Pricing multi-asset derivatives by finite difference method on a quantum computer

Koichi Miyamoto[1,*] and Kenji Kubo[2,3]
[1]Center for Quantum Information and Quantum Biology, Osaka University, Japan
[2]R4D, Mercari Inc., Japan
[3]Graduate School of Engineering Science, Osaka University, Japan

Following the recent great advance of quantum computing technology, there are growing interests in its applications to industries, including finance. In this paper, we focus on derivative pricing based on solving the Black-Scholes partial differential equation by finite difference method (FDM), which is a suitable approach for some types of derivatives but suffers from the *curse of dimensionality*, that is, exponential growth of complexity in the case of multiple underlying assets. We propose a quantum algorithm for FDM-based pricing of multi-asset derivative with exponential speedup with respect to dimensionality compared with classical algorithms. The proposed algorithm utilizes the quantum algorithm for solving differential equations, which is based on quantum linear system algorithms. Addressing the specific issue in derivative pricing, that is, extracting the derivative price for the present underlying asset prices from the output state of the quantum algorithm, we present the whole of the calculation process and estimate its complexity. We believe that the proposed method opens the new possibility of accurate and high-speed derivative pricing by quantum computers.

I. INTRODUCTION

The recent advance of quantum computing is invoking a strong interest in its applications to industries, including finance. Since large banks perform enormous computational tasks in their daily business, it is expected that quantum speedup of them will make a large impact. In fact, some recent papers have already discussed applications of quantum algorithms to concrete problems in financial engineering such as derivative pricing [2–5]. See [6, 7] as comprehensive reviews.

In this paper, we focus on the derivative pricing method based on solving the partial differential equation (PDE) by finite difference method (FDM). Let us describe the outline of the problem. First of all, a *financial derivatives*, or simply a *derivatives* is a contract between two parties, in which amounts (payoffs) determined by prices of some underlying assets (e.g., stocks and bonds) are paid and received. One simple example is an European call (resp. put) option, the right to buy (resp. sell) some asset at the predetermined price (*strike*) K and time (*maturity*) T. This is equivalent to the contract that the option buyer receives the payoff $f_{\rm pay}(S_T) = \max\{S_T - K, 0\}$ (resp. $\max\{K - S_T, 0\}$) at T, where S_t is the underlying asset price at time t. Since large banks hold a large number of derivatives, pricing them is crucial for their business. We can evaluate a derivative price by modeling the random time evolution of underlying asset prices as some stochastic processes and calculating the expected values of the discounted payoff under some probability measure. Analytical formulas for derivative prices are available only in limited settings, and therefore we often resort to numerical methods. Since the expected value obeys the so-called Black-Scholes (BS) PDE, we can obtain the derivative price by solving it [8]. More concretely, starting from the maturity T, at which the derivative price is trivially determined as the payoff itself, we solve the PDE backward to the present, and then find the present derivative price. This PDE approach is suitable for derivatives whose price is subject to some continuous *boundary conditions*. One example is the *barrier option*. In this product, one or multiple levels of underlying asset prices (*barriers*) are set, and they determine whether the payoff is paid at the maturity or not. For example, in a *knock-out* barrier option, the payoff is not paid if either of barriers is reached once or more by T. This means that the price of the knock-out barrier option is 0 at barriers. Such a boundary condition is difficult to be strictly taken into account in the other approach such as the Monte Carlo method because of discretization in time, but can be dealt with in the PDE approach.

Although the PDE approach is suitable in some cases, it is difficult to apply it to *multi-asset derivatives*, where the number of underlying assets d is larger than 1, because of the *curse of dimensionality*, the exponential growth of complexity with respect to d. We can see this as follows. The BS PDE is $(d+1)$-dimensional, where d and 1 correspond to asset prices and time, respectively. In FDM, which is often adopted for solving a PDE numerically, the discretization grid points are set in the asset price directions, and partial derivatives are replaced with matrices which correspond to finite difference approximation. This converts a PDE into a linear ordinary differential equation (ODE) system, in which the dependent variables are the derivative prices on grid points and the independent variable is time. Then, we solve the resulting ODE system. The point is that this calculation contains manipulations of the matrices with exponentially large size, which scales on the desired accuracy ϵ as $O((1/\epsilon)^{\mathrm{poly}(d)})$, and so does the time complexity. This makes the PDE approach, at least in combination with FDM, intractable on classical computers.

