IEEE Transactions on Quantum Engineering

Received 5 August 2022; revised 11 April 2023; accepted 17 April 2023; date of publication 28 April 2023; date of current version 21 June 2023.

Digital Object Identifier 10.1109/TQE.2023.3269525

Pricing Multiasset Derivatives by Variational Quantum Algorithms

KENJI KUBO[1,2], KOICHI MIYAMOTO[3], KOSUKE MITARAI[2,3,4], AND KEISUKE FUJII[2,3,5]

[1]Mercari R4D, Mercari, Inc., Tokyo 106-6118, Japan
[2]Graduate School of Engineering Science, Osaka University, Osaka 560-8531, Japan
[3]Center for Quantum Information and Quantum Biology, Osaka University, Osaka 560-0043, Japan
[4]Japan Science and Technology Agency (JST), Precursory Research for Embryonic Science and Technology (PRESTO), Saitama 332-0012, Japan
[5]RIKEN Center for Quantum Computing, Wako Saitama 351-0198, Japan

Corresponding author: Kenji Kubo (e-mail: kenji.kubo.q@gmail.com).

The work of Koichi Miyamoto was supported by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under Grant 22K11924. The work of Kosuke Mitarai was supported in part by the Japan Science and Technology Agency (JST) Precursory Research for Embryonic Science and Technology (PRESTO) under Grant JPMJPR2019 and in part by JSPS KAKENHI under Grant 20K22330. This work was supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) Quantum Leap Flagship Program (QLEAP) under Grant JPMXS0118067394 and Grant JPMXS0120319794, and in part by JST Center of Innovation (COI)-NEXT Program.

ABSTRACT Pricing a multiasset derivative is an important problem in financial engineering, both theoretically and practically. Although it is suitable to numerically solve partial differential equations to calculate the prices of certain types of derivatives, the computational complexity increases exponentially as the number of underlying assets increases in some classical methods, such as the finite difference method. Therefore, there are efforts to reduce the computational complexity by using quantum computation. However, when solving with naive quantum algorithms, the target derivative price is embedded in the amplitude of one basis of the quantum state, and so an exponential complexity is required to obtain the solution. To avoid the bottleneck, our previous study utilizes the fact that the present price of a derivative can be obtained by its discounted expected value at any future point in time and shows that the quantum algorithm can reduce the complexity. In this article, to make the algorithm feasible to run on a small quantum computer, we use variational quantum simulation to solve the Black–Scholes equation and compute the derivative price from the inner product between the solution and a probability distribution. This avoids the measurement bottleneck of the naive approach and would provide quantum speedup even in noisy quantum computers. We also conduct numerical experiments to validate our method. Our method will be an important breakthrough in derivative pricing using small-scale quantum computers.

INDEX TERMS Derivative pricing, finite difference methods (FDMs), variational quantum computing.

I. INTRODUCTION

Quantum computers actively utilize quantum phenomena to solve large-scale problems that could not be performed with the conventional classical computers. In recent years, applications of quantum computers have been discussed in financial engineering. Specifically, the applications include portfolio optimization [1], [2], [3], risk measurement [4], [5], [6], [7], [8], and derivative pricing [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]. Comprehensive reviews of these topics are presented in [23], [24], [25], and [26].

Among these applications, we consider the pricing of derivatives. Derivatives are the products that refer to the prices of underlying assets, such as stocks, bonds, and currencies, and their payoff depends on the prices of the assets. For example, a European call option, one of the simplest derivatives, has a predetermined maturity $T > 0$ and strike price K, and its holder gets paid back $\max(S(T) - K, 0)$ for the asset price $S(T)$ at T. For such a simple derivative, the theoretical price can be computed analytically in some models, such as the Black–Scholes (BS) model [27]. If one wishes to calculate prices for derivatives with more complex payoffs, numerical calculations are required [28].

There are many algorithms for numerical calculations. For the pricing of certain types of derivatives, such as barrier

options, it is suitable to solve the partial differential equations (PDEs) called Black–Scholes PDE (BSPDE) [29], [30] by discretizing them using the finite difference method (FDM). However, in the case of multiasset derivatives, the number of grid points increases exponentially with respect to the number of referenced assets, making price calculation difficult. When the number of assets is d and the number of grid points is n_{gr} for one asset, the total number of grid points is n_{gr}^d. If we take n_{gr} in proportion to $\epsilon^{-1/2}$ to achieve the error level ϵ (see Lemma II.1 in [31]), classical FDM requires the computational complexity of $O((1/\epsilon)^{O(d)})$ in terms of time and space.[1,2]

To overcome this difficulty, several methods [13], [14], [15], [22] have been proposed to efficiently solve the BSPDE using quantum computers. However, when solving the discretized BSPDE with these quantum algorithms, the target derivative price is embedded in the amplitude of one basis of the resulting quantum state, so it requires exponentially large computational complexity to extract it as classical information. Miyamoto and Kubo [31] have shown that the complexity can be substantially reduced using the fact that the present derivative price can be calculated as the expected value of the discounted derivative price at a future point in time. They calculate the inner product of the state in which the future derivative prices are embedded and the state in which the probability distribution is embedded using the quantum amplitude estimation (QAE) [37]. Instead of retrieving one of the amplitudes of the output state of the quantum algorithm, the present price of the derivative can be efficiently calculated since all of the amplitudes can be used. In fact, the complexity of the method proposed in [31] does not have a factor, such as $(1/\epsilon)^{O(d)}$, but has only $\mathrm{poly}(1/\epsilon, d)$. This means that their method has substantial speedup compared with the classical FDM. Furthermore, their method achieves exponential reduction of space complexity since function values on $(1/\epsilon)^{O(d)}$ grid points can be encoded in amplitudes of a $O(d\log(1/\epsilon))$-qubit state.

However, it should be noted that their method is constructed on the quantum ordinary differential equation (ODE) solver [38] and the QAE, which requires a large-scale quantum computer with error correction. In addition, it is assumed that we are given the oracle that generates a quantum state in which the boundary conditions of the BSPDE are encoded in amplitudes. As the derivatives are currently dealt with in practice, it is desirable to calculate derivative prices even with a small-scale quantum computer closer to realization.

In this article, we propose a variational quantum algorithm for pricing multiasset derivatives. We show that the SWAP test can be used to avoid the problem of retrieving the derivative price from the quantum state. This is the way to exploit the essential feature proposed in [31] with variational quantum algorithms and, hence, thought to work with near-term quantum computers. Our algorithm has the following three parts; embedding the probability distribution of the underlying asset prices into the quantum state, solving the BSPDE with boundary conditions, and calculating the inner product. For the first part, we can use the quantum generative algorithms [39], [40], [41], [42], [43] or variational quantum simulation (VQS) for the Fokker–Planck equations [22], [44], [45], [46], [47], which describe the time evolution of the probability density functions of the stochastic processes. For the second part, we discretize the BSPDE using the FDM and solve it using VQS. For the third part, we evaluate the square of the inner product of the states, obtained by the first and the second parts of our method, using the SWAP test [48]. Taking the square root of the output of the SWAP test and discounting by the interest rate, we obtain the present price of the derivative. Although there is no guarantee of overall computational complexity due to the heuristic nature of the variational algorithm, we show that the number of measurements of the SWAP test has no factors, such as $(1/\epsilon)^{O(d)}$, which means that our method can avoid the bottleneck of retrieving derivative prices from the quantum state. Since our algorithm requires quantum circuits with $O(d\log(1/\epsilon))$ qubits and $O(\mathrm{poly}(d\log(1/\epsilon))$ few-qubit gates, even a small-scale quantum computer would be able to perform derivatives pricing with our method. We perform numerical calculations for a single asset double barrier option and confirm that our method is feasible.

The rest of this article is organized as follows. Section II is the preliminary section. Our formulations are slightly different from previous studies, such as [22] and [31]. We describe the detailed formulation in this section. The notations in this article are listed in Section II-A. We summarize the related works in Section II-B. In Section II-C, we introduce derivative pricing using the BSPDE with boundary conditions. We also introduce FDM to discretize the BSPDE and obtain an ODE in Section II-D. Section II-F gives an introduction to VQS, which is an algorithm for solving the ODE. Section II-E introduces the fact that the present price of the derivative can be approximated by the expected value of the future price. In Section III, we describe the proposed method. We estimate the number of measurements required by the SWAP test in Section III-A and the whole time complexity of the proposed method in Section III-B. We show the feasibility of our method through numerical simulations in Section IV. Finally, Section V concludes this article.

[1] Although the classical Monte Carlo method can avoid this issue, there would be other issues, such as hitting time errors [32]. Both the classical Monte Carlo method and the classical PDE approach have been studied for a long time, and each has its own advantages. We focus only on the FDM approach in this article, and the comparison between the FDM approach and the Monte Carlo method is not the subject of our study.

[2] As listed in [33], there are some classical algorithms, such as adaptive FDM/FEM [34], spectral method [35], and sparse grid FDM/FEM [36], that have $O(\mathrm{poly}((\log(1/\epsilon))^d))$ time complexity and $\Omega((\log(1/\epsilon))^d)$ space complexity under some conditions. While these methods may be able to calculate option prices faster than regular FDM, a detailed discussion of the applicability of these methods is not the subject of this study.

IEEE Transactions on Quantum Engineering

II. PRELIMINARY

A. NOTATION

Here, we introduce the notation used in this article. We define $\mathbb{R}_+$ as a set of all positive real numbers, and for a positive integer d, $\mathbb{R}_+^d$ as a d-times direct product of $\mathbb{R}_+$. For a positive integer n, $[n] := \{1, 2, \ldots, n\}$. For $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)^\top \in \mathbb{R}^n$, where n is an integer not less than 2, and $i \in [n]$, we define $\boldsymbol{v}_{\wedge i} \in \mathbb{R}^{n-1}$ as a vector, which is made by removing an element v_i from $\boldsymbol{v}$, that is, $\boldsymbol{v}_{\wedge i} := (v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$. We define the Euclidean norm for a vector $\boldsymbol{v}$ as $\|\boldsymbol{v}\| = \sqrt{\sum_i v_i^2}$. For an integer i, we define $|i\rangle$ as one of the computational basis states with a binary representation of i and for a vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top \in \mathbb{C}^d$, we denote $|\boldsymbol{y}\rangle$ as an unnormalized state where the elements of $\boldsymbol{y}$ are encoded in the amplitudes, that is, $|\boldsymbol{y}\rangle := \sum_{i=1}^n y_i|i\rangle$.

B. RELATED WORK

In this section, we explain the existing algorithms for solving the BSPDE with quantum computers. Gonzalez-Conde et al. [15] transform the BSPDE into a Schrödinger equation, discretize the Hamiltonian by FDM, and solves it by diagonalization of discretized momentum operator with a quantum Fourier transformation. Fontanela et al. [13] and Radha [14] solve the discretized Schrödinger equation by VQS, which is a variational quantum algorithm for solving ODEs. Also, Alghassi et al. [22] propose a VQS to directly solve PDEs, including the BSPDE. In the previous studies mentioned above, the time complexity required to solve the BSPDE depends on the grid points only logarithmically. However, there is still a problem that cannot be overlooked; extracting the calculated result from quantum computers may take exponentially long time with respect to d as mentioned in [Appendix E in [22]]. Solving the BSPDE from the maturity ($t = T$) to the present ($t = 0$) with these quantum algorithms yields unnormalized state $|\boldsymbol{V}(0)\rangle$ whose elements are the derivative prices on the grid points of underlying asset prices. Note that, typically, we are interested in only one element of $|\boldsymbol{V}(0)\rangle$, the derivative price on the grid point corresponding to the present underlying asset prices. However, since $|\boldsymbol{V}(0)\rangle$ has $O((1/\epsilon)^{O(d)})$ elements, the amplitude corresponding to V_0 in (normalized) $|\boldsymbol{V}(0)\rangle$ is exponentially small. Therefore, the exponential time complexity is required to retrieve V_0 as classical information, and the quantum speedup will be lost.

Miyamoto and Kubo [31] show the algorithm to overcome the problem. They prepare the state $|\boldsymbol{p}(t_{\text{ter}})\rangle$ in which the probability distribution of underlying asset prices on the grid points at a certain time $t_{\text{ter}} \in [0, T]$ is embedded in the amplitudes. Then, they discretize the BSPDE using FDM, solve it not to $t = 0$ but $t = t_{\text{ter}}$ with quantum ODE solver, and obtain the state $|\boldsymbol{V}(t_{\text{ter}})\rangle$. The inner product of these quantum states, which can be obtained by QAE, corresponds to the expected value of the derivative price at t_{ter} by $E[V(t_{\text{ter}})] \simeq \sum_{i\in\mathcal{G}} p_i(t_{\text{ter}})V_i(t_{\text{ter}}) = \langle \boldsymbol{p}(t_{\text{ter}})|\boldsymbol{V}(t_{\text{ter}})\rangle$. Discounting this expected value by the risk-free interest rate yields the present price of the derivative [28].

Our algorithm is a variational version of the article presented in [31]. Instead of using the quantum ODE solver and QAE, we use VQS and the SWAP test, respectively. This enables derivatives pricing by BSPDE to be realized on a small-sized quantum computer.

C. DERIVATIVE PRICING

To evaluate the price of a derivative, we need to model the dynamics of the prices of the underlying asset. We adopt the BS model [27] in which the prices of the underlying assets are assumed to follow geometric Brownian motions. That is, we suppose that the prices of d underlying assets at $t \in [0, T]$ are stochastic processes $\boldsymbol{S}(t) = (S_1(t), S_2(t), \ldots, S_d(t))^\top \in \mathbb{R}_+^d$ that, under the risk-neutral measure, obey stochastic differential equations

$$dS_i(t) = rS_i(t)dt + \sigma_i S_i(t)dW_i(t). \tag{1}$$

Here, $r > 0$ is the risk-free interest rate and $\sigma_i > 0$ is the volatility of the ith underlying asset for each $i \in [d]$. $W_i(t)$ are the Brownian motions that satisfy $dW_i dW_j = \rho_{i,j}dt$, $(i, j) \in [d] \times [d]$ with the correlation matrix $(\rho_{ij})_{1\le i,j\le d}$, which satisfies $\rho_{i,i} = 1$ and $-1 < \rho_{i,j} = \rho_{j,i} < 1$ for $i \neq j$.

We assume that the derivatives are characterized by the payoff function f_{pay} at the maturity T and the payoff conditions, which must be satisfied in order for the payoff to arise. More specifically, the payoff condition is the binary condition of whether the underlying asset price process meets predetermined criteria during the period, which is formally regarded as a map taking an asset price path $\{\boldsymbol{S}(t)\}_{t\in[0,T]}$ to a binary. Note that there are derivatives that do not satisfy these assumptions. For example, American and Bermudan options cannot be characterized only by the payoff function at maturity and the payoff condition because there are multiple possible exercise times. Also, look-back options cannot be characterized only by the payoff function at maturity because the payoff function depends on the price of the underlying asset before maturity. However, for example, Asian options, whose payoff function depends on the price of the underlying asset prior to maturity, can be converted by introducing an additional random variable to make the payoff function refer only to the price of the asset at maturity [29], [30]. We describe the typical cases of the payoff functions and the payoff conditions later.

