018).

[4] C. Kato, Y. Yamade, K. Nagano, K. Kumahata, K. Minami, and T. Nishikawa, in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, GA, USA* (IEEE, Piscataway, NJ, 2020), pp. 1–13.

[5] K. Fujita, K. Koyama, K. Minami, H. Inoue, S. Nishizawa, M. Tsuji, T. Nishiki, T. Ichimura, M. Hori, and L. Maddegedara, J. Comput. Sci. **49**, 101277 (2021).

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

[7] A. M. Childs, R. Kothari, and R. D. Somma, SIAM J. Comput. **46**, 1920 (2017).

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

[9] D. J. Griffiths, *Introduction to Electrodynamics* (Pearson Addison Wesley, Upper Saddle River, 1999).

[10] T. Chung, *Computational Fluid Dynamics* (Cambridge University Press, Cambridge, 2010).

[11] J. Blazek, *Computational Fluid Dynamics: Principles and Applications* (Butterworth-Heinemann, Oxford, 2015).

[12] J. Preskill, Quantum **2**, 79 (2018).

[13] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Nat. Rev. Phys. **3**, 1 (2021).

[14] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, Nat. Commun. **5**, 4213 (2014).

[15] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, arXiv:2103.08505 (2021).

[16] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature (London) **549**, 242 (2017).

[17] M. Ganzhorn, D. J. Egger, P. Barkoutsos, P. Ollitrault, G. Salis, N. Moll, M. Roth, A. Fuhrer, P. Mueller, S. Woerner, I. Tavernelli, and S. Filipp, Phys. Rev. Appl. **11**, 044092 (2019).

[18] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, Phys. Rev. A **99**, 062304 (2019).

[19] O. Higgott, D. Wang, and S. Brierley, Quantum **3**, 156 (2019).

[20] Z.-H. Yuan, T. Yin, and D.-B. Zhang, Phys. Rev. A **103**, 012413 (2021).

[21] E. Farhi, J. Goldstone, and S. Gutmann, arXiv:1411.4028.

[22] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, arXiv:quant-ph/0001106.

[23] A. M. Childs, E. Farhi, and J. Preskill, Phys. Rev. A **65**, 012322 (2001).

[24] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, Algorithms **12**, 34 (2019).

[25] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. Coles, Bull. Am. Phys. Soc. **65**, F07.00005 (2020).

[26] H.-L. Liu, Y.-S. Wu, L.-C. Wan, S.-J. Pan, S.-J. Qin, F. Gao, and Q.-Y. Wen, Phys. Rev. A **104**, 022418 (2021).

[27] T. J. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis* (Courier Corporation, North Chelmsford, MA, 2012).

[28] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, Quantum Inf. Comput. **5**, 467 (2005).

[29] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, Phys. Rev. A **93**, 032318 (2016).

[30] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, Sci. Rep. **11**, 6329 (2021).

[31] D. Maslov, Phys. Rev. A **93**, 022311 (2016).

[32] S. Oomura, T. Satoh, M. Sugawara, and N. Yamamoto, arXiv:2102.06117.

[33] Y. Saad, *Iterative Methods for Sparse Linear Systems* (SIAM, Philadelphia, 2003).

[34] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Nat. Commun. **12**, 1 (2021).

[35] H. Abraham, AduOffei, R. Agarwal, G. Agliardi, M. Aharoni, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, M. Amy, S. Anagolum *et al.*, QISKIT: An Open-source Framework for Quantum Computing (2019), https://github.com/Qiskit/qiskit.

[36] C. G. Broyden, IMA J. Appl. Math. **6**, 222 (1970).

[37] R. Fletcher, Comput. J. **13**, 317 (1970).

[38] D. Goldfarb, Math. Comput. **24**, 23 (1970).

[39] D. F. Shanno, Math. Comput. **24**, 647 (1970).

[40] A. Abreu, L. Cunha, T. Fernandes, C. de Figueiredo, L. Kowada, F. Marquezino, D. Posner, and R. Portugal, in *Latin American Symposium on Theoretical Informatics* (Springer, New York, 2018), pp. 1–13.