Fortunately, quantum computers might change the situation. There are some quantum algorithms for solving linear ODE systems, whose time complexities depend on dimensionality only logarithmically[9–12]. This means that, in combination with these algorithms, we can remove the exponential dependency of time complexity of FDM-based PDE

[*] koichi.miyamoto@qiqb.osaka-u.ac.jp
This document is the short version of the full paper [1]. This work was supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant Number JPMXS0120319794.

solving on dimensionality. Then, this paper aims to speedup FDM-based pricing of multi-asset derivatives, using the quantum algorithm. Although one might think that this is just a straightforward application of an existing algorithm to some problem, there is a nontrivial issue specific for derivative pricing, that is, how to extract the derivative price for the present underlying asset prices from the output of the quantum algorithm. By solving the BS PDE up to the present ($t = 0$) using the quantum algorithm, we obtain the vector $\vec{V}(0)$, which consists of the present derivative prices on the grid points. However, it is given not as classical data but as a quantum state $|\vec{V}(0)\rangle$, in which the elements of $\vec{V}(0)$ are encoded as amplitudes of computational basis states. On the other hand, typically, we are interested in only one element V_0 in $\vec{V}(0)$, which corresponds to the derivative price for the present underlying asset prices. That is, we want only the amplitude of the specific computational basis state in $|\vec{V}(0)\rangle$. Since the amplitude is exponentially small if there are exponentially many grid points, reading it out requires exponentially large time complexity, which ruins the quantum speedup.

We circumvent this issue by solving PDE up to not the present but some future time $t_{\rm ter}$. The key observation is that V_0 can be expressed as the expected value of its discounted price at an arbitrary future time. Concretely, we may take the following way. First, we generate two states: $|\vec{V}(t_{\rm ter})\rangle$, in which the derivative prices at $t_{\rm ter}$ are encoded, and $|\vec{p}(t_{\rm ter})\rangle$, in which the probability distribution of underlying asset prices at $t_{\rm ter}$ is encoded. Then, we estimate the inner product $\langle\vec{p}(t_{\rm ter})|\vec{V}(t_{\rm ter})\rangle$, which is an approximation of V_0. Note that, in this way, the amplitudes of all basis states in $|\vec{V}(t_{\rm ter})\rangle$ are used to calculate V_0. This is in contrast to read-out from $|\vec{V}(0)\rangle$, in which the amplitude of one specific basis state is the sole necessary information. This leads to much smaller time complexity in the above way than reading V_0 out from $|\vec{V}(0)\rangle$.

In the following sections, we describe the entire process of the above calculation: setting $t_{\rm ter}$, generating $|\vec{V}(t_{\rm ter})\rangle$ by the quantum algorithm, generating $|\vec{p}(t_{\rm ter})\rangle$, and estimating V_0. Besides, we estimate the complexity of the proposed method. We see that, in the expression of the complexity, there are not any factors like $(1/\epsilon)^{{\rm poly}(d)}$ but only some logarithmic factors to the power of d, which means substantial speedup compared with classical FDM.

The rest of this paper is organized as follows. Sections II and III are preliminary ones, which outline FDM-based derivative pricing and the quantum algorithm for solving ODE systems, respectively. Section IV discusses approximating V_0 as the expected value of the price at $t_{\rm ter}$, including its optimal setting. Section V presents the main result, that is, the quantum calculation procedure for V_0 and its complexity. Section VI summarizes this paper. All proofs are shown in [1].

Here, we explain the notations used in this paper. $\mathbb{R}_+ := \{x \in \mathbb{R} \,|\, x > 0\}$. For $n \in \mathbb{N}$, $[n] := \{1, ..., n\}$ and $[n]_0 := \{0\} \cup [n]$. For $n \in \mathbb{N}$, I_n denotes the $n\times n$ identity matrix. $\|\cdot\|$ denotes the Euclidian norm for a vector and the spectral norm for a matrix. We call each of them a "norm" simply. When a matrix A has at most s nonzero entries in any row and column, we say that the sparsity of A is s. For $\vec{v} = (v_1, ..., v_n)^T \in \mathbb{R}^n$, where n is an integer not less than 2, and $i \in [n]$, we define $\vec{v}_{\wedge i} \in \mathbb{R}^{n-1}$ as the vector made by removing the i-th entry v_i from $\vec{v}$.

In this paper, we consider quantum states of systems consisting of some quantum registers with some qubits. For a real number x, $|x\rangle$ denotes one of the computational basis states on some register, whose bit string corresponds to the binary representation of x. For $i \in \{0, 1\}$, we let $|i\rangle$ and $|\bar{i}\rangle$ denote a state on a multi-qubit register and a state on one qubit, respectively, in order to distinguish them. For $\vec{x} := (x_1, ..., x_d)^T \in \mathbb{R}^d$, $|\vec{x}\rangle$ denotes the (unnormalized) state in which the elements of $\vec{x}$ are encoded in the amplitudes of computational basis states, that is, $|\vec{x}\rangle := \sum_{i=1}^{d} x_i |i\rangle$. For a (unnormalized) state $|\psi\rangle$, its norm is defined as $\| |\psi\rangle \| := \sqrt{\langle\psi|\psi\rangle}$. If a state $|\psi\rangle$ satisfies $\| |\psi\rangle - |\psi'\rangle \| < \epsilon$, where ϵ is a positive real number and $|\psi'\rangle$ is another state, we say that $|\psi\rangle$ is ϵ-close to $|\psi'\rangle$.