The price of the derivative is obtained as the conditional expected value of the payoff, conditioned on the price of the underlying assets, discounted by the risk-free rate [29], [30]. That is, given the underlying asset prices at time t as $\boldsymbol{s} = (s_1, \ldots, s_d)^\top \in \mathbb{R}_+^d$, and the payoff function at maturity T as $f_{\text{pay}}(\boldsymbol{S}(T))$, the price of the derivative is

$$V(t, \boldsymbol{s}) = E_Q\left[e^{-r(T-t)} f_{\text{pay}}(\boldsymbol{S}(T))\mathbb{1}_{\text{NB}}|\boldsymbol{S}(t) = \boldsymbol{s}\right] \tag{2}$$

where E_Q is the expected value under the so-called risk-neutral measure. Note that $\boldsymbol{S}(T)$ is a vector of random variables resulting from the time evolution of (1) from t to T with the condition $\boldsymbol{S}(t) = \boldsymbol{s}$. $\mathbb{1}_{\mathrm{NB}}$ is a random variable that takes 1 if the payoff conditions are satisfied or 0 otherwise.

The goal of derivative pricing is to find the present price of the derivative, that is, $V(0, \boldsymbol{s}_0)$, where $\boldsymbol{s}_0 = (s_{1,0}, \ldots, s_{d,0})^\top \in \mathbb{R}_+^d$ is the present price of the underlying assets. To this end, we use the BSPDE, which describes the time evolution of $V(t, \boldsymbol{s})$ [29], [30]. That is, the derivative price $V(t, \boldsymbol{s})$ is the solution of the BSPDE

$$\frac{\partial}{\partial t}V(t,\boldsymbol{s}) + \frac{1}{2}\sum_{i,j=1}^{d}\sigma_i\sigma_j s_i s_j \rho_{ij}\frac{\partial^2}{\partial s_i \partial s_j}V(t,\boldsymbol{s}) + r\left(\sum_{i=1}^{d} s_i \frac{\partial}{\partial s_i}V(t,\boldsymbol{s}) - V(t,\boldsymbol{s})\right) = 0 \tag{3}$$

on $[0, T) \times D$ with the boundary conditions

$$V(T,\boldsymbol{s}) = f_{\mathrm{pay}}(\boldsymbol{s}) \tag{4}$$

$$V\left(t, (s_1, \ldots, s_{i-1}, u_i, s_{i+1}, \ldots, s_d)^\top\right) =: V_i^{\mathrm{UB}}(t, \boldsymbol{s}_{\wedge i}), \text{ for } i \in [d] \tag{5}$$

$$V\left(t, (s_1, \ldots, s_{i-1}, l_i, s_{i+1}, \ldots, s_d)^\top\right) =: V_i^{\mathrm{LB}}(t, \boldsymbol{s}_{\wedge i}), \text{ for } i \in [d] \tag{6}$$

where u_i and l_i are the upper and lower bounds of the ith asset price, respectively, $D := (l_1, u_1) \times \cdots \times (l_d, u_d)$, and V_i^{UB} and V_i^{LB} are some functions on boundaries of D that represent the derivative price on the boundaries. Here, we assume that, according to the payoff function and the payoff condition, we can predetermine V_i^{UB} and V_i^{LB} as explicit formulae. Although it is not clear that this is possible for general payoff functions and payoff conditions, it is in fact possible in some cases, as we show typical examples in the following text.

1) If an *up and out barrier* is set on the ith asset, the payoff is zero if the asset price $S_i(t)$ exceeds u_i at least once before maturity, and then the boundary condition is

$$V_i^{\mathrm{UB}}(t, \boldsymbol{s}_{\wedge i}) = 0. \tag{7}$$

Similarly, if a *down and out barrier* is set on the ith asset, the payoff is zero if the asset price falls below l_i at least once before maturity, and then, the boundary condition is

$$V_i^{\mathrm{LB}}(t, \boldsymbol{s}_{\wedge i}) = 0. \tag{8}$$

2) Suppose that the payoff at maturity T is given by

$$f_{\mathrm{pay}}(\boldsymbol{S}(T)) = \max(a_0 + \sum_{i=1}^{d} a_i S_i(T), 0) \tag{9}$$

with $a_0, \ldots, a_d \in \mathbb{R}$. This is the case with many derivatives. In this form of payoff function, upper boundary or lower boundary can be set depending on the values of $a_0, \ldots, a_d$. In some cases, if any of $\{S_i(t)\}_{i\in[d]}$ is sufficiently high or low at some time $t \in (0, T)$, the payoff at T is highly likely to be positive. For example, in the case of the basket call option, that is, $a_0 < 0, a_1, \ldots, a_d > 0$, if $\boldsymbol{S}(t) = \boldsymbol{s}$ such that $s_i \gg -a_0/a_i$ for some $i \in [d]$, $f_{\mathrm{pay}}(\boldsymbol{S}(T))$ is likely to be positive. In this situation, the derivative price is approximately equal to $E_Q[e^{-r(T-t)}(a_0 + \sum_{i=1}^{d} a_i S_i(T))|\boldsymbol{S}(t) = \boldsymbol{s}] = e^{-r(T-t)}a_0 + \sum_{i=1}^{d} a_i s_i$. Thus, we can set

$$V_i^{\mathrm{UB}}(t, \boldsymbol{s}_{\wedge i}) = e^{-r(T-t)}a_0 + \sum_{1\le j\le d, j\neq i} a_j s_j + a_i u_i \tag{10}$$

for sufficiently large u_i. In some other cases, e.g., when $a_i < 0$ and $a_j > 0$ for $j \neq i$, we can set

$$V_i^{\mathrm{LB}}(t, \boldsymbol{s}_{\wedge i}) = e^{-r(T-t)}a_0 + \sum_{1\le j\le d, j\neq i} a_j s_j + a_i l_i \tag{11}$$

for sufficiently small l_i.

Note that the boundary conditions listed here do not cover all possible cases. Also note that (10) and (11) are different cases and need not be simultaneously satisfied.

D. FDM FOR THE BSPDE

Consider solving (3) using the FDM. In the FDM, we discretize the PDE with respect to the underlying asset prices and obtain the ODE. Then, we can use a numerical solver for ODEs, such as the Euler method and Runge–Kutta method [49]. Note that the BSPDE is often simplified by log transforming the asset prices as in [31]. However, it is more convenient not to perform a log transformation to solve the BSPDE by VQS. This is because our formulation presented in Section III can only handle linear boundary conditions with respect to s_i, as shown in Appendix B, but a logarithmic transformation will result in the terms, such as e^{s_i}. Thus, we do not perform the log transformation in this work.

First, the value range, excluding the boundary of each underlying asset price s_i, is split into n_{gr} grids. That is, we take

$$\boldsymbol{x}^{(k)} = \left(x_1^{(k_1)}, \ldots, x_d^{(k_d)}\right)^\top \tag{12}$$

$$x_i^{(k_i)} := l_i + (k_i + 1)\, h_i \tag{13}$$

$$k = \sum_{i=1}^{d} n_{\mathrm{gr}}^{d-i} k_i + 1 \tag{14}$$

$$k_i := 0, \ldots, n_{\mathrm{gr}} - 1 \tag{15}$$

$$h_i := \frac{u_i - l_i}{n_{\mathrm{gr}} + 1} \tag{16}$$

for $i \in [d]$. By this discretization, we approximate $V(t, \boldsymbol{s})$ by a vector

$$\boldsymbol{V}(t) := \left(V(t, \boldsymbol{x}^{(1)}), V(t, \boldsymbol{x}^{(2)}), \ldots, V\left(t, \boldsymbol{x}^{(N_{\mathrm{gr}})}\right)\right)^\top \tag{17}$$

IEEE Transactions on Quantum Engineering

where $N_{\rm gr} = n_{\rm gr}^d$. We also replace the differentials by differences as follows:

$$\frac{\partial V(t,\boldsymbol{x}^{(k)})}{\partial s_i} \to \frac{V(t,\boldsymbol{x}^{(k)}+h_i\boldsymbol{e}_i)-V(t,\boldsymbol{x}^{(k)}-h_i\boldsymbol{e}_i)}{2h_i} \tag{18}$$

$$\frac{\partial^2 V(t,\boldsymbol{x}^{(k)})}{\partial s_i^2} \to \frac{1}{h_i^2}\Big(V(t,\boldsymbol{x}^{(k)}+h_i\boldsymbol{e}_i)+V\left(t,\boldsymbol{x}^{(k)}-h_i\boldsymbol{e}_i\right) -2V(t,\boldsymbol{x}^{(k)})\Big) \tag{19}$$

$$\frac{\partial^2 V(t,\boldsymbol{x}^{(k)})}{\partial s_i \partial s_j} \to \frac{1}{4h_ih_j}\Big(V\left(t,\boldsymbol{x}^{(k)}+h_i\boldsymbol{e}_i+h_j\boldsymbol{e}_j\right) + V\left(t,\boldsymbol{x}^{(k)}-h_i\boldsymbol{e}_i-h_j\boldsymbol{e}_j\right) - V\left(t,\boldsymbol{x}^{(k)}+h_i\boldsymbol{e}_i-h_j\boldsymbol{e}_j\right) -V\left(t,\boldsymbol{x}^{(k)}-h_i\boldsymbol{e}_i+h_j\boldsymbol{e}_j\right)\Big) \tag{20}$$

where

$$\boldsymbol{e}_i = (\underbrace{0,\ldots,0}_{i-1},1,\underbrace{0,\ldots,0}_{d-i})^\top, i\in[d]$$

is a unit vector of the i-direction. We define $\tau := T-t$ for the convenience of the later notation. Introducing $\bar{V}(\tau,\boldsymbol{x}^{(i)}) := V(T-\tau,\boldsymbol{x}^{(i)}), i\in[d]$ and $\bar{\boldsymbol{V}}(\tau) := (\bar{V}(\tau,\boldsymbol{x}^{(1)}),\bar{V}(\tau,\boldsymbol{x}^{(2)}),\ldots,\bar{V}(\tau,\boldsymbol{x}^{(N_{\rm gr})}))^\top$, we eventually obtain the ODE

$$\frac{d}{d\tau}\bar{\boldsymbol{V}}(\tau) = F\bar{\boldsymbol{V}}(\tau)+\boldsymbol{C}(\tau) \tag{21}$$

and initial condition

$$\bar{\boldsymbol{V}}(0) = \left(f_{\rm pay}(\boldsymbol{x}^{(1)}),\ldots,f_{\rm pay}\left(\boldsymbol{x}^{(N_{\rm gr})}\right)\right)^\top. \tag{22}$$

Here, F is an $N_{\rm gr}\times N_{\rm gr}$ real matrix

$$F := F^{\rm 1st}+F^{\rm 2nd}-r\boldsymbol{I}^{\otimes d} \tag{23}$$

$$F^{\rm 2nd} := \sum_{i=1}^{d}\frac{\sigma_i^2}{2h_i^2}I^{\otimes i-1}\otimes D_{x_i}^{\rm 2nd}\otimes I^{\otimes d-i}+\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\frac{\sigma_i\sigma_j\rho_{ij}}{4h_ih_j} \times I^{\otimes i-1}\otimes D_{x_i}^{\rm 1st}\otimes I^{\otimes j-i-1}\otimes D_{x_j}^{\rm 1st}\otimes I^{\otimes d-j} \tag{24}$$

$$F^{\rm 1st} := r\sum_{i=1}^{d}\frac{1}{2h_i}I^{\otimes i-1}\otimes D_{x_i}^{\rm 1st}\otimes I^{\otimes d-i} \tag{25}$$

where I is an $n_{\rm gr}\times n_{\rm gr}$ identity matrix, and $D_{x_i}^{\rm 1st}$ and $D_{x_i}^{\rm 2nd}$ are the $n_{\rm gr}\times n_{\rm gr}$ real matrices. $\boldsymbol{C}(\tau)$ is a vector corresponding to the boundary conditions. The elements of the $D_{x_i}^{\rm 1st}$, $D_{x_i}^{\rm 2nd}$, and $\boldsymbol{C}(\tau)$ are shown in Appendix A. $n_{\rm gr}$ has to be proportional to $O(\epsilon^{-1/2})$ to obtain the present price of the derivative within the accuracy ϵ [31]. Then, the dimension of $\bar{\boldsymbol{V}}(\tau)$ is $O((1/\epsilon)^{d/2})$. Thus, it becomes difficult to solve the BSPDE discretized by FDM using the classical algorithm when multiple assets need to be considered.

E. APPROXIMATION OF THE PRESENT DERIVATIVE PRICE BY THE EXPECTED VALUE OF THE DERIVATIVE PRICE AT THE FUTURE TIME

As shown in [31], we can evaluate the present price of the derivative by the expected value of the price at a future time $t_{\rm ter}$. Here, we briefly review the method. To calculate the present value of the derivative, recalling the fact that the derivative price is a martingale [29], [30], we evaluate $V(0,\boldsymbol{s}_0)$ as

$$V(0,\boldsymbol{s}_0) = e^{-rt_{\rm ter}}\int_{\mathbb{R}_+^d} d\boldsymbol{s}p(t_{\rm ter},\boldsymbol{s})p_{\rm NB}(t_{\rm ter},\boldsymbol{s})V(t_{\rm ter},\boldsymbol{s}) \tag{26}$$

where $t_{\rm ter}$ is any value in $[0,T]$, $p(t,\boldsymbol{s})$ is the probability density function of $\boldsymbol{S}(t)$, $p_{\rm NB}(t,\boldsymbol{s})$ is the probability that no event that leads to extinction of the payoff (hereafter, the *out event*) happens by t given $\boldsymbol{S}(t_{\rm ter})=\boldsymbol{s}$, and $V(t_{\rm ter},\boldsymbol{s})$ is the derivative price at $t_{\rm ter}$ when $\boldsymbol{S}(t_{\rm ter})=\boldsymbol{s}$ and no out event happens by $t_{\rm ter}$.

Some cares must be taken to utilize (26). First, although we can obtain the solution of (3) only within the boundaries, (26) contains the information of the events outside the boundaries. Second, it is difficult to calculate $p_{\rm NB}(t_{\rm ter},\boldsymbol{s})$ explicitly in the multiasset case. The first problem can be neglected for small $t_{\rm ter}$ since the distribution of $\boldsymbol{S}(t_{\rm ter})$ outside the boundary is negligible in this case, and so is the contribution from the outside of D in (26). The second problem is also solved by using sufficiently small $t_{\rm ter}$; in this case, the probability that the underlying asset prices reach any boundaries is negligible, and thus, $p_{\rm NB}(t_{\rm ter},\boldsymbol{s})$ is nearly equal to 1 since we are now assuming that the payoff will be paid as far as the underlying asset prices stay in the boundaries. Therefore, for such $t_{\rm ter}$, we can evaluate $V(0,\boldsymbol{s}_0)$ as

$$V(0,\boldsymbol{s}_0) \simeq e^{-rt_{\rm ter}}\int_D d\boldsymbol{s}p(t_{\rm ter},\boldsymbol{s})V(t_{\rm ter},\boldsymbol{s}). \tag{27}$$

When we use a quantum algorithm to calculate (26) by $\int_D d\boldsymbol{s}p(t_{\rm ter},\boldsymbol{s})V(t_{\rm ter},\boldsymbol{s}) \simeq \langle\boldsymbol{p}(t_{\rm ter})|\boldsymbol{V}(t_{\rm ter})\rangle$, the overlap between $|\boldsymbol{p}(t_{\rm ter})\rangle$ and $|\boldsymbol{V}(t_{\rm ter})\rangle$ should be as large as possible since the number of measurements for the evaluation of the inner product decreases as the overlap becomes large (see Section III-A for details). As the probability density function $p(t,\boldsymbol{s})$ broadens over time, taking a large $t_{\rm ter}$ results in a large overlap. Thus, we want to take $t_{\rm ter}$ as large as possible from this viewpoint.

Taking into account this tradeoff, we set $t_{\rm ter}$ as large as possible to the extent that (26) is well approximated with (27). As a conclusion, for sufficiently small ϵ, we may set

$$t_{\rm ter} = \min\left\{\frac{2\left(\log\left(\frac{u_1}{s_{1,0}}\right)\right)^2}{25\sigma_1^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)},\ldots,\frac{2\left(\log\left(\frac{u_d}{s_{d,0}}\right)\right)^2}{25\sigma_d^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}\right.$$

$$\left.\frac{2\left(\log\left(\frac{s_{1,0}}{l_1}\right)\right)^2}{25\sigma_1^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)},\ldots,\frac{2\left(\log\left(\frac{s_{d,0}}{l_d}\right)\right)^2}{25\sigma_d^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}\right\} \tag{28}$$

for the approximation (27) with $O(\epsilon)$ accuracy. Here, we assume that there exist positive constants $A_0, A_1, \ldots, A_d$ such that f_{pay} satisfies $f_{\text{pay}}(\boldsymbol{s}) \le \sum_{i=1}^{d} A_i s_i + A_0$ for any $\boldsymbol{s} \in D$ and define $\tilde{A} = \max\{A_1\sqrt{u_1 s_{1,0}}, \ldots, A_d\sqrt{u_d s_{d,0}}, A_0\}$. For the full detail, see [31, Sec. IV].

F. VARIATIONAL QUANTUM SIMULATION

In this section, we introduce the VQS, which is a variational quantum algorithm to solve linear ODEs [44], [45], [46]. Consider solving the following linear ODE:

$$\frac{d}{dt}\boldsymbol{v}(t) = L(t)\boldsymbol{v}(t) + \boldsymbol{u}(t),\ \boldsymbol{v}(0) = \boldsymbol{v}_0 \tag{29}$$

where $\boldsymbol{v}(t) = (v_1(t), \ldots, v_{N_v}(t))$, $\boldsymbol{v}_0 = (v_{0,1}, \ldots, v_{0,N_v})$, $\boldsymbol{u}(t) = (u_1(t), \ldots, u_{N_v}(t)) \in \mathbb{C}^{N_v}$, and $L(t)$ is an (possibly non-Hermitian) operator. To simulate the vector $\boldsymbol{v}(t)$, we instead simulate an unnormalized quantum state $|\boldsymbol{v}(t)\rangle$, which is the solution of

$$\frac{d}{dt}|\boldsymbol{v}(t)\rangle = L(t)|\boldsymbol{v}(t)\rangle + |\boldsymbol{u}(t)\rangle,\ |\boldsymbol{v}(0)\rangle = |\boldsymbol{v}_0\rangle. \tag{30}$$

Here, we make three assumptions. First, $L(t)$ can be decomposed as

$$L(t) = \sum_{k=1}^{N_L} \lambda_k(t) U_k^L(t) \tag{31}$$

where $\lambda_k(t)$ is real, and $U_k^L(t)$ are quantum gates. Second, $|\boldsymbol{u}(t)\rangle$ can be written as

$$|\boldsymbol{u}(t)\rangle = \sum_{l=1}^{N_u} \eta_l(t) U_l^u(t)|0\rangle \tag{32}$$

where $\eta_l(t)$ is real, and $U_l^u(t)$ are the quantum gates. Third, there are some constant $\alpha_v \in \mathbb{C}$ and a quantum gate U_v such that $|\boldsymbol{v}_0\rangle = \alpha_v U_v|0\rangle$. In VQS, we approximate $|\boldsymbol{v}(t)\rangle$ by an unnormalized ansatz state $|\tilde{v}(\boldsymbol{\theta}(t))\rangle := \theta_0(t)R_1(\theta_1(t))R_2(\theta_2(t))\cdots R_{N_a}(\theta_{N_a}(t))|\boldsymbol{v}_0\rangle$ and determine parameters $\boldsymbol{\theta}(t) = (\theta_0(t), \theta_1(t), \ldots, \theta_{N_a}(t))^\top \in \mathbb{R}^{N_a+1}$ by the variational principle. Here, $R_k(\theta_k) = W_k e^{i\theta_k G_k}$ are the parameterized quantum circuits, W_k are the quantum gates, and $G_k \in \{X, Y, Z, I\}^{\otimes n}$ are the multiqubit Pauli gates with n-qubit system. By McLachlan's variational principle [50], the parameters $\boldsymbol{\theta}$ that satisfy (30) are obtained by solving

$$\min_{\dot{\boldsymbol{\theta}}} \left\| \frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle - L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle - |\boldsymbol{u}(t)\rangle \right\|^2 \tag{33}$$

and then, we obtain the differential equation [44]

$$\sum_{n=0}^{N_a} \mathcal{M}_{m,n}\dot{\theta}_n(t) = \mathcal{V}_m \tag{34}$$

Algorithm 1: Derivative Pricing With Variational Quantum Algorithms.

1: Prepare $\alpha_p U_p$ such that $\alpha_p U_p|0\rangle = |\psi_p\rangle \simeq \sum_{k=1}^{N_{\text{gr}}} p_k(t_{\text{ter}})|k\rangle$ by VQS for Fokker–Planck equation or quantum generative models.
2: Prepare $\alpha_V U_V$ such that $\alpha_V U_V|0\rangle = |\psi_V\rangle \simeq \sum_{k=1}^{N_{\text{gr}}} f_{\text{pay}}(\boldsymbol{x}^{(k)})|k\rangle$ by quantum generative models.
3: Calculate $|\tilde{v}(\boldsymbol{\theta}(\tau_{\text{ter}}))\rangle$ by performing VQS from $\tau = 0$ to $\tau = \tau_{\text{ter}}$.
4: Perform the SWAP test and get an estimation of $|\langle\psi^p|\tilde{v}(\boldsymbol{\theta}(\tau_{\text{ter}}))\rangle|^2$
5: $V_0 \leftarrow e^{-rt_{\text{ter}}}\langle\psi_p|\tilde{v}(\boldsymbol{\theta}(\tau_{\text{ter}}))\rangle$.

where

$$\mathcal{M}_{i,j} = \text{Re}\left(\frac{\partial\langle\tilde{v}(\boldsymbol{\theta}(t))|}{\partial\theta_i}\frac{\partial|\tilde{v}(\boldsymbol{\theta}(t))\rangle}{\partial\theta_j}\right) \tag{35}$$

$$\mathcal{V}_j = \sum_{k=1}^{N_L}\lambda_k(t)\text{Re}\left(\frac{\partial\langle\tilde{v}(\boldsymbol{\theta}(t))|}{\partial\theta_j}U_k^L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right) + \sum_{l=1}^{N_u}\eta_l(t)\text{Re}\left(\frac{\partial\langle\tilde{v}(\boldsymbol{\theta}(t))|}{\partial\theta_n}U_l^u(t)|0\rangle\right). \tag{36}$$

We can evaluate each term in (35) and (36) by quantum circuits presented in Appendix D. Then, we solve (34) classically and obtain $\dot{\theta}_j(t)$. Note that the number of measurements needed to evaluate $\mathcal{M}_{i,j}$ and $\mathcal{V}_i$ by the Hadamard test within the accuracy $\bar{\epsilon}$ is $O(|\theta_0(t)|^2/\bar{\epsilon}^2)$. This is because $\mathcal{M}_{i,j}$ and $\mathcal{V}_i$ contain the normalization factor $\theta_0(t)$ when $i > 0$ or $j > 0$ (see Appendix D). We assume that $|\theta_0(t)|$ is upper bounded by some constant. In derivative pricing, $|\theta_0(t)|^2$ is about a ratio of the sum of the squares of the derivative prices at time $T - t$ to the sum of the squares of the payoff function at maturity. Since the derivative price is the expected value of the payoff function, this assumption is satisfied if the value range of the payoff function is finite. Starting from $t = 0$, we obtain the time evolution of $\boldsymbol{\theta}(t)$ by repeating

$$\boldsymbol{\theta}(t + \Delta t) \leftarrow \boldsymbol{\theta}(t) + \dot{\boldsymbol{\theta}}(t)\Delta t \tag{37}$$

where Δt is an interval in time direction. Consequently, we obtain $|\tilde{v}(\boldsymbol{\theta}(t))\rangle$ that approximates $|\boldsymbol{v}(t)\rangle$.

III. PROPOSED METHOD

In this section, we describe the variational quantum algorithm for derivative pricing and the computational complexity of the proposed method. The overall algorithm is shown in Algorithm 1. We assume that $N_{\text{gr}} = 2^{nd}$ with the n-qubit system.

First, we set $\tau_{\text{ter}} = T - t_{\text{ter}}$, where t_{ter} is defined in (28). To perform VQS, we need to represent the operator corresponding to F in (23) and the operator $\tilde{G}$ such that $\tilde{G}|0\rangle = |\boldsymbol{C}\rangle = \sum_{k=1}^{N_{\text{gr}}} C_k(\tau)|k\rangle$ by a linear combination of quantum gates, respectively, because of the assumptions (31) and (32). Such

IEEE Transactions on Quantum Engineering

decomposition can be obtained in a similar way to the articles presented in [22] and [47] and is shown in Appendix B. F can be represented as a sum of $O(d^2n^4)$ unitaries each of which requires at most $O(n^2)$ gates to be implemented. $\tilde{G}$ for typical boundary conditions discussed in Section II-D can be represented by $O(d^3n^2)$ unitaries, each of which requires at most $O(n^2)$ gates to be implemented.

Second, we prepare the unnormalized state

$$\begin{aligned}|\psi_p\rangle &:= \alpha_p U_p|0\rangle \\ &\simeq |\boldsymbol{p}(t_{\text{ter}})\rangle \\ &= \sum_{k=1}^{N_{\text{gr}}} p_k(t_{\text{ter}})|k\rangle \end{aligned} \tag{38}$$

where $\alpha_p \in \mathbb{C}$, and U_p is a quantum gate. $p_k(t_{\text{ter}})$ is a probability that the underlying asset prices are on $\boldsymbol{x}^{(k)}$ at t_{ter}. We can obtain such α_p and U_p by solving the Fokker–Planck equation, which describes the time evolution of the probability density function, using VQS [22], [47]. Alternatively, they can also be obtained by quantum generative models [39], [40], [41], [42], [43] since the probability density function of the underlying asset price at any $t \in [0, T]$ can be obtained analytically under the BS model [see (54) in Section III-A].

Third, we prepare $\alpha_V \in \mathbb{C}$ and U_V such that $\alpha_V U_V|0\rangle =: |\psi_V\rangle$ approximates the initial state of the discretized BSPDE, that is

$$\begin{aligned}|\psi_V\rangle &\simeq |\bar{\boldsymbol{V}}(0)\rangle \\ &= \sum_{k=1}^{N_{\text{gr}}} f_{\text{pay}}\left(\boldsymbol{x}^{(k)}\right)|k\rangle. \end{aligned} \tag{39}$$

To find such α_V and U_V, we can use the quantum generative models [39], [40], [41], [42], [43].

Fourth, we solve the BSPDE from $\tau = 0$ to τ_{ter} using VQS and obtain an unnormalized state

$$|\tilde{v}(\boldsymbol{\theta}(\tau_{\text{ter}}))\rangle \simeq |\bar{\boldsymbol{V}}(\tau_{\text{ter}})\rangle = \sum_{k=1}^{N_{\text{gr}}} \bar{V}_k\left(\tau_{\text{ter}}, \boldsymbol{x}^{(k)}\right)|k\rangle \tag{40}$$

where

$$\begin{aligned}|\tilde{v}\left(\boldsymbol{\theta}(\tau_{\text{ter}})\right)\rangle &:= \theta_0(\tau_{\text{ter}})R_1(\theta_1(\tau_{\text{ter}}))R_2\left(\theta_2(\tau_{\text{ter}})\right) \\ &\quad \cdots R_{N_a}\left(\theta_{N_a}(\tau_{\text{ter}})\right)|\psi_V\rangle \end{aligned} \tag{41}$$

$\{R_k\}_{k\in[N_a]}$ are the parameterized quantum circuits, and $\boldsymbol{\theta}(\tau_{\text{ter}}) := (\theta_0(\tau_{\text{ter}}), \ldots, \theta_{N_a}(\tau_{\text{ter}}))^\top \in \mathbb{R}^{N_a+1}$ are the variational parameters. Note that $\theta_0(0)R_1(\theta_1(0))\cdots R_{N_a}(\theta_{N_a}(0))$ should be an identity operator to satisfy $|\tilde{v}(\boldsymbol{\theta}(0))\rangle \simeq |\bar{\boldsymbol{V}}(0)\rangle$. For example, the ansatz, as shown in Fig. 1, in Section IV with even number of layers can be used as $\{R_k\}_{k\in[N_a]}$ that satisfies this condition with the parameters $\boldsymbol{\theta}(0) = (0, \ldots, 0)^\top$ since RY gates are identity for the parameters, and CZ layers cancel each other and also become identity.