II. FDM-BASED DERIVATIVE PRICING

A. Derivative pricing problem and the Black-Scholes PDE

In this paper, we consider the following problem.

Problem 1. *Let d be a positive integer and $T, U_1, ..., U_d, L_1, ..., L_d$ be positive real numbers such that $L_i < U_i$ for $i \in [d]$. Define $D := (L_1, U_1) \times \cdots \times (L_d, U_d)$, $\bar{D} := [L_1, U_1] \times \cdots \times [L_d, U_d]$ and $\hat{D}^i := [L_1, U_1] \times \cdots \times [L_{i-1}, U_{i-1}] \times [L_{i+1}, U_{i+1}] \times \cdots \times [L_d, U_d]$ for $i \in [d]$. Assume that a function $V : [0, T] \times \bar{D} \to \mathbb{R}$ satisfies the following PDE*

$$\frac{\partial}{\partial t}V(t,\vec{S}) + \frac{1}{2}\sum_{i,j=1}^{d}\sigma_i\sigma_j\rho_{ij}S_iS_j\frac{\partial^2}{\partial S_i\partial S_j}V(t,\vec{S}) + r\left(\sum_{i=1}^{d}S_i\frac{\partial}{\partial S_i}V(t,\vec{S}) - V(t,\vec{S})\right) = 0, \tag{1}$$

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

$$\begin{aligned} &V(T,\vec{S}) = f_{\rm pay}(\vec{S}), \\ &V(t,(S_1,...,S_{i-1},U_i,S_{i+1},...,S_d)^T) = V_i^{\rm UB}(t,\vec{S}_{\wedge i}) \text{ for } i \in [d], \\ &V(t,(S_1,...,S_{i-1},L_i,S_{i+1},...,S_d)^T) = V_i^{\rm LB}(t,\vec{S}_{\wedge i}) \text{ for } i \in [d]. \end{aligned} \tag{2}$$

Here, $t \in [0, T]$, $\vec{S} := (S_1, ..., S_d)^T \in D$, $\sigma_1, ..., \sigma_d, r$ are positive real constants such that $r < \frac{1}{2}\sigma_i^2$ for $i \in [d]$, $\rho_{ij}, i, j \in [d]$ are real constants such that $\rho_{11} = \cdots = \rho_{dd} = 1$ and the matrix $\rho := (\rho_{ij})_{1\le i,j\le d}$ is symmetric and positive-definite, and $f_{\rm pay} : D \to \mathbb{R}$, $V_i^{\rm UB} : [0, T] \times \hat{D}^i \to \mathbb{R}$ and $V_i^{\rm LB} : [0, T] \times \hat{D}^i \to \mathbb{R}$ are given functions. Then, for a given $\vec{S}_0 := (S_{1,0}, ..., S_{d,0})^T \in D$, find $V_0 := V(0, \vec{S}_0)$.

Here, we make some comments. (1) is the so-called BS PDE, which corresponds to the following derivative pricing problem. Under some probability space $(\Omega, \mathcal{F}, P)$, we consider the d-dimensional stochastic process $\vec{S}(t) :=$

$(S_1(t),...,S_d(t))^T, t \geq 0$ obeying the following stochastic differential equation (SDE) system

$$dS_i(t) = rS_i(t)dt + \sigma_i S_i(t)dW_i(t), i \in [d] \tag{3}$$

where $W_1,...,W_d$ are the Brownian motions on $(\Omega, \mathcal{F}, P)$ satisfying $dW_i dW_j = \rho_{ij}dt$ for $i, j \in [d]$, with the initial value $\vec{S}(0) = \vec{S}_0$. $S_1,...,S_d$ correspond to prices of d underlying assets and (3) describes the random time evolution of $\vec{S}(t)$ under the so-called risk-neutral measure, where any asset price grows with the risk-free rate r in expectation. σ_i is the parameter called *volatility*, which parameterizes how volatile the random movement of S_i is. This is the so-called BS model. Then, the derivative price is given by the conditional expected value of the payoff discounted by the risk-free rate. That is, the price of the derivative in which the payoff $f_{\rm pay}(\vec{S}(T))$ arises at maturity T is

$$V(t,\vec{S}) = E[e^{-r(T-t)} f_{\rm pay}(\vec{S}(T)) 1_{\rm NB} | \vec{S}(t) = \vec{S}] \tag{4}$$

at time t, if $\vec{S}(t) = \vec{S}$. Here, $1_{\rm NB}$ takes 1 if the condition for the payoff to be paid (e.g., barrier condition) is satisfied and 0 otherwise. It is known that $V(t,\vec{S})$ satisfies (1) and appropriate boundary conditions, which should be set according to the product characteristics of the derivative. For example, if U_i (resp. L_i) is a knock-out barrier, $V_i^{\rm UB} = 0$ (resp. $V_i^{\rm LB} = 0$).