Finally, we use the SWAP test [48] for two normalized states $U_p|0\rangle, R_1(\theta_1(\tau_{\text{ter}}))\cdots R_{N_a}(\theta_{N_a}(\tau_{\text{ter}}))U_V|0\rangle$ and obtain

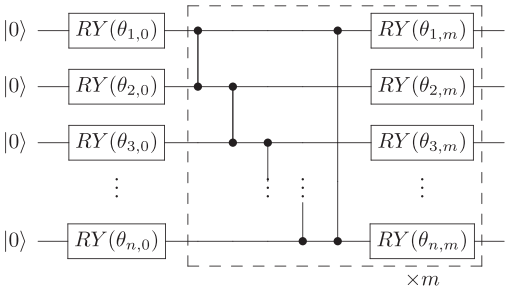

FIGURE 1. In a depth-*m* circuit, *CZ* and *RY* gates (enclosed by dashed lines) are repeated *m*-times. The circuit has $n(m+1)$ parameters.

$$\begin{aligned}\left|\langle\psi_p|\tilde{v}(\boldsymbol{\theta}(\tau_{\text{ter}}))\rangle\right|^2 &= \left|\alpha_p\alpha_V\theta_0(\tau_{\text{ter}})\right|^2 \\ &\times \left|\langle 0|U_p^\dagger R_1(\theta_1(\tau_{\text{ter}}))\cdots R_{N_a}(\theta_{N_a}(\tau_{\text{ter}}))U_V|0\rangle\right|^2. \end{aligned} \tag{42}$$

As discussed in Section II-E, the present price of the derivative is approximated by the inner product $\langle \boldsymbol{p}(t_{\text{ter}})|\bar{\boldsymbol{V}}(\tau_{\text{ter}})\rangle$ discounted by the risk-free rate. We can approximate the inner product by the square root of the result of the SWAP test and obtain the present price of the derivative by

$$V_0 \simeq e^{-rt_{\text{ter}}}\langle\psi_p|\tilde{v}\left(\boldsymbol{\theta}(\tau_{\text{ter}})\right)\rangle. \tag{43}$$

For the third and fourth parts, we may take a slightly different approach. That is, we find $\boldsymbol{\theta}(0)$ such that

$$|\bar{\boldsymbol{V}}(0)\rangle \simeq \theta_0(0)R_1(\theta_1(0))R_2(\theta_2(0))\cdots R_{N_a}(\theta_{N_a}(0))|0\rangle \tag{44}$$

and obtain

$$\begin{aligned}|\tilde{v}(\boldsymbol{\theta}(\tau_{\text{ter}}))\rangle &= \theta_0(\tau_{\text{ter}})R_1(\theta_1(\tau_{\text{ter}}))R_2\left(\theta_2(\tau_{\text{ter}})\right) \\ &\quad \cdots R_{N_a}\left(\theta_{N_a}(\tau_{\text{ter}})\right)|0\rangle \\ &\simeq |\bar{\boldsymbol{V}}\left(\tau_{\text{ter}}\right)\rangle \end{aligned} \tag{45}$$

using VQS. This approach may reduce the number of gates by eliminating U_V but since the ansatz for the initial state also serves as the ansatz for VQS, the number of gates required for the ansatz may become larger. For this reason, it is difficult to say which approach is better in general, but we adopt the one in Algorithm 1 for the numerical simulation in Section IV.

A. NUMBER OF MEASUREMENTS IN THE SWAP TEST

In this section, we estimate the number of measurements required for the SWAP test to estimate V_0 with accuracy ϵ. For simplicity, we consider the case where $|\tilde{v}(\boldsymbol{\theta}(\tau_{\text{ter}}))\rangle = |\bar{\boldsymbol{V}}(\tau_{\text{ter}})\rangle$ and $|\psi_p\rangle = |\boldsymbol{p}(t_{\text{ter}})\rangle$. We perform the SWAP test for two normalized states $|\tilde{\boldsymbol{p}}\rangle$ and $|\tilde{\bar{\boldsymbol{V}}}\rangle$ such that $|\boldsymbol{p}(t_{\text{ter}})\rangle = \alpha|\tilde{\boldsymbol{p}}\rangle$, $|\bar{\boldsymbol{V}}(\tau_{\text{ter}})\rangle = \beta|\tilde{\bar{\boldsymbol{V}}}\rangle$, where

$$\alpha = \sqrt{\sum_{k=1}^{N_{\text{gr}}} p_k(t_{\text{ter}})^2} \tag{46}$$

$$\beta = \sqrt{\sum_{k=1}^{N_{\mathrm{gr}}} \bar{V}\left(\tau_{\mathrm{ter}}, \boldsymbol{x}^{(k)}\right)^2}. \tag{47}$$

To obtain the value of the inner product $|\langle\tilde{\boldsymbol{p}}|\tilde{\bar{\boldsymbol{V}}}\rangle|^2$ with precision $\bar{\varepsilon}$, the SWAP test requires $O(\frac{1}{\bar{\varepsilon}^2})$ measurements [48]. When we have the estimation $\widetilde{|\langle\tilde{\boldsymbol{p}}|\tilde{\bar{\boldsymbol{V}}}\rangle|^2}$ such that

$$\left|\left|\left\langle\tilde{\boldsymbol{p}}|\tilde{\bar{\boldsymbol{V}}}\right\rangle\right|^2 - \widetilde{\left|\langle\tilde{\boldsymbol{p}}|\tilde{\bar{\boldsymbol{V}}}\rangle\right|^2}\right| < \bar{\varepsilon} \tag{48}$$

the estimation of the inner product of unnormalized states $\widetilde{|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle|^2}$ satisfies

$$\left|\left|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle\right|^2 - \widetilde{\left|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle\right|^2}\right| < \alpha^2\beta^2\bar{\varepsilon}. \tag{49}$$

Here, we denote $\widetilde{|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle|^2}$ by c for notational simplicity. Then, $\sqrt{c}$ can be the estimator of $|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle|$. Since $|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle| = e^{rt_{\mathrm{ter}}}V_0$ holds, we obtain

$$\begin{aligned}\left|\left|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle\right| - \sqrt{c}\right| &< \frac{\alpha^2\beta^2\bar{\varepsilon}}{\left|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle\right| + \sqrt{c}} \\ &\le \frac{\alpha^2\beta^2\bar{\varepsilon}}{e^{rt_{\mathrm{ter}}}V_0}.\end{aligned} \tag{50}$$

Thus, $O(\frac{\alpha^4\beta^4}{e^{2rt_{\mathrm{ter}}}V_0^2\bar{\varepsilon}^2})$ measurements are required to obtain $|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle|$ with precision $\alpha^2\beta^2\bar{\varepsilon}/e^{rt_{\mathrm{ter}}}V_0$ and, thus, $e^{-rt_{\mathrm{ter}}}|\langle\boldsymbol{p}(t_{\mathrm{ter}})|\bar{\boldsymbol{V}}(\tau_{\mathrm{ter}})\rangle|$, an estimation of V_0, with precision $\varepsilon := \alpha^2\beta^2\bar{\varepsilon}/e^{2rt_{\mathrm{ter}}}V_0$. Also note that since we can classically calculate α by the analytical form of $p(t,\boldsymbol{s})$, and β is calculated by $\alpha_V\theta_0(\tau_{\mathrm{ter}})$, we can determine the number of measurements before the SWAP test from VQS results.

To estimate the number of measurements of the SWAP test, we estimate $\alpha^2\beta^2$, which is calculated as

$$\begin{aligned}\alpha^2\beta^2 &= \left(\sum_{k=1}^{N_{\mathrm{gr}}} p_k(t_{\mathrm{ter}})^2\right)\left(\sum_{k=1}^{N_{\mathrm{gr}}} \bar{V}\left(\tau_{\mathrm{ter}}, \boldsymbol{x}^{(k)}\right)^2\right) \\ &= \left(\sum_{k=1}^{N_{\mathrm{gr}}} p_k(t_{\mathrm{ter}})^2\right)\left(\sum_{k=1}^{N_{\mathrm{gr}}} f_{\mathrm{pay}}(\boldsymbol{x}^{(k)})^2\right)\frac{\sum_{k=1}^{N_{\mathrm{gr}}} \bar{V}(\tau_{\mathrm{ter}}, \boldsymbol{x}^{(k)})^2}{\sum_{k=1}^{N_{\mathrm{gr}}} f_{\mathrm{pay}}(\boldsymbol{x}^{(k)})^2}.\end{aligned} \tag{51}$$

Although it is difficult to estimate the factor $\frac{\sum_{k=1}^{N_{\mathrm{gr}}} \bar{V}(\tau_{\mathrm{ter}}, \boldsymbol{x}^{(k)})^2}{\sum_{k=1}^{N_{\mathrm{gr}}} f_{\mathrm{pay}}(\boldsymbol{x}^{(k)})^2}$ in advance, we assume that the factor is bounded by some constant ζ. This assumption means that the rate of change in derivative prices over time is suppressed by a certain constant. Under the assumption, we estimate $(\sum_{k=1}^{N_{\mathrm{gr}}} f_{\mathrm{pay}}(\boldsymbol{x}^{(k)})^2)(\sum_{k=1}^{N_{\mathrm{gr}}} p_k(t_{\mathrm{ter}})^2)$. We assume that $f_{\mathrm{pay}}(\boldsymbol{x})$ for $\boldsymbol{x} \in D$ is upper bounded by some constant B. For example, in the case of the basket call option with $a_0 < 0, a_1, \ldots, a_d > 0$

$$\begin{aligned}f_{\mathrm{pay}}(\boldsymbol{x}) &= \max\left(a_0 + \sum_{i=1}^{d} a_i x_i, 0\right) \\ &\le a_0 + \sum_{i=1}^{d} a_i x_i \\ &\le a_0 + \sum_{i=1}^{d} a_i u_i\end{aligned} \tag{52}$$

holds. From this assumption, we obtain

$$\begin{aligned}\sum_{k=1}^{N_{\mathrm{gr}}} f_{\mathrm{pay}}(\boldsymbol{x}^{(k)})^2 &\le \sum_{k=1}^{N_{\mathrm{gr}}} B^2 \\ &= N_{\mathrm{gr}}B^2.\end{aligned} \tag{53}$$

On the other hand, the probability density function of the d-dimensional geometric Brownian motion with $\boldsymbol{x}(0) = \boldsymbol{x}_0 := (x_{0,1}, \ldots, x_{0,d})^\top$ is

$$\begin{aligned}p(t,\boldsymbol{x}) = &\frac{1}{(2\pi t)^{d/2}\left(\prod_{i=1}^{d}\sigma_i x_i\right)\sqrt{\det\rho}} \\ &\times \exp\left(-\frac{1}{2}(\ln\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\ln\boldsymbol{x} - \boldsymbol{\mu})\right)\end{aligned} \tag{54}$$

where

$$\boldsymbol{\mu} = \left(\left(r - \frac{\sigma_1^2}{2}\right)t + \ln x_{0,1}, \ldots, \left(r - \frac{\sigma_d^2}{2}\right)t + \ln x_{0,d}\right)^\top. \tag{55}$$

The square of probability density function is

$$\begin{aligned}p(t,\boldsymbol{x})^2 = &\left(\frac{1}{(2\pi t)^{d/2}\left(\prod_{i=1}^{d}\sigma_i x_i\right)\sqrt{\det\rho}}\right)^2 \\ &\times \exp\left(-(\ln\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\ln\boldsymbol{x} - \boldsymbol{\mu})\right) \\ = &\frac{\gamma(t)}{\prod_{i=1}^{d} x_i}\frac{1}{(2\pi t)^{d/2}\left(\prod_{i=1}^{d}\frac{\sigma_i x_i}{2}\right)\sqrt{\det\rho}} \\ &\times \exp\left(-\frac{1}{2}(\ln\boldsymbol{x} - \boldsymbol{\mu})^\top\left(\frac{1}{2}\Sigma\right)^{-1}(\ln\boldsymbol{x} - \boldsymbol{\mu})\right) \\ = &\frac{\gamma(t)}{\prod_{i=1}^{d} x_i}\varphi(t,\boldsymbol{x})\end{aligned} \tag{56}$$

where

$$\gamma(t) = \frac{1}{(8\pi t)^{d/2}\prod_{i=1}^{d}\sigma_i} \tag{57}$$

and $\varphi(t,\boldsymbol{x})$ is a probability density function of some log-normal distribution. Using the probability distribution function, the square sum of the discretized density function is

represented by

$$\begin{aligned}\sum_{k=1}^{N_{\rm gr}} p_k(t_{\rm ter})^2 &= \sum_{k=1}^{N_{\rm gr}} \left(p\left(t_{\rm ter}, \boldsymbol{x}^{(k)}\right)\right)^2 \left(\prod_{i=1}^{d} h_i\right)^2 \\ &= \sum_{k=1}^{N_{\rm gr}} \frac{\gamma(t_{\rm ter})}{\prod_{i=1}^{d} x_i^{(k_i)}} \varphi\left(t_{\rm ter}, \boldsymbol{x}^{(k)}\right) \left(\prod_{i=1}^{d} h_i\right)^2 \\ &\le \frac{\gamma(t_{\rm ter})}{\prod_{i=1}^{d} l_i} \sum_{k=1}^{N_{\rm gr}} \varphi(t_{\rm ter}, \boldsymbol{x}^{(k)}) \left(\prod_{i=1}^{d} h_i\right)^2 \\ &\simeq \frac{\gamma(t_{\rm ter})}{\prod_{i=1}^{d} l_i} \prod_{i=1}^{d} h_i \int_{\mathbb{R}_+^d} \varphi(t_{\rm ter}, \boldsymbol{x})\, d\boldsymbol{x} \\ &= \frac{\gamma(t_{\rm ter})}{\prod_{i=1}^{d} l_i} \prod_{i=1}^{d} h_i \\ &\simeq \frac{\gamma(t_{\rm ter})}{\prod_{i=1}^{d} l_i} \frac{1}{N_{\rm gr}} \prod_{i=1}^{d} (u_i - l_i). \end{aligned} \tag{58}$$

From (53) and (58), we obtain

$$\begin{aligned}\alpha^2\beta^2 &\lesssim \zeta B^2 \frac{1}{(8\pi t_{\rm ter})^{d/2}} \prod_{i=1}^{d} \frac{1}{\sigma_i} \left(\frac{u_i}{l_i} - 1\right) \\ &=: \Xi. \end{aligned} \tag{59}$$

Since $t_{\rm ter}$ is lower bounded by

$$\begin{aligned} & t_{\rm ter} \\ &= \min\left\{ \frac{2\left(\log\left(\frac{u_1}{s_{1,0}}\right)\right)^2}{25\sigma_1^2 \log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}, \ldots, \frac{2\left(\log\left(\frac{u_d}{s_{d,0}}\right)\right)^2}{25\sigma_d^2 \log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)} \right. \\ &\quad \left. \frac{2\left(\log\left(\frac{s_{1,0}}{l_1}\right)\right)^2}{25\sigma_1^2 \log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}, \ldots, \frac{2\left(\log\left(\frac{s_{d,0}}{l_d}\right)\right)^2}{25\sigma_d^2 \log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)} \right\} \\ &\ge \frac{2(\log\chi_{\min})^2}{25\sigma_{\max}^2} \left(\log\frac{2\tilde{A}d(d+1)}{\epsilon}\right)^{-1} \end{aligned} \tag{60}$$

where $\sigma_{\max} := \max_{i\in[d]}\{\sigma_i\}$ and $\chi_{\min} := \min_{i\in[d]}\{u_i/s_{i,0}\} \cup \{s_{i,0}/l_i\}$, we obtain

$$\Xi \le \zeta B^2 \left(\frac{5}{4\pi^2} \frac{\xi_{\max} - 1}{\log\chi_{\min}} \frac{\sigma_{\max}}{\sigma_{\min}}\right)^d \left(\log\frac{2\tilde{A}d(d+1)}{\epsilon}\right)^{d/2} \tag{61}$$

where $\sigma_{\min} := \min_{i\in[d]}\{\sigma_i\}$ and $\xi_{\max} := \max_{i\in[d]}\{u_i/l_i\}$[3]. We find that the number of measurements required by the SWAP test is

$$N_{\rm SWAP} = \frac{\zeta^2 B^4}{2e^{4rt_{\rm ter}} V_0^2 \varepsilon^2} \left(\frac{5}{4\pi^2} \frac{\xi_{\max} - 1}{\log\chi_{\min}} \frac{\sigma_{\min}}{\sigma_{\max}}\right)^{2d} \left(\log\frac{2\tilde{A}d(d+1)}{\epsilon}\right)^d. \tag{62}$$

Note that $N_{\rm SWAP}$ does not have the dependency of the form as $(1/\epsilon)^{O(d)}$, which means that the proposed method achieves a significant speedup over classical FDM with respect to ϵ and d, when the other parts of the proposed method are sufficiently efficient.

Here, we consider the limit of $t_{\rm ter} \to 0$. This is the case with taking the inner product of $|\boldsymbol{V}(0)\rangle$ and $|\boldsymbol{p}(0)\rangle$. In this case, the probability density function (54) is a delta function, which means that the present price of the underlying assets is $\boldsymbol{x}_0$ with probability 1. Assuming that $\boldsymbol{x}_0$ is on a grid point with the index k_0, $p_k(0)$ is 1 for $k = k_0$ and 0 otherwise, and the sum of the squares of $p_k(0)$ is 1. As a result, $\alpha^2\beta^2$ is upper bounded as follows:

$$\alpha^2\beta^2 \le \zeta N_{\rm gr} B^2. \tag{63}$$

Thus, the number of the measurement is proportional to $N_{\rm gr} = n_{\rm gr}^d$, and the quantum speedup will be lost, as in the case of retrieving one amplitude of the computational basis from $|\boldsymbol{V}(0)\rangle$ as in [13] and [15].

B. COMPUTATIONAL COMPLEXITY OF THE PROPOSED METHOD

Here, we discuss the computational complexity of our algorithm. We assume that the number of quantum gates required for preparing $|\boldsymbol{p}(t_{\rm ter})\rangle$ and $|\bar{\boldsymbol{V}}(0)\rangle$ is $N_{\rm gate}^p$ and $N_{\rm gate}^V$, respectively. We also assume that the number of measurements required to prepare $|\boldsymbol{p}(t_{\rm ter})\rangle$ and $|\bar{\boldsymbol{V}}(0)\rangle$ is $N_{\rm measure}^p$ and $N_{\rm measure}^V$, respectively. $N_{\rm gate}^p$, $N_{\rm gate}^V$, $N_{\rm measure}^p$, and $N_{\rm measure}^V$ depend on the implementation of the generative models, but we assume that all of them are $O(\mathrm{poly}(d\log(1/\epsilon)))$. This means that we assume that the generative models efficiently generate the (unnormalized) quantum states. Note that VQS requires controlled versions of U_k^L, U_l^u in (36), or those of $\mathcal{R}U_v$, where $\mathcal{R}$ is defined in (105) (see Appendix D). Since U_k^L and U_l^u are the terms of the linear combination of F and $\tilde{G}$, respectively, they are made by $O(n^2) = O(d^2\log(1/\epsilon)^2)$ gates. Thus, $O(\mathrm{poly}(d\log(1/\epsilon)))$ gates are required for the control unitaries of U_k^L and U_l^u. Assuming $\mathcal{R}U_v$ is made by $O(\mathrm{poly}(d\log(1/\epsilon)))$ gates, the controlled-$\mathcal{R}U_v$ gate requires $O(\mathrm{poly}(d\log(1/\epsilon)))$ gates. Consequently, quantum circuits containing $O(\mathrm{poly}(d\log(1/\epsilon)))$ quantum gates are required for VQS. We assume that the number of measurements to estimate $\mathcal{M}_{i,j}$ and $\mathcal{V}_i$ is $N_{\rm measure}^{\rm VQS}$. As defined in Section II-F, the number of parameters is N_a, and then the number of elements $\mathcal{M}_{i,j}$ and $\mathcal{V}_i$ is $(N_a+1)^2$ and N_a+1, respectively. Assuming that the number of steps for VQS is N_τ, we obtain that the total number of measurements for VQS is $N_{\rm measure}^{\rm VQS}(N_a+1)^2 N_\tau$. The number of quantum gates to perform the SWAP test is $O(\mathrm{poly}(d\log(1/\epsilon)))$ since the SWAP test requires $O(n) = O(d\log(1/\epsilon))$ quantum gates in addition to the quantum gates to generate $|\boldsymbol{p}(t_{\rm ter})\rangle$ and $|\bar{\boldsymbol{V}}(\tau)\rangle$ [48]. The number of measurements for the SWAP test

[3]When $\xi_{\max}$ is close to 1, one may find it strange that as Ξ decreases exponentially with respect to the number of assets d, and then, the number of measurements also decreases exponentially. We show that such an exponential decrease does not occur by evaluating the lower bound of Ξ. See Appendix E for details.

TABLE I Complexities of the Proposed Method

Part of the algorithm	# of quantum gates	# of measurements
Preparing $\|\boldsymbol{p}_{\rm ter}\rangle$	$N^{p}_{\rm gate}$	$N^{p}_{\rm measure}$
Preparing $\|\bar{V}_{\rm ter}\rangle$	$N^{V}_{\rm gate}$	$N^{V}_{\rm measure}$
VQS	$O(\mathrm{poly}(d\log(1/\epsilon)))$	$N^{\rm VQS}_{\rm measure}(N_a+1)^2N_\tau$
SWAP test	$O(\mathrm{poly}(d\log(1/\epsilon)))$	$N_{\rm SWAP}$ in Eq.(62)

is $N_{\rm SWAP}$ in (62). The summary of the complexities of the proposed method is shown in Table 1.

Note that, although there remains the exponential dependency with respect to d in $N_{\rm SWAP}$, the time complexity does not have any factor as $(1/\epsilon)^{O(d)}$, as discussed in Section III-A. This is the possible advantage of our method since the complexity of the classical FDM and conventional quantum algorithm has a factor like $(1/\epsilon)^{O(d)}$. Furthermore, since $|\boldsymbol{p}(t_{\rm ter})\rangle$ and $|\bar{\boldsymbol{V}}(\tau)\rangle$ are the states on registers with $O(\log N_{\rm gr}) = O(d\log(1/\epsilon))$ qubits, our method exponentially reduces space complexity compared with that of classical FDM, $(1/\epsilon)^{O(d)}$.

IV. NUMERICAL RESULTS

In this section, we validate the proposed method using numerical calculations. This experiment focuses on a single-asset double knock-out barrier option, which contains both *up and out* and *down and out* conditions[4]. According to Kunitomo and Ikeda [51], the analytical solution for the single-asset double-barrier option $\tilde{V}$ with an upper bound u and a lower bound l is

$$\begin{aligned}
&\tilde{V}(t)\\
&= S_0\sum_{n=-\infty}^{\infty}\left\{\left(\frac{u^n}{l^n}\right)^c\left[\mathcal{N}(d_{1n})-\mathcal{N}(d_{2n})\right]\right.\\
&\quad\left.-\left(\frac{u^{n+1}}{l^nS_0}\right)^c\left[\mathcal{N}(d_{3n})-\mathcal{N}(d_{4n})\right]\right\}-Ke^{-r\tau}\\
&\quad\times\sum_{n=-\infty}^{\infty}\left\{\left(\frac{u^n}{l^n}\right)^{c-2}\left[\mathcal{N}\left(d_{1n}-\sigma\sqrt{\tau}\right)-\mathcal{N}\left(d_{2n}-\sigma\sqrt{\tau}\right)\right]\right.\\
&\quad\left.-\left(\frac{u^{n+1}}{l^nS_0}\right)^{c-2}\left[\mathcal{N}(d_{3n}-\sigma\sqrt{\tau})-\mathcal{N}\left(d_{4n}-\sigma\sqrt{\tau}\right)\right]\right\}
\end{aligned}\tag{64}$$

where

$$d_{1n}=\frac{\ln\left(\frac{S_0}{K}\left(\frac{u}{l}\right)^{2n}\right)-\left(r+\frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}\tag{65}$$

$$d_{2n}=\frac{\ln\left(S_0\frac{u^{2n-1}}{l^{2n}}\right)-\left(r+\frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}\tag{66}$$

$$d_{3n}=\frac{\ln\left(\frac{u^{2n+2}}{KS_0l^{2n}}\right)-\left(r+\frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}\tag{67}$$

$$d_{4n}=\frac{\ln\left(S_0\frac{u^{2n+1}}{l^{2n}}\right)-\left(r+\frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}\tag{68}$$

$$c=\frac{2r}{\sigma}+1\tag{69}$$

and $\mathcal{N}(\cdot)$ is the cumulative distribution function of the standard normal distribution. We compare the results obtained by the proposed method with the analytical solution. We use the Euler method for the time evolution of the parameter (34). The step size for the Euler method is $\Delta\tau = 2.5\times10^{-5}$. The parameters are $r=0.001$, $\sigma := \sigma_1 = 0.3$, $T=1$, $S_0=1$, $l := l_1 = 0.5$, $u := u_1 = 2.0$, and $K=1$. The ansatz of VQS for solving the BS model is shown in Fig. 1. This ansatz repeats m parameterized layers consisting of n RY gates and an entanglement layer consisting of CZ gates. The ansatz have $n(m+1)$ parameters. We do not consider noise and statistical errors in the simulation of quantum circuits. In addition, we assume that the initial states $|\bar{\boldsymbol{V}}(0)\rangle$ and $|\boldsymbol{p}(t)\rangle$ for all $t\in[0,T]$ are given. For the simulation of quantum states, we use NumPy [52].

A. PARAMETER DEPENDENCIES OF VQS RESULTS

Before discussing our results, we show the results using the classical FDM in Fig. 2. The plotted curves are $V(0,S_0)\simeq e^{-rt}E[V(t,S)|S(0)=S_0]$ at each $t\in[0,T]$, where $V(t,S)$ is calculated by classical FDM and the expectation is taken with respect to the analytical $p(t,s)$. The error from the analytical solution increases as t increases for $t\geq t_{\rm ter}$. This is because, in the range greater than $t_{\rm ter}$, the probability that the underlying asset price exceeds or falls under the boundary conditions is higher. As the number of the grid points increases, the derivative price by FDM gets closer to the analytical solution at $t_{\rm ter}$. Since $S_0=1$ is not on the grid points, the error increases when the probability distribution approaches the indicator function with $t\to0$.

Fig. 3 shows the present price of the derivative calculated by our proposed method. We perform VQS on the simulator and obtain $|\tilde{\boldsymbol{v}}(\boldsymbol{\theta}(\tau))\rangle$, which is an approximation of $|\bar{\boldsymbol{V}}(\tau)\rangle$. Taking the inner product between $|\tilde{\boldsymbol{v}}(\boldsymbol{\theta}(\tau))\rangle$ and $|\boldsymbol{p}(t)\rangle$, which is calculated by (38) and (54), we obtain the estimation of the present price of the derivative. In the four qubits case, the result of VQS is a good approximation to the classical FDM solutions of 16 grid points. The use of the larger number of qubits, i.e., the larger $n_{\rm gr}$, gives us solutions that are closer to the analytical solution as in the case of the classical FDM. In the case of six qubits with four layers, the number of parameters is 30, which is smaller than the number of grid points of 64, but the solution is somewhat close to the classical FDM. Due to the computational time requirements, we do not run simulations of larger sizes. However, we find that the solution obtained with more layers better approximates the classical FDM solution.

[4]Note that, it is also possible to calculate the price of the knock-in barrier option by the in–out parity once the price of the knock-out barrier option and the price of the vanilla option are calculated [29], [30]. The price of the vanilla option can be obtained by solving BSPDE analytically.

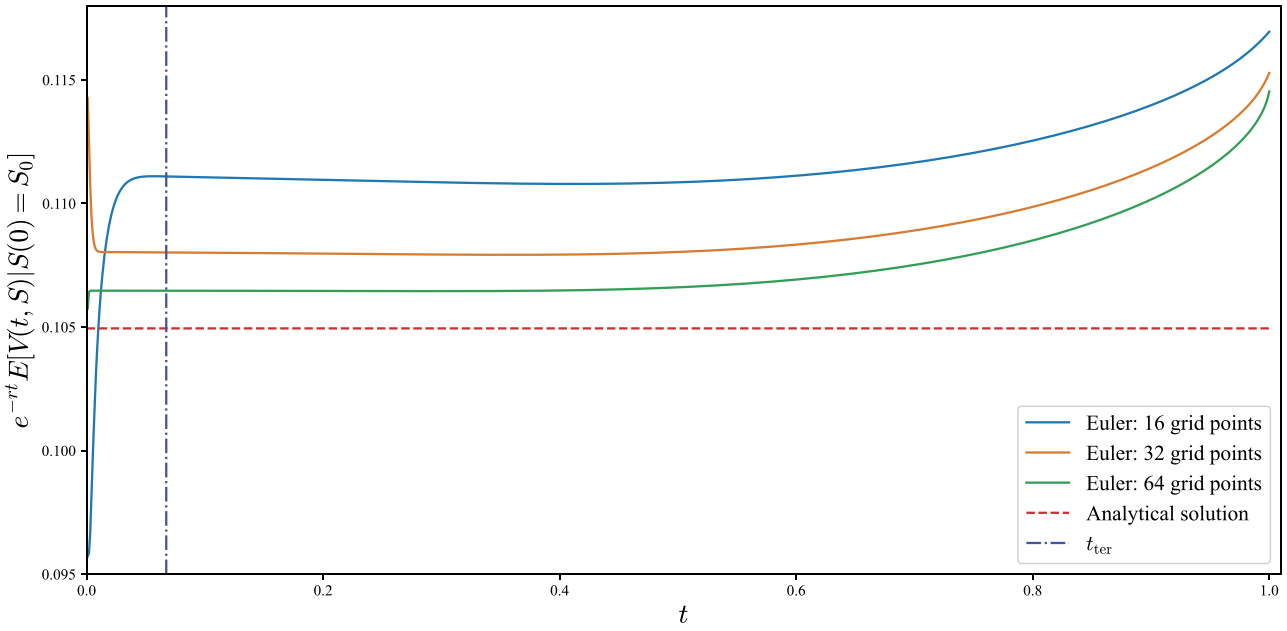

FIGURE 2. Estimated price of the single-asset double-barrier option by classical FDM.

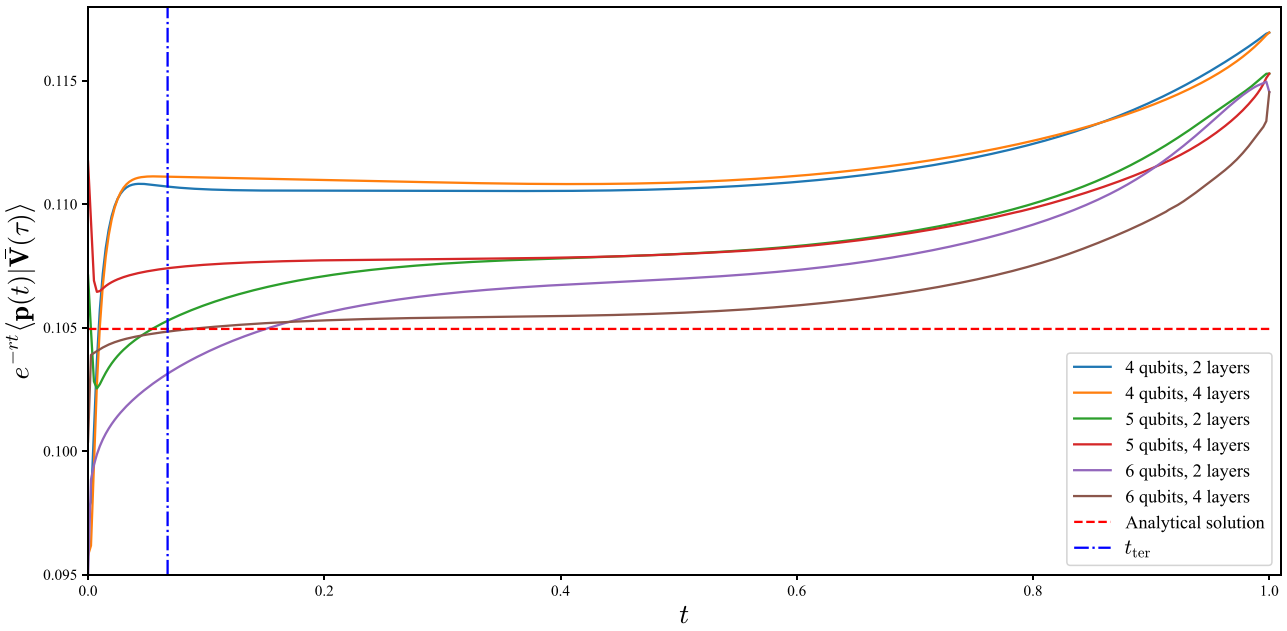

FIGURE 3. Estimated price of the single-asset double-barrier option by the proposed method.