For a later convenience, we here transform the PDE (1) on $[0,T) \times D$ into

$$\frac{\partial}{\partial\tau} Y(\tau,\vec{x}) = \left(\frac{1}{2}\sum_{i,j=1}^{d} \sigma_i\sigma_j\rho_{ij}\frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{d}\left(r - \frac{1}{2}\sigma_i^2\right)\frac{\partial}{\partial x_i}\right) Y(\tau,\vec{x}) \tag{5}$$

on $(0,T] \times \tilde{D}$, where $\tau := T - t, \vec{x} := (x_1,...,x_d)^T := (\log S_1,...,\log S_d)^T$, $Y(\tau,\vec{x}) := e^{r\tau} V(T-\tau,(e^{x_1},...,e^{x_d})^T)$, $\tilde{D} := (l_1,u_1) \times \cdots \times (l_d,u_d)$ and $u_i := \log U_i, l_i := \log L_i$ for $i \in [d]$. The boundary conditions become

$$\begin{aligned}
&Y(0,\vec{x}) = \tilde{f}_{\rm pay}(\vec{x}) := f_{\rm pay}((e^{x_1},...,e^{x_d})^T),\\
&Y(\tau,(x_1,...,x_{i-1},u_i,x_{i+1},...,x_d)^T)\\
&\quad = Y_i^{\rm UB}(\tau,\vec{x}_{\wedge i}) := V_i^{\rm UB}(T-\tau,(e^{x_1},...,e^{x_{i-1}},e^{x_{i+1}},...,e^{x_d})^T),\\
&Y(\tau,(x_1,...,x_{i-1},l_i,x_{i+1},...,x_d)^T)\\
&\quad = Y_i^{\rm LB}(\tau,\vec{x}_{\wedge i}) := V_i^{\rm LB}(T-\tau,(e^{x_1},...,e^{x_{i-1}},e^{x_{i+1}},...,e^{x_d})^T).
\end{aligned} \tag{6}$$

B. Application of FDM to the BS PDE

FDM is a method for solving a PDE by replacing partial derivatives with finite difference approximations. In the case of (5), the approximation is as follows. First, letting $n_{\rm gr}$ be a positive integer, we introduce the grid points in the directions of $\vec{x}$: for $k = \sum_{i=1}^{d} n_{\rm gr}^{d-i} k_i + 1$ with $k_1,...,k_d \in [n_{\rm gr}-1]_0$,

$$\vec{x}^{(k)} := (x_1^{(k_1)},...,x_d^{(k_d)})^T, x_i^{(k_i)} := l_i + (k_i+1)h_i, h_i := \frac{u_i - l_i}{n_{\rm gr}+1}. \tag{7}$$

Namely, there are $n_{\rm gr}$ equally spaced grid points in one direction and the total number of the grid points in D is $N_{\rm gr} := n_{\rm gr}^d$, except ones on the boundaries. For later convenience, we set $x_i^{(-1)} = l_1$ and $x_i^{(n_{\rm gr})} = h_1$. Hereafter, we assume that $n_{\rm gr}$ is a power of 2 for simplicity, whose detail is explained in Section V, and define $m_{\rm gr} := \log_2 n_{\rm gr}$.

Then, (5) is transformed into the $N_{\rm gr}$-dimensional ODE system

$$\frac{d}{d\tau}\vec{\tilde{Y}}(\tau) = F\vec{\tilde{Y}}(\tau) + \vec{C}(\tau). \tag{8}$$

with the initial value

$$\begin{aligned}
\vec{\tilde{Y}}(0) &= (Y(0,\vec{x}^{(1)}),...,Y(0,\vec{x}^{(N_{\rm gr})}))^T\\
&= (\tilde{f}_{\rm pay}(\vec{x}^{(1)}),...,\tilde{f}_{\rm pay}(\vec{x}^{(N_{\rm gr})}))^T =: \vec{\tilde{f}}_{\rm pay}.
\end{aligned} \tag{9}$$

Here, $\vec{\tilde{Y}}(\tau), F$ and $\vec{C}(\tau)$ are as follows. $\vec{\tilde{Y}}(\tau) := (\tilde{Y}_1(\tau),...,\tilde{Y}_{N_{\rm gr}}(\tau)) \in \mathbb{R}^{N_{\rm gr}}$ and its k-th element is an approximation of $Y(\tau,x^{(k)})$. F is a $N_{\rm gr} \times N_{\rm gr}$ real matrix, which is expressed by a sum of Kronecker products of $n_{\rm gr} \times n_{\rm gr}$ matrices, that is,