B. POSSIBILITY OF INITIAL STATE GENERATION

To solve the terminal value problem of the BSPDE, it is necessary to prepare the (unnormalized) initial state $|\bar{V}(0)\rangle = \sum_k f_{\mathrm{pay}}(\boldsymbol{x}^{(k)})|k\rangle$, which we assumed to be given in Section IV-A. Here, we show by simulation that, for a typical f_{pay}, we can approximate the initial state $|\bar{V}(0)\rangle$ using an appropriate ansatz. To show that the initial state can be approximated by $|\nu(\boldsymbol{\theta}_0)\rangle = \alpha_0 R_0(\boldsymbol{\theta}_0)|0\rangle$, where $\alpha_0 = \sqrt{\sum_k f_{\mathrm{pay}}(\boldsymbol{x}^{(k)})^2}$ and $R_0(\boldsymbol{\theta}_0)$ is the ansatz, as shown in Fig. 1, we adopt L-BFGS-B to find $\boldsymbol{\theta}_0$ such that

$$\max_{\boldsymbol{\theta}_0} |\langle \bar{V}(0)|\nu(\boldsymbol{\theta}_0)\rangle|^2 \tag{70}$$

with SciPy [53]. For the calculation of the gradient, we use the parameter shift rule [54]. We choose the parameters as $K = 1$, $l = 0.5$, and $u = 2$, and the ansatz with six qubits and six layers. By doing maximization of (70), the value $\alpha^{-2}|\langle \bar{V}(0)|\nu(\boldsymbol{\theta}_0)\rangle|^2$, which corresponds to fidelity, should asymptotically converge to 1. The result for a payoff function of the single asset call option $f_{\mathrm{pay}}(x) = \max(x - K, 0)$ is

shown in Fig. 4. We can see that the ansatz approximates the payoff function well. Indeed, the result satisfies $|1-\alpha_0^{-2}|\langle\bar{\boldsymbol{V}}(0)|\nu(\boldsymbol{\theta}_0)\rangle|^2| \leq 1.2\times10^{-5}$.

Note that this optimization does not correspond to real physical operations. What we show is that there exists $\boldsymbol{\theta}_0$ that at least approximates $|\bar{\boldsymbol{V}}(0)\rangle$ well, and we leave the efficient search algorithm for such $\boldsymbol{\theta}_0$ to future work.

V. CONCLUSION

In this article, we simulate the BSPDE by VQS and obtain the state that embeds the solution of the BSPDE $|\boldsymbol{V}(t_{\text{ter}})\rangle$ at t_{ter}, and utilizing the fact that the derivative price is a martingale, we calculate the derivative price by the inner product of the state $|\boldsymbol{V}(t_{\text{ter}})\rangle$ and the state $|\boldsymbol{p}(t_{\text{ter}})\rangle$ that embeds the probability distribution. Although it is difficult to accurately estimate the complexity due to the heuristic nature of variational quantum computation, at least in the numerical simulation, we confirm that the proposed method can be performed for the one-asset double-barrier option and that the derivative price can be obtained with better accuracy by increasing the number of qubits and the number of layers of ansatz. We see that the computational complexity is obtained by Table 1 under certain assumptions, and the complexity with respect to ϵ is $O(1/\epsilon^2(\log(1/\epsilon))^d)$. This means that there would be a significant improvement compared with the classical FDM and conventional quantum algorithms whose complexity has factors, such as $(1/\epsilon)^{O(d)}$. Furthermore, we show that an oracle that generates an initial state with embedded payoff functions for typical payoff functions could be represented using an appropriate ansatz.

In this article, we simply assumed that the initial state of the BSPDE and the state with embedded probability distribution are effectively generated by some variational quantum algorithms. We will confirm this point in future work.

APPENDIX A
ELEMENTS OF THE MATRIX AND THE VECTOR OF THE FDM FOR THE BSPDE

Here, we show the concrete elements of $D_{x_i}^{1\text{st}}$ in (25), (24), $D_{x_i}^{2\text{nd}}$ in (24), and $\boldsymbol{C}$ in (21). $D_{x_i}^{1\text{st}}$ and $D_{x_i}^{2\text{nd}}$ are written by (71) and (72), shown at the bottom of this page. $\boldsymbol{C}(\tau)$ corresponds to the boundary conditions, and its elements $C_k(\tau)$ are

$$\begin{aligned}
C_k(\tau) = &\sum_{i=1}^{d}\frac{\sigma_i^2}{2h_i^2}\Big[(l_i+h_i)^2\delta_{k_i,0}\bar{V}_i^{\text{LB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})\\
&+(l_i+n_{\text{gr}}h_i)^2\delta_{k_i,n_{\text{gr}}-1}\bar{V}_i^{\text{UB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})\Big]\\
&+\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\frac{\sigma_i\sigma_j\rho_{ij}}{4h_ih_j}\\
&\times\Big[-(l_i+h_i)(l_j+(k_j+1)h_j)\delta_{k_i,0}\bar{V}_i^{\text{LB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})\\
&-(l_i+(k_i+1)h_i)(l_j+h_j)\delta_{k_j,0}\bar{V}_j^{\text{LB}}(\tau,\boldsymbol{x}_{\wedge j}^{(k)})\\
&+(l_i+n_{\text{gr}}h_i)(l_j+(k_j+1)h_j)\delta_{k_i,n_{\text{gr}}-1}\bar{V}_i^{\text{UB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})\\
&+(l_i+(k_i+1)h_i)(l_j+n_{\text{gr}}h_j)\delta_{k_j,n_{\text{gr}}-1}\bar{V}_j^{\text{UB}}(\tau,\boldsymbol{x}_{\wedge j}^{(k)})\Big]\\
&+r\sum_{i=1}^{d}\frac{1}{2h_i}\Big[(l_i+n_{\text{gr}}h_i)\delta_{k_i,n_{\text{gr}}-1}\bar{V}_i^{\text{UB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})\\
&-(l_i+h_i)\delta_{k_i,0}\bar{V}_i^{\text{LB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})\Big] \qquad (73)
\end{aligned}$$

where $\bar{V}_i^{\text{UB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)}) = V_i^{\text{UB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})$ and $\bar{V}_i^{\text{LB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)}) = V_i^{\text{LB}}(\tau,\boldsymbol{x}_{\wedge i}^{(k)})$.

$$D_{x_i}^{1\text{st}} = \begin{pmatrix}
0 & x_i^{(1)} & & & & \\
-x_i^{(0)} & 0 & x_i^{(2)} & & & \\
 & -x_i^{(1)} & 0 & x_i^{(3)} & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & -x_i^{(n_{\text{gr}}-3)} & 0 & x_i^{(n_{\text{gr}}-1)} \\
 & & & & -x_i^{(n_{\text{gr}}-2)} & 0
\end{pmatrix} \qquad (71)$$

$$D_{x_i}^{2\text{nd}} = \begin{pmatrix}
-2\left(x_i^{(0)}\right)^2 & \left(x_i^{(1)}\right)^2 & & & & \\
\left(x_i^{(0)}\right)^2 & -2\left(x_i^{(1)}\right)^2 & \left(x_i^{(2)}\right)^2 & & & \\
 & \left(x_i^{(1)}\right)^2 & -2\left(x_i^{(2)}\right)^2 & \left(x_i^{(3)}\right)^2 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \left(x_i^{(n_{\text{gr}}-3)}\right)^2 & -2\left(x_i^{(n_{\text{gr}}-2)}\right)^2 & \left(x_i^{(n_{\text{gr}}-1)}\right)^2 \\
 & & & & \left(x_i^{(n_{\text{gr}}-2)}\right)^2 & -2\left(x_i^{(n_{\text{gr}}-1)}\right)^2
\end{pmatrix} \qquad (72)$$

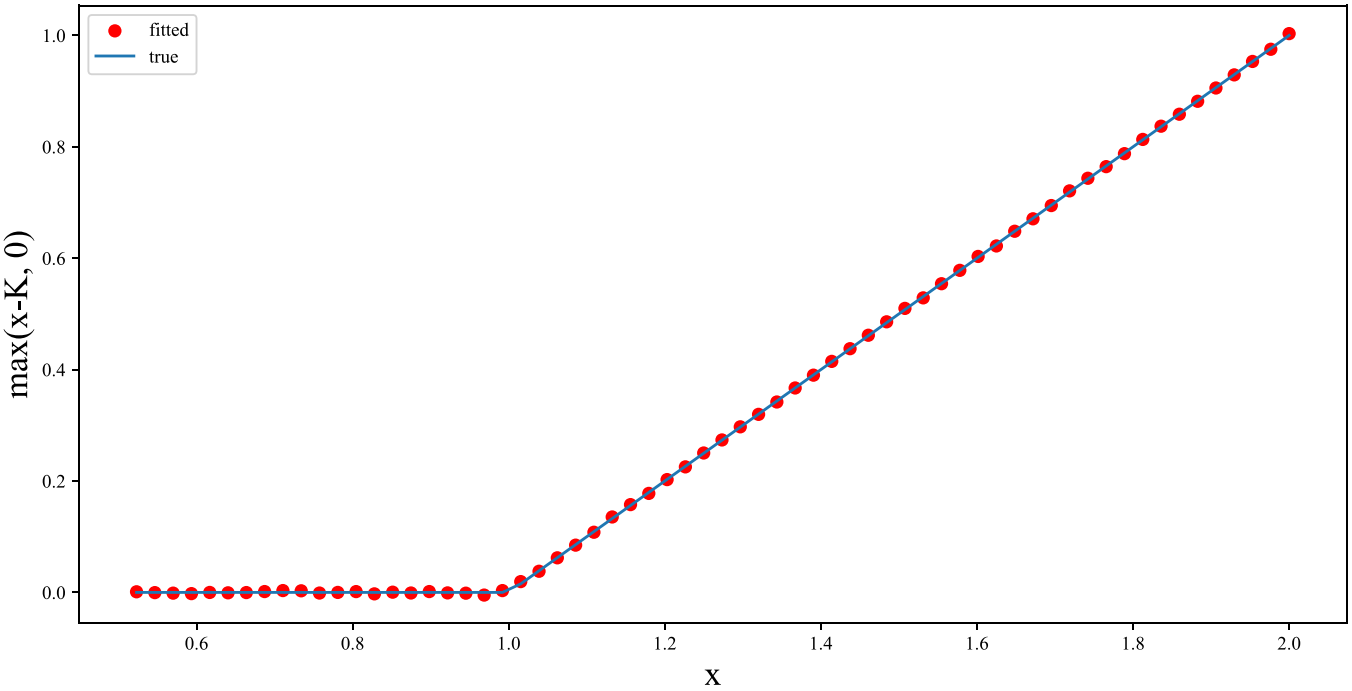

FIGURE 4. Target initial condition for single asset (solid line) and the initial state obtained by fidelity maximization (circle dots). Parameters are $f_{\text{pay}}(x) = \max(x - K, 0)$ and $K = 1$. The value corresponding to fidelity satisfies $|1 - \alpha_0^{-2}|\langle\tilde{V}(0)|u(\theta_0)\rangle|^2| \leq 1.2 \times 10^{-5}$.

APPENDIX B
DECOMPOSITION OF MATRICES

As discussed in Section III, we need to express F and $|C(\tau)\rangle$ in terms of linear combination of quantum gates to perform the VQS for the BSPDE. Here, we show that such decomposition is possible. The decomposition of F is based on the way, as shown in [22] and [47]. We also obtain a linear combination of quantum gates that generates $|C(\tau)\rangle$ by slightly modifying the decomposition of F. For simplicity, we assume $N_{\text{gr}} = 2^{nd}$ where n is the number of qubits. $D_{x_i}^{1\text{st}}$and$D_{x_i}^{2\text{nd}}$ in (25) and (24) are decomposed as follows:

$$D_{x_i}^{1\text{st}} = l_i D^{1\text{st}} + h_i \left(\text{Dec}(n)(J(n) + 2I^{\otimes n}) - \text{Inc}(n)\left(J(n) + I^{\otimes n}\right)\right) \tag{74}$$

$$D_{x_i}^{2\text{nd}} = l_i^2 D^{2\text{nd}} + 2l_i h_i \left(\text{Dec}(n) - 2I^{\otimes n} + \text{Inc}(n)\right)(J(n) + I) + h_i^2 \left(\text{Dec}(n) - 2I^{\otimes n} + \text{Inc}(n)\right)(J(n) + I)^2. \tag{75}$$

Here, we define

$$D^{1\text{st}} := -\text{Inc}(n) + \text{Dec}(n) \tag{76}$$

$$D^{2\text{nd}} := \text{Inc}(n) + \text{Dec}(n) - 2I^{\otimes n} \tag{77}$$

$$J(n) := \sum_{i=0}^{2^n-1} i|i\rangle\langle i| = \frac{2^n - 1}{2} I^{\otimes n} - \sum_{i=1}^{n} 2^{n-i-1} Z_i \tag{78}$$

$$\text{Inc}(n) := \sum_{i=0}^{2^n-2} |i+1\rangle\langle i| \tag{79}$$

$$\text{Dec}(n) := \sum_{i=1}^{2^n-1} |i-1\rangle\langle i| \tag{80}$$

where $Z_i := I^{\otimes i-1} \otimes Z \otimes I^{\otimes n-i}$. Inc$(n)$andDec$(n)$ are constructed by the following operators:

$$\text{CycInc}(n) := \sum_{i=0}^{2^n-1} |i+1\rangle\ \langle i| \tag{81}$$

$$\text{CycDec}(n) := \sum_{i=1}^{2^n-1} |i-1\rangle\ \langle i| \tag{82}$$

where we define $|-1\rangle := |2^n - 1\rangle$, $|2^n\rangle := |0\rangle$. CycInc(n) and CycDec(n) can be decomposed into a product of $O(n)$ Toffoli, CNOT, and X gates with $O(n)$ ancilla qubits [55]. With these circuits, we obtain

$$\text{Inc}(n) = \frac{1}{2}\text{CycInc}(n)(C^{n-1}Z + I^{\otimes n}) \tag{83}$$

$$\text{Dec}(n) = \frac{1}{2}(C^{n-1}Z + I^{\otimes n})\text{CycDec}(n). \tag{84}$$

$C^{n-1}Z := \sum_{i=0}^{2^n-2} |i\rangle\langle i| - |2^n - 1\rangle\langle 2^n - 1|$ is an n-qubit control Z gate and can be implemented as a product of $O(n^2)$ Toffoli, CNOT, and single-qubit gates [56]. We can express $D_{x_i}^{1\text{st}}$ and $D_{x_i}^{2\text{nd}}$ as the sums of $O(n^2)$ unitary operators, each of which is a product of $O(n^2)$ few-qubit gates. Then, the first term of (24) is a sum of $O(dn^2)$ operators, each of which is made by $O(n^2)$ few-qubit gates. The second term is the sum of $O(d^2n^4)$ unitary operators each of which is made by $O(n^2)$ few-qubit gates. From (24) and (25), we see that

F can eventually be expressed as a sum of $O(d^2n^4)$ unitary operators each of which is made by $O(n^2)$ few-qubit gates.