$$\begin{aligned}
F := &\sum_{i=1}^{d}\frac{\sigma_i^2}{2h_i^2} I_{n_{\rm gr}}^{\otimes i-1} \otimes D^{\rm 2nd} \otimes I_{n_{\rm gr}}^{\otimes d-i}\\
&+ \sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\frac{\sigma_i\sigma_j\rho_{ij}}{4h_ih_j} I_{n_{\rm gr}}^{\otimes i-1} \otimes D^{\rm 1st} \otimes I_{n_{\rm gr}}^{\otimes j-i-1} \otimes D^{\rm 1st} \otimes I_{n_{\rm gr}}^{\otimes d-j}\\
&+ \sum_{i=1}^{d}\frac{1}{2h_i}\left(r - \frac{1}{2}\sigma_i^2\right) I_{n_{\rm gr}}^{\otimes i-1} \otimes D^{\rm 1st} \otimes I_{n_{\rm gr}}^{\otimes d-i},
\end{aligned} \tag{10}$$

$$D^{\rm 1st} := \begin{pmatrix} 0 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ & & & & -1 & 0 \end{pmatrix}, D^{\rm 2nd} := \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix}. \tag{11}$$

$\vec{C}(\tau) := (C_1(\tau),...,C_{N_{\rm gr}}(\tau))^T$ is necessary to take into account the boundary conditions and its k-th element is

$$\begin{aligned}
C_k(\tau) = &\sum_{i=1}^{d}\frac{\sigma_i^2}{2h_i^2}\left[\delta_{k_i,0}Y_i^{\rm LB}(\tau,\vec{x}_{\wedge i}^{(k)}) + \delta_{k_i,n_{\rm gr}-1}Y_i^{\rm UB}(\tau,\vec{x}_{\wedge i}^{(k)})\right]\\
&+ \sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\frac{\sigma_i\sigma_j\rho_{ij}}{4h_ih_j}\left[-\delta_{k_i,0}Y_i^{\rm LB}(\tau,\vec{x}_{\wedge i}^{(k)}) - \delta_{k_j,0}Y_j^{\rm LB}(\tau,\vec{x}_{\wedge j}^{(k)})\right.\\
&\qquad\left. + \delta_{k_i,n_{\rm gr}-1}Y_i^{\rm UB}(\tau,\vec{x}_{\wedge i}^{(k)}) + \delta_{k_j,n_{\rm gr}-1}Y_j^{\rm UB}(\tau,\vec{x}_{\wedge j}^{(k)})\right]\\
&+ \sum_{i=1}^{d}\frac{1}{2h_i}\left(r - \frac{1}{2}\sigma_i^2\right)\left[\delta_{k_i,n_{\rm gr}-1}Y_i^{\rm UB}(\tau,\vec{x}_{\wedge i}^{(k)}) - \delta_{k_i,0}Y_i^{\rm LB}(\tau,\vec{x}_{\wedge i}^{(k)})\right].
\end{aligned} \tag{12}$$

Then, under the following assumption, we have a lemma on the accuracy of the approximation (8).

Assumption II.1. *$Y(\tau, \vec{x})$, the solution of (5) and (6), is four-times differentiable with respect to $x_1, ..., x_d$ and there exist $\zeta, \xi \in \mathbb{R}$ such that*

$$\forall i,j,k,l \in [d], \tau \in (0,T), \vec{x} \in \tilde{D},$$
$$\left|\frac{\partial^3 Y}{\partial x_i \partial x_j \partial x_k}(\tau, \vec{x})\right| < \zeta, \left|\frac{\partial^4 Y}{\partial x_i \partial x_j \partial x_k \partial x_l}(\tau, \vec{x})\right| < \xi. \quad (13)$$

Lemma II.1. *Let $Y(\tau, \vec{x})$ be the solution of (5) and (6), and $\vec{\tilde{Y}}(\tau)$ be that of (8) and (9). Under Assumption II.1, if, for a given $\epsilon \in \mathbb{R}_+$,*

$$h_i \leq \min\left\{\frac{1}{d\sigma_i}\sqrt{\frac{3\epsilon}{2\xi T}}, \frac{1}{\sigma_i}\sqrt{\frac{3\epsilon}{\zeta d T}}\right\}, i \in [d] \quad (14)$$

then, for any $\tau \in (0,T)$, the inequality

$$\|\vec{\tilde{Y}}(\tau) - \vec{Y}(\tau)\| < \sqrt{N_{\rm gr}}\epsilon \quad (15)$$

holds, where $\vec{Y}(\tau) = (Y(\tau, \vec{x}^{(1)}), ..., Y(\tau, \vec{x}^{(N_{\rm gr})}))^T$.

Lemma II.1 means that the root mean square of the differences between $\tilde{Y}_i(\tau)$ and $Y(\tau, \vec{x}^{(i)})$ is upper bounded by ϵ.

III. QUANTUM ALGORITHM FOR SOLVING ORDINARY DIFFERENTIAL EQUATION SYSTEMS

In this section, we outline the algorithm of [10]. This is the algorithm for solving the linear ODE system

$$\frac{d}{dt}\vec{x}(t) = A\vec{x} + \vec{b}, \quad (16)$$

with the initial condition $\vec{x}(0) = \vec{x}_{\rm ini}$. Here, $\vec{x}(t) \in \mathbb{R}^N$, $A \in \mathbb{R}^{N\times N}$ is a constant diagonalizable matrix, and $\vec{b} \in \mathbb{R}^N$ is a constant vector. Suppose that we want to find $\vec{x}(T)$ for some $T \in \mathbb{R}_+$. The algorithm is based on the formal solution of (16)

$$\vec{x}(T) = e^{AT}\vec{x}_{\rm ini} + (e^{AT} - I_N)A^{-1}\vec{b}. \quad (17)$$

In order to calculate this, we consider the linear equation system on the tensor product space $V := \mathbb{R}^{q+1} \otimes \mathbb{R}^N$, where the former is the auxiliary space and the latter is the original space on which A operates:

$$C_{m,k,p}(Ah_t)\vec{X} = \vec{e}_0 \otimes \vec{x}_{\rm ini} + h\sum_{i=0}^{m-1}\vec{e}_{i(k+1)+1} \otimes \vec{b}. \quad (18)$$

Here, m, p, k are positive integers set large enough (see the statement of Theorem III.1), $q := m(k+1) + p$, $h_t = T/m$, $\vec{X} \in \mathbb{R}^{N(q+1)}$ and $\{\vec{e}_i\}_{i=0,1,...,q}$ is an orthonormal basis of $\mathbb{R}^{q+1}$. For $B \in \mathbb{R}^{N\times N}$, the $N(q+1)\times N(q+1)$ matrix $C_{m,k,p}(B)$ is defined as

$$C_{m,k,p}(B) := \sum_{j=0}^{q}\vec{e}_j\vec{e}_j^T \otimes I_N - \sum_{i=0}^{m-1}\sum_{j=1}^{k}\vec{e}_{i(k+1)+j}\vec{e}_{i(k+1)+j-1}^T \otimes \frac{1}{j}B$$
$$-\sum_{i=0}^{m-1}\sum_{j=0}^{k}\vec{e}_{(i+1)(k+1)}\vec{e}_{i(k+1)+j}^T \otimes I_N - \sum_{j=m(k+1)+1}^{q}\vec{e}_j\vec{e}_{j-1}^T \otimes I. \quad (19)$$

$C_{m,k,p}$ is designed based on the Taylor expansion of (17). The solution of (18) can be written as

$$\vec{X} = \sum_{i=0}^{m-1}\sum_{j=1}^{k}\vec{e}_{i(k+1)+j} \otimes \vec{x}_{i,j} + \sum_{j=0}^{p}\vec{e}_{m(k+1)+j} \otimes \vec{x}_m, \quad (20)$$

for some vectors $\vec{x}_{i,j}, \vec{x}_m \in \mathbb{R}^N$, and $\vec{x}_m$ becomes close to $\vec{x}(T)$, which we want to find. Note that $\vec{x}_m$ is repeated p times in the solution $\vec{X}$, which enhances the probability of obtaining the desired vector in the output quantum state of the algorithm.

Although the $C_{m,k,p}(Ah_t)$ is an extremely large matrix, the quantum algorithms for solving linear equation systems (QLS algorithms)[13–15] can output the solution of (18) only with complexity of $O(\log \mathcal{N})$, where $\mathcal{N}$ is the number of rows (or columns) in $C_{m,k,p}(Ah_t)$. The quantum algorithm in [10] leverages the algorithm in [15]. In order to use it, [10] assumes that the following oracles (i.e. unitary operators) are available:

- $O_{A,1}$: this returns the column index of the l-th nonzero entry in the j-th row of the matrix A
$$O_{A,1} : |j\rangle|l\rangle \mapsto |j\rangle|\nu(j,l)\rangle \quad (21)$$
- $O_{A,2}$: this returns the (j,k) entry of the matrix A
$$O_{A,2} : |j\rangle|k\rangle|z\rangle \mapsto |j\rangle|k\rangle|z \oplus A_{jk}\rangle \quad (22)$$
- $$O_{\vec{x}_{\rm ini}} : \begin{cases}|\bar{0}\rangle|0\rangle \mapsto \frac{1}{\|\vec{x}_{\rm ini}\|}|\bar{0}\rangle|\vec{x}_{\rm ini}\rangle \\ |\bar{1}\rangle|\psi\rangle \mapsto |\bar{1}\rangle|\psi\rangle \text{ for any } |\psi\rangle\end{cases} \quad (23)$$
- $$O_{\vec{b}} : \begin{cases}|\bar{0}\rangle|\psi\rangle \mapsto |\bar{0}\rangle|\psi\rangle \text{ for any } |\psi\rangle \\ |\bar{1}\rangle|0\rangle \mapsto \frac{1}{\|\vec{b}\|}|\bar{1}\rangle|\vec{b}\rangle\end{cases}. \quad (24)$$

  for $\vec{b} \neq 0$. When $\vec{b} = 0$, this is an identity operator.