It is also necessary to construct a linear combination of unitary operators that outputs the quantum state $|\boldsymbol{C}(\tau)\rangle = \sum_{k=1}^{N_{\rm gr}} C_k(\tau)|k\rangle$. Here, we consider specific cases where $f_{\rm pay}(\boldsymbol{S}(T)) = \max(a_0 + \sum_{i=1}^{d} a_j S_j(T), 0)$, and some assets have knock-out conditions. These are the cases where the typical boundary conditions introduced in Section II-C are compounded. In these cases, we can write

$$|\boldsymbol{C}(\tau)\rangle = \tilde{G}|0\rangle = 2^{nd/2} G(\tau) H^{\otimes nd}|0\rangle \tag{85}$$

where

$$\begin{aligned}
&G(\tau)\\
&= \sum_{i=1}^{d} \frac{\sigma_i^2}{2h_i}\Big[(l_i+h_i)^2\, G_i^{(0)} B_i^{\rm LB}(\tau)\delta_i^{\rm UB}\\
&\quad + (l_i + n_{\rm gr}h_i)^2\, G_i^{(n_{\rm gr}-1)} B_i^{\rm UB}(\tau)\delta_i^{\rm UB}\Big]\\
&\quad + \sum_{i=1}^{d-1}\sum_{j=i+1}^{d} \frac{\sigma_i\sigma_j\rho_{ij}}{4h_ih_j}\\
&\quad \times \Big[-(l_i+h_i)(l_jI^{\otimes dn} + h_jJ_j(n))G_i^{(0)}B_i^{\rm LB}(\tau)\delta_i^{\rm LB}\\
&\quad - (l_iI^{\otimes dn} + h_iJ_i(n))(l_j+h_j)G_j^{(0)}B_j^{\rm LB}(\tau)\delta_i^{\rm LB}\\
&\quad + (l_i+n_{\rm gr}h_i)(l_jI^{\otimes dn} + h_jJ_j(n))G_i^{(n_{\rm gr}-1)}B_i^{\rm UB}(\tau)\delta_i^{\rm UB}\\
&\quad + (l_iI^{\otimes dn} + h_iJ_i(n))(l_j+n_{\rm gr}h_j)G_j^{(n_{\rm gr}-1)}B_j^{\rm UB}(\tau)\delta_i^{\rm UB}\Big]\\
&\quad + r\sum_{i=1}^{d}\frac{1}{2h_i}\Big[(l_i+n_{\rm gr}h_i)G_i^{(n_{\rm gr}-1)}B_i^{\rm UB}(\tau)\delta_i^{\rm LB}\\
&\quad - (l_i+h_i)G_i^{(0)}B_i^{\rm LB}(\tau)\delta_i^{\rm LB}\Big]
\end{aligned} \tag{86}$$

where

$$\delta_i^{\rm UB} = \begin{cases} 0 & \text{up and out barrier is set the ith asset} \\ 1 & \text{otherwise} \end{cases} \tag{87}$$

$$\delta_i^{\rm LB} = \begin{cases} 0 & \text{down and out barrier is set to the ith asset} \\ 1 & \text{otherwise} \end{cases} \tag{88}$$

and

$$G_i^{(0)} = I^{\otimes n(i-1)} \otimes |0\rangle\langle 0|^{\otimes n} \otimes I^{n(d-i)} \tag{89}$$

$$G_i^{(n_{\rm gr}-1)} = I^{\otimes n(i-1)} \otimes |1\rangle\langle 1|^{\otimes n} \otimes I^{n(d-i)} \tag{90}$$

$$B_i^{\rm UB}(\tau) = e^{-r\tau}a_0 I^{\otimes nd} + \sum_{1\le j\le d, j\ne i} a_j\left(l_jI^{\otimes nd} + h_jJ_j(n)\right) x + a_iu_iI^{\otimes nd} \tag{91}$$

$$B_i^{\rm LB}(\tau) = e^{-r\tau}a_0 I^{\otimes nd} + \sum_{1\le j\le d, j\ne i} a_j\left(l_jI^{\otimes nd} + h_jJ_j(n)\right) + a_il_iI^{\otimes nd} \tag{92}$$

$$J_i(n) = I^{\otimes n(i-1)} \otimes (J(n) + I^{\otimes n}) \otimes I^{\otimes n(d-i)} \tag{93}$$

where $|0\rangle\langle 0|^{\otimes n} = \frac{1}{2}(I^{\otimes n} - X^{\otimes n}\cdot C^nZ\cdot X^{\otimes n})$ and $|1\rangle\langle 1|^{\otimes n} = \frac{1}{2}(I^{\otimes n} + C^{n-1}Z)$. $G_i^{(0)}$ and $G_i^{(n_{\rm gr})}$ are expressed as a sum of $O(1)$ unitary operator each of which is made by $O(n^2)$ few-qubit gates. $B_i^{\rm UB}$ and $B_i^{\rm LB}$ are expressed as a sum of $O(dn)$ unitary operators, each of which is made by $O(n)$ few-qubit gates. Thus, $G(\tau)$ is a sum of $O(d^2\times n\times dn) = O(d^3n^2)$ unitary operators, each of which is made by $O(n^2)$ few-qubit gates.

APPENDIX C
VARIATIONAL PRINCIPLE FOR VQS

Here, we derive (34) from a variational principle. The square of the difference between both sides of (30) is

$$\begin{aligned}
&\left\|\frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle - L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle - |\boldsymbol{u}(t)\rangle\right\|^2\\
&= \left\|\frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle - L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right\|^2\\
&\quad - 2{\rm Re}\left[\langle \boldsymbol{u}(t)|\left(\frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle - L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right)\right] + \||\boldsymbol{u}(t)\rangle\|^2\\
&= \left\|\frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right\|^2 - 2{\rm Re}\left[\langle\tilde{v}(\boldsymbol{\theta}(t))|L(t)\frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right]\\
&\quad + \|L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle\|^2\\
&\quad - 2\Re\left[\langle \boldsymbol{u}(t)|\left(\frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle - L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right)\right] + \||\boldsymbol{u}(t)\rangle\|^2\\
&= 2\Re\sum_{j,k}\frac{d\langle\tilde{v}(\theta_j(t))|}{d\theta_j(t)}\frac{d|\tilde{v}(\theta_k(t))\rangle}{d\theta_k(t)}\dot{\theta}_j\dot{\theta}_k\\
&\quad - 2\Re\left[\frac{d\langle\tilde{v}(\boldsymbol{\theta}(t))|}{d\theta_j(t)}L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle + \frac{d\langle\tilde{v}(\boldsymbol{\theta}(t))|}{d\theta_j(t)}|\boldsymbol{u}(t)\rangle\right]\dot{\theta}_j\\
&\quad + 2\Re\left[\langle\boldsymbol{u}(t)|L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right] + \|L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle\|^2 + \||\boldsymbol{u}(t)\rangle\|^2.
\end{aligned} \tag{94}$$

Then, the first-order variation of the R.H.S. of (94) is

$$\begin{aligned}
&\delta\left\|\frac{d}{dt}|\tilde{v}(\boldsymbol{\theta}(t))\rangle - L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle - |\boldsymbol{u}(t)\rangle\right\|^2\\
&= 2\Re\sum_{j,k}\frac{d\langle\tilde{v}(\theta_j(t))|}{d\theta_j(t)}\frac{d|\tilde{v}(\theta_k(t))\rangle}{d\theta_k(t)}\dot{\theta}_j\delta\dot{\theta}_k\\
&\quad - 2\Re\left[\frac{d\langle\tilde{v}(\boldsymbol{\theta}(t))|}{d\theta_j(t)}L(t)|\tilde{v}(\boldsymbol{\theta}(t))\rangle + \frac{d\langle\tilde{v}(\boldsymbol{\theta}(t))|}{d\theta_j(t)}|\boldsymbol{u}(t)\rangle\right]\delta\dot{\theta}_j.
\end{aligned} \tag{95}$$

Thus, we obtain (34).

APPENDIX D
QUANTUM CIRCUITS TO EVALUATE $\mathcal{M}_{i,j}$ AND $\mathcal{V}_i$

Here, we show the quantum circuits to evaluate $\mathcal{M}_{i,j}$ and $\mathcal{V}_j$. Without the loss of generality, we can set $i \le j$. The terms in

IEEE Transactions on Quantum Engineering

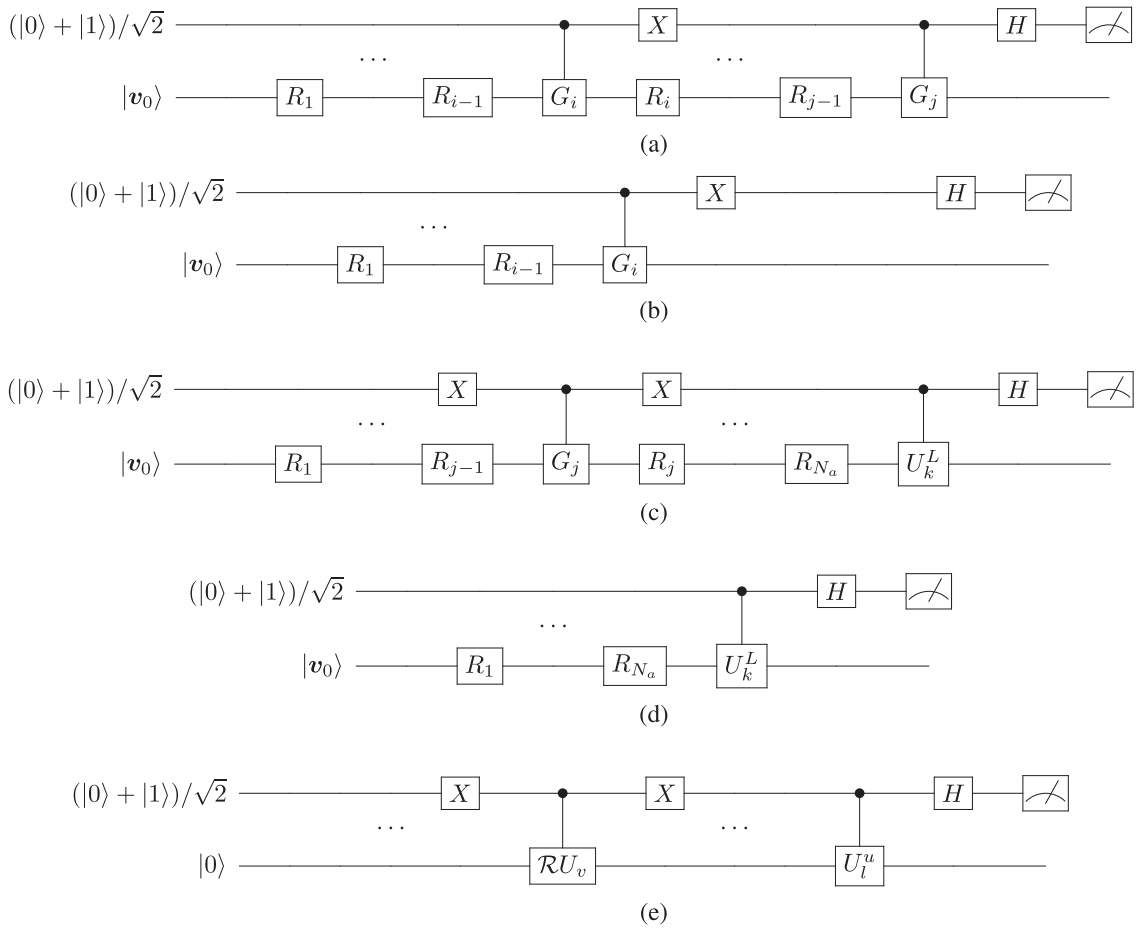

FIGURE 5. Quantum circuits for evaluating (a) (98), (b) (99), (c) (101), (d) (101), and (e) (103) and (104). $\mathcal{R}$ is $R_1 \cdots R_{i-1} G_i R_i \cdots R_{N_a}$ for (101) and $R_1 \cdots R_{i-1} G_i R_i \cdots R_{N_a}$ for (102), respectively [44].

(35) and (36) are written by (96)–(102), shown at the bottom of the page.

We can evaluate these terms using quantum circuits, as depicted in Fig. 5. Note that, although the quantum circuit evaluating (103) and (104) contains the control-$\mathcal{R}U_v$ gate, where

$$\mathcal{R} = R_1 \cdots R_{i-1} G_i R_i \cdots R_{N_a} \tag{103}$$

for (103) and $R_1 \cdots R_{i-1} G_i R_i \cdots R_{N_a}$ for (104), respectively, in the case where all boundary conditions are knock-out barriers, that is, in the case of $|\boldsymbol{C}(t)\rangle = \boldsymbol{0}$, we do not need to evaluate (101) and (102).