Then, we present the theorem (Theorem 9 in [10]), which states the query complexity of the algorithm.

Theorem III.1. *(Theorem 9 in [10], slightly modified) Suppose $A = VDV^{-1}$ is an $N \times N$ diagonalizable matrix, where $D = {\rm diag}(\lambda_0, \lambda_1, ..., \lambda_{N-1})$ satisfies ${\rm Re}(\lambda_j) \leq 0$ for any $j \in 0, 1, ..., N-1$. In addition, suppose A has at most s nonzero entries in any row and column, and we have oracles $O_{A,1}, O_{A,2}$ as above. Suppose $\vec{x}_{\rm ini}$ and $\vec{b}$ are N-dimensional vectors with known norms and we have oracles $O_{\vec{x}_{\rm ini}}$ and $O_{\vec{b}}$ as above. Let $\vec{x}$ evolve according to the differential equation (16) with the initial condition $\vec{x}(0) = \vec{x}_{\rm ini}$. Let $T > 0$ and $g := \max_{t\in[0,T]}\|\vec{x}(t)\|/\|\vec{x}(T)\|$. Then there exists a quantum algorithm that produces a state $|\tilde{\Psi}\rangle$, which is ϵ-close to*

$$|\Psi\rangle := \frac{1}{\sqrt{\langle\Psi_{\rm gar}|\Psi_{\rm gar}\rangle + (p+1)\|\vec{x}(T)\|^2}}\left(|\Psi_{\rm gar}\rangle + \sum_{j=p(k+1)}^{p(k+2)}|j\rangle|\vec{x}(T)\rangle\right) \quad (25)$$

making

$$O\left(\kappa_V sT\|A\| \times {\rm poly}\left(\log\left(\frac{\kappa_V sT\|A\|}{\epsilon}\right)\right)\right) \quad (26)$$

queries to $O_{A,1}$, $O_{A,2}$, O_x, and O_b. Here, $\kappa_V = \|V\| \cdot \|V^{-1}\|$ is the condition number of V, $p = \lceil T\|A\|\rceil$, $k = \lfloor 2\log\Omega/\log(\log\Omega)\rfloor$, $\Omega = 70g\kappa_V p^{3/2}(\|\vec{x}_{\rm ini}\| + T\|\vec{b}\|)/\epsilon\|\vec{x}(T)\|$, and $|\Psi_{\rm gar}\rangle$ is an unnormalized state which takes the form of $|\Psi_{\rm gar}\rangle = \sum_{j=0}^{p(k+1)-1}|j\rangle|\psi_j\rangle$ with some unnormalized states $|\psi_0\rangle, |\psi_1\rangle, ..., |\psi_{p(k+1)-1}\rangle$ and satisfies $\langle\Psi_{\rm gar}|\Psi_{\rm gar}\rangle = O(g^2(p+1)\|\vec{x}(T)\|^2)$.

The modifications from Theorem 9 in [10] are as follows. First, in [10], it is assumed that we perform post-selection to obtain $|\vec{x}(T)\rangle\rangle/\||\vec{x}(T)\rangle\rangle\|$ (strictly speaking, a state close to it). On the other hand, in Theorem III.1, the output state is not purely $|\vec{x}(T)\rangle\rangle/\||\vec{x}(T)\rangle\rangle\|$ but contains $|\vec{x}(T)\rangle\rangle$ as a part in addition to the unnecessary state $|\Psi_{\rm gar}\rangle$. This is because, in this paper, we use the algorithm of [10] as a subroutine in the quantum amplitude estimation (QAE)[16, 17], as explained in Section V, and the iterated subroutine in QAE must be an unitary operation. This means that we cannot perform post-selection, since it is a non-unitary operation. Note also that, we do not perform amplitude amplification for $|\Psi_1\rangle$, which is done before post-selection in [10], and thus a factor g, which exists in the expression of the complexity (112) in [10], has dropped from (26) in this paper. Moreover, the meaning of the closeness ϵ is different between Theorem III.1 in this paper and Theorem 9 in [10]. In the former, ϵ is the closeness between $|\tilde{\Psi}\rangle$ and $|\Psi\rangle$, which corresponds to δ in [10]. On the other hand, Theorem 9 in [10] refers to the closeness of the state after post-selection to $|\vec{x}(T)\rangle\rangle/\||\vec{x}(T)\rangle\rangle\|$. This difference also makes (26) different from (112) in [10].