APPENDIX E
LOWER BOUND OF Ξ

Here, we evaluate the lower bound of Ξ and show that the Ξ does not decrease exponentially with respect to the number

$$\Re\left(\frac{\partial\langle\tilde{v}(\boldsymbol{\theta}(t))|}{\partial\theta_i}\frac{\partial|\tilde{v}(\boldsymbol{\theta}(t))\rangle}{\partial\theta_j}\right) = \begin{cases} \theta_0(t)^2\Re\left(\langle \boldsymbol{v}_0|R_1^\dagger\cdots R_{i-1}^\dagger G_i^\dagger R_i^\dagger\cdots R_{j-1}^\dagger G_j R_{j-1}\cdots R_1|\boldsymbol{v}_0\rangle\right) & 0 < \mathrm{i} \le \mathrm{j} \le \mathrm{N_a} \quad (96) \\ \theta_0(t)\Re\left(\langle \boldsymbol{v}_0|R_1^\dagger\cdots R_{j-1}^\dagger G_j^\dagger R_{j-1}\cdots R_1|\boldsymbol{v}_0\rangle\right) & 0 = \mathrm{i} < \mathrm{j} \le \mathrm{N_a} \quad (97) \\ 1 & \mathrm{i} = \mathrm{j} = 0 \quad (98) \end{cases}$$

$$\Re\left(\frac{\partial\langle\tilde{v}(\boldsymbol{\theta}(t))|}{\partial\theta_i}U_k^L|\tilde{v}(\boldsymbol{\theta}(t))\rangle\right) = \begin{cases} \theta_0(t)\Re\left(\langle \boldsymbol{v}_0|R_1^\dagger\cdots R_{i-1}^\dagger G_i^\dagger R_i^\dagger\cdots R_{N_a}^\dagger U_k^L R_{N_a}\cdots R_1|\boldsymbol{v}_0\rangle\right) & \mathrm{i} \ne 0 \quad (99) \\ \Re\left(\langle \boldsymbol{v}_0|R_1^\dagger\cdots R_{N_a}^\dagger U_k^L R_{N_a}\cdots R_1|\boldsymbol{v}_0\rangle\right) & \mathrm{i} = 0 \quad (100) \end{cases}$$

$$\Re\left(\frac{\partial\langle\tilde{v}(\boldsymbol{\theta}(t))|}{\partial\theta_m}U_l^u|0\rangle\right) = \begin{cases} \theta_0(t)\Re\left(\langle \boldsymbol{v}_0|R_1^\dagger\cdots R_{i-1}^\dagger G_i^\dagger R_i^\dagger\cdots R_{N_a}^\dagger U_l^u|0\rangle\right) & \mathrm{i} \ne 0 \quad (101) \\ \Re\left(\langle \boldsymbol{v}_0|R_1^\dagger\cdots R_{N_a}^\dagger U_l^u|0\rangle\right) & \mathrm{i} = 0. \quad (102) \end{cases}$$

of assets d. Using the inequality

$$\min(a^2, b^2) \leq \frac{1}{4}(a+b)^2 \tag{104}$$

for $a, b \in \mathbb{R}_+$, we obtain

$$t_{\text{ter}} \leq \min\left\{ \frac{2\left(\log \frac{u_i}{s_{i,0}}\right)^2}{25\sigma_i^2 \log \frac{2\tilde{A}d(d+1)}{\epsilon}}, \frac{2\left(\log \frac{s_{i,0}}{l_i}\right)^2}{25\sigma_i^2 \log \frac{2\tilde{A}d(d+1)}{\epsilon}} \right\}$$
$$\leq \frac{\left(\log \frac{u_i}{l_i}\right)^2}{50\sigma_i^2 \log \frac{2\tilde{A}d(d+1)}{\epsilon}} \tag{105}$$

for $i \in [d]$. From (57) and (105), Ξ is evaluated by

$$\Xi \geq \zeta B^2 \left(\frac{25}{4\pi} \log \frac{2\tilde{A}d(d+1)}{\epsilon}\right)^{d/2} \prod_{i=1}^{d} \frac{\frac{u_i}{l_i} - 1}{\log \frac{u_i}{l_i}}. \tag{106}$$

As easily verified by elementary analysis, for any $z > 1$

$$\frac{z-1}{\log z} \geq 1 \tag{107}$$

holds, and then, we obtain

$$\Xi \geq \zeta B^2 \left(\frac{25}{4\pi} \log \frac{2\tilde{A}d(d+1)}{\epsilon}\right)^{d/2}. \tag{108}$$

Thus, we can see that Ξ does not decrease exponentially with respect to d.

REFERENCES

[1] M. Hodson, B. Ruck, H. Ong, D. Garvin, and S. Dulman, "Portfolio rebalancing experiments using the quantum alternating operator ansatz," 2019, doi: 10.48550/arXiv.1911.05296.

[2] P. Rebentrost and S. Lloyd, "Quantum computational finance: Quantum algorithm for portfolio optimization," 2018, doi: 10.48550/arXiv.1811.03975.

[3] I. Kerenidis, A. Prakash, and D. Szilágyi, "Quantum algorithms for portfolio optimization," in *Proc. 1st ACM Conf. Adv. Financial Technol.*, 2019, pp. 147–155, doi: 10.1145/3318041.3355465.

[4] K. Miyamoto and K. Shiohara, "Reduction of qubits in a quantum algorithm for Monte Carlo simulation by a pseudo-random-number generator," *Phys. Rev. A*, vol. 102, Aug. 2020, Art. no. 022424, doi: 10.1103/PhysRevA.102.022424.

[5] S. Woerner and D. J. Egger, "Quantum risk analysis," *npj Quantum Inf.*, vol. 5, no. 1, 2019, Art. no. 15, doi: 10.1038/s41534-019-0130-6.

[6] D. J. Egger, R. G. Gutiérrez, J. C. Mestre, and S. Woerner, "Credit risk analysis using quantum computers," *IEEE Trans. Comput.*, vol. 70, no. 12, pp. 2136–2145, Dec. 2021, doi: 10.1109/TC.2020.3038063.

[7] K. Kaneko, K. Miyamoto, N. Takeda, and K. Yoshino, "Quantum speedup of Monte Carlo integration with respect to the number of dimensions and its application to finance," *Quantum Inf. Process.*, vol. 20, no. 5,2021, Art. no. 185, doi: 10.1007/s11128-021-03127-8.

[8] K. Miyamoto, "Quantum algorithm for calculating risk contributions in a credit portfolio," *EPJ Quantum Technol.*, vol. 9, 2022, Art. no. 32, doi: 10.1140/epjqt/s40507-022-00153-y.

[9] P. Rebentrost, B. Gupt, and T. R. Bromley, "Quantum computational finance: Monte Carlo pricing of financial derivatives," *Phys. Rev. A*, vol. 98, Aug. 2018, Art. no. 022321, doi: 10.1103/PhysRevA.98.022321.

[10] A. Martin et al., "Toward pricing financial derivatives with an IBM quantum computer," *Phys. Rev. Res.*, vol. 3, Feb. 2021, Art. no. 013167, doi: 10.1103/PhysRevResearch.3.013167.

[11] N. Stamatopoulos et al., "Option pricing using quantum computers," *Quantum*, vol. 4, Jul. 2020, Art. no. 291, doi: 10.22331/q-2020-07-06-291.

[12] S. Ramos-Calderer et al., "Quantum unary approach to option pricing," *Phys. Rev. A*, vol. 103, Mar. 2021, Art. no. 032414, doi: 10.1103/PhysRevA.103.032414.

[13] F. Fontanela, A. Jacquier, and M. Oumgari, "Short communication: A quantum algorithm for linear PDEs arising in finance," *SIAM J. Financial Math.*, vol. 12, no. 4, pp. SC98–SC114, 2021, doi: 10.1137/21M1397878.

[14] S. K. Radha, "Quantum option pricing using wick rotated imaginary time evolution," 2021, doi: 10.48550/arXiv:2101.04280.

[15] J. Gonzalez-Conde, Á. Rodríguez-Rozas, E. Solano, and M. Sanz, "Simulating option price dynamics with exponential quantum speedup," 2022, *arXiv:2101.04023*, doi: 10.48550/arXiv.2101.04023.

[16] A. C. Vazquez and S. Woerner, "Efficient state preparation for quantum amplitude estimation," *Phys. Rev. Appl.*, vol. 15, Mar. 2021, Art. no. 034027, doi: 10.1103/PhysRevApplied.15.034027.

[17] H. Tang, A. Pal, T.-Y. Wang, L.-F. Qiao, J. Gao, and X.-M. Jin, "Quantum computation for pricing the collateralized debt obligations," *Quantum Eng.*, vol. 3, no. 4, 2021, Art. no. e84, doi: 10.1002/que2.84.

[18] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, "A threshold for quantum advantage in derivative pricing," *Quantum*, vol. 5,Jun. 2021, Art. no. 463, doi: 10.22331/q-2021-06-01-463.

[19] D. An, N. Linden, J.-P. Liu, A. Montanaro, C. Shao, and J. Wang, "Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance," *Quantum*, vo. 5, Jun. 2021, Art. no. 481, doi: 10.22331/q-2021-06-24-481.

[20] K. Kaneko, K. Miyamoto, N. Takeda, and K. Yoshino, "Quantum pricing with a smile: Implementation of local volatility model on quantum computer," *EPJ Quantum Technol.*, vol. 9, no. 1, 2022, Art. no. 7, doi: 10.1140/epjqt/s40507-022-00125-2.

[21] K. Miyamoto, "Bermudan option pricing by quantum amplitude estimation and Chebyshev interpolation," *EPJ Quantum Technol.*, vol. 9, no. 1, 2022, Art. no. 3, doi: 10.1140/epjqt/s40507-022-00124-3.

[22] H. Alghassi, A. Deshmukh, N. Ibrahim, N. Robles, S. Woerner, and C. Zoufal, "A variational quantum algorithm for the Feynman-Kac formula," *Quantum*, vol. 6, 2022, Art. no. 730, doi: 10.22331/q-2022-06-07-730.

[23] D. J. Egger et al., "Quantum computing for finance: State-of-the-art and future prospects," *IEEE Trans. Quantum Eng.*, vol. 1, 2020, Art. no. 3101724, doi: 10.1109/TQE.2020.3030314.

[24] R. Orús, S. Mugel, and E. Lizaso, "Quantum computing for finance: Overview and prospects," *Rev. Phys.*, vol. 4, 2019, Art. no. 100028, doi: 10.1016/j.revip.2019.100028.

[25] A. Bouland, W. van Dam, H. Joorati, I. Kerenidis, and A. Prakash, "Prospects and challenges of quantum finance," 2020, doi: 10.48550/arXiv.2011.06492.

[26] D. Herman et al., "A survey of quantum computing for finance," 2022, doi: 10.48550/arXiv:2201.02773.

[27] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *J. Political Econ.*, vol. 81, no. 3, pp. 637–654, 1973, doi: 10.1086/260062.

[28] J. Hull, *Options, Futures, and Other Derivatives*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2012.

[29] S. E. Shreve, *Stochastic Calculus for Finance I: Continuous-Time Models.* ser. Springer Finance Textbooks. New York, NY, USA: Springer, 2004, doi: 10.1007/978-0-387-22527-2.

[30] S. E. Shreve, *Stochastic Calculus for Finance II: Continuous-Time Models.* ser. Springer Finance Textbooks. New York, NY, USA: Springer, 2004. [Online]. Available: https://link.springer.com/book/9780387401010

[31] K. Miyamoto and K. Kubo, "Pricing multi-asset derivatives by finite-difference method on a quantum computer," *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 3100225, doi: 10.1109/TQE.2021.3128643.

[32] M. Broadie, P. Glasserman, and S. Kou, "A continuity correction for discrete barrier options," *Math. Finance*, vol. 7, no. 4, pp. 325–349, 1997, doi: 10.1111/1467-9965.00035.

[33] A. M. Childs, J.-P. Liu, and A. Ostrander, "High-precision quantum algorithms for partial differential equations," *Quantum*, vol. 5,Nov. 2021, Art. no. 574, doi: 10.22331/q-2021-11-10-574.

[34] I. Babuska and M. Suri, "The h-p version of the finite element method with quasiuniform meshes," *M2AN-Model. Math. Anal. Numer.*, vol. 21, no. 2, pp. 199–238, 1987. Available: http://www.numdam.org/item/M2AN_1987__21_2_199_0/

[35] J. Shen, T. Tang, and L.-L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, vol. 41, ser. Springer Series in Computational Mathematics. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-540-71041-7.

[36] H.-J. Bungartz and M. Griebel, "Sparse grids," *Acta Numerica*, vol. 13, pp. 147–269, 2004, doi: 10.1017/S0962492904000182.

[37] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *Quantum Comput. Quantum Inf.*, vol. 305, pp. 53–74, 2002, doi: 10.48550/arXiv.quant-ph/0005055.

[38] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, "Quantum algorithm for linear differential equations with exponentially improved dependence on precision," *Commun. Math. Phys.*, vol. 356, no. 3, pp. 1057–1081, 2017, doi: 10.1007/s00220-017-3002-y.

[39] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Inf.*, vol. 5, no. 1,2019, Art. no. 103, doi: 10.1038/s41534-019-0223-2.

[40] H. Situ, Z. He, Y. Wang, L. Li, and S. Zheng, "Quantum generative adversarial network for generating discrete distribution," *Inf. Sci.*, vol. 538, pp. 193–208, 2020, doi: 10.1016/j.ins.2020.05.127.

[41] S. Lloyd and C. Weedbrook, "Quantum generative adversarial learning," *Phys. Rev. Lett.*, vol. 121, no. 4, 2018, Art. no. 040502, doi: 10.1103/PhysRevLett.121.040502.

[42] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," *Phys. Rev. A*, vol. 98, no. 1, 2018, Art. no. 012324, doi: 10.1103/PhysRevA.98.012324.

[43] O. Kyriienko, A. E. Paine, and V. E. Elfving, "Protocols for trainable and differentiable quantum generative modelling," 2022, doi: 10.48550/arXiv.2202.08253.

[44] S. Endo, J. Sun, Y. Li, S. C. Benjamin, and X. Yuan, "Variational quantum simulation of general processes," *Phys. Rev. Lett.*, vol. 125, no. 1, 2020, Art. no. 010501, doi: 10.1103/PhysRevLett.125.010501.

[45] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, "Theory of variational quantum simulation," *Quantum*, vol. 3, Oct. 2019, Art. no. 191, doi: 10.22331/q-2019-10-07-191.

[46] M. Cerezo et al., "Variational quantum algorithms," *Nature Rev. Phys.*, vol. 3, no. 9, pp. 625–644, 2021, doi: 10.1038/s42254-021-00348-9.

[47] K. Kubo, Y. O. Nakagawa, S. Endo, and S. Nagayama, "Variational quantum simulations of stochastic differential equations," *Phys. Rev. A*, vol. 103, no. 5, 2021, Art. no. 052425, doi: 10.1103/PhysRevA.103.052425.

[48] J. C. Garcia-Escartin and P. Chamorro-Posada, "SWAP test and Hong-Ou-Mandel effect are equivalent," *Phys. Rev. A*, vol. 87, May 2013, Art. no. 052330, doi: 10.1103/PhysRevA.87.052330.

[49] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes With Source Code CD-ROM: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007. [Online]. Available: http://numerical.recipes/

[50] A. D. McLachlan, "A variational solution of the time-dependent Schrodinger equation," *Mol. Phys.*, vol. 8, no. 1, pp. 39–44, 1964, doi: 10.1080/00268976400100041.

[51] N. Kunitomo and M. Ikeda, "Pricing options with curved boundaries," *Math. Finance*, vol. 2, no. 4, pp. 275–298, 1992, doi: 10.1111/j.1467-9965.1992.tb00033.x.

[52] C. R. Harris et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[53] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.

[54] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A*, vol. 98, no. 3, 2018, Art. no. 032309, doi: 10.1103/PhysRevA.98.032309.

[55] X. Li, G. Yang, C. M. Torres, D. Zheng, and K. L. Wang, "A class of efficient quantum incrementer gates for quantum circuit synthesis," *Int. J. Modern Phys. B*, vol. 28, no. 1, 2014, Art. no. 1350191, doi: 10.1142/S0217979213501919.

[56] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge, U.K.: Cambridge Univ. Press, 2010, doi: 10.1017/CBO9780511976667.