IV. APPROXIMATING THE PRESENT DERIVATIVE PRICE AS THE EXPECTED VALUE OF THE PRICE AT A FUTURE TIME

As we explained in the introduction, we aim to calculate V_0 as the expected value of the discounted price at some future time. Concretely, we set $t_{\rm ter} \in (0, T)$ and calculate

$$V_0 = e^{-rt_{\rm ter}}\int_{\mathbb{R}_+^d} d\vec{S}\phi(t_{\rm ter},\vec{S})p_{\rm NB}(t_{\rm ter},\vec{S})V(t_{\rm ter},\vec{S}), \qquad (27)$$

where $\phi(t,\vec{S})$ is the probability density function of $\vec{S}(t)$, and $p_{\rm NB}(t,\vec{S})$ is the conditional probability that the no event which leads to extinction of the payoff happens by t given $\vec{S}(t) = \vec{S}$. Although (27) holds for any $t_{\rm ter}$, for the effective numerical calculation, $t_{\rm ter}$ should be set carefully. Recalling our motivation to evade exponential complexity to read out V_0, which is explained in Section I, we want to set $t_{\rm ter}$ as large as possible. On the other hand, there are some reasons to set $t_{\rm ter}$ small because of existence of boundaries. First, note that it is difficult to find $p_{\rm NB}(t_{\rm ter},\vec{S})$ explicitly in the multi-asset case. However, for sufficiently small $t_{\rm ter}$, $p_{\rm NB}(t_{\rm ter},\vec{S})$ is nearly equal to 1, since the payoff is paid at least if $\vec{S}(t)$ does not reach any boundaries and the probability that $\vec{S}(t)$ reaches any boundaries can be neglected for time close to 0. Besides, note that we obtain the derivative prices only on the points in boundaries by solving PDE. For small $t_{\rm ter}$, we can approximately calculate V_0 using only the information in boundaries, since the probability distribution of $\vec{S}(t_{\rm ter})$ over the boundaries is negligible. In summary, we should set $t_{\rm ter}$ as large as possible in the range of the value for which the probability distribution of $\vec{S}(t_{\rm ter})$ is almost confined within the boundaries. For such $t_{\rm ter}$, we can approximate

$$V_0 \approx e^{-rt_{\rm ter}}\int_D d\vec{S}\phi(t_{\rm ter},\vec{S})V(t_{\rm ter},\vec{S}) = e^{-rT}\int_{\tilde{D}} d\vec{x}\tilde{\phi}(t_{\rm ter},\vec{x})Y(\tau_{\rm ter},\vec{x}), \qquad (28)$$

where $\tau_{\rm ter} := T - t_{\rm ter}$ and $\tilde{\phi}(t,\vec{x})$ is the probability density of $\vec{x}(t)$ under the BS model (3) and will be explicitly given later.

Considering the above points, we obtain the lemma, which shows a criterion to set $t_{\rm ter}$. First, we make an assumption, which is necessary to upper bound the contribution from the outside of the boundaries to the integral (27).

Assumption IV.1. *There exist positive constants $A_0, A_1, ..., A_d$ such that, for any $\vec{S} \in D$, $f_{\rm pay}$ in Problem 1 satisfies*

$$f_{\rm pay}(\vec{S}) \le \sum_{i=1}^d A_i S_i + A_0. \qquad (29)$$

That is, we assume that the payoff is upper bounded by some linear function, which is the case for many cases such as call/put options on linear combinations of $S_1, ..., S_d$ (i.e. basket options). Then, the following lemma holds.

Lemma IV.1. *Consider Problem 1. Under Assumption IV.1, for any $\epsilon \in \mathbb{R}_+$ satisfying*

$$\log\left(\frac{\tilde{A}d(d+1)}{\epsilon}\right) > \max\left\{\frac{2}{5}\left(1-\frac{2r}{\sigma_i^2}\right)\log\left(\frac{U_i}{S_{i,0}}\right), \frac{2}{5}\left(1-\frac{2r}{\sigma_i^2}\right)\log\left(\frac{S_{i,0}}{L_i}\right)\right\}, i \in [d], \qquad (30)$$

where $\tilde{A} = \max\{A_1\sqrt{U_1 S_{1,0}}, ..., A_d\sqrt{U_d S_{d,0}}, A_0\}$, and

$$\epsilon < 2d(d+1) \times \max\{A_0, A_1 S_{1,0}, ..., A_d S_{d,0}\}, \qquad (31)$$

the inequality

$$\left|V(0,\vec{S}_0) - e^{-rT}\int_{\tilde{\tilde{D}}} d\vec{x}\tilde{\phi}(t_{\rm ter},\vec{x})Y(t_{\rm ter},\vec{x})\right| \le 2\epsilon \qquad (32)$$

holds, where

$$t_{\rm ter} := \min\left\{\frac{2\left(\log\left(\frac{U_1}{S_{1,0}}\right)\right)^2}{25\sigma_1^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}, ..., \frac{2\left(\log\left(\frac{U_d}{S_{d,0}}\right)\right)^2}{25\sigma_d^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}, \frac{2\left(\log\left(\frac{S_{1,0}}{L_1}\right)\right)^2}{25\sigma_1^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}, ..., \frac{2\left(\log\left(\frac{S_{d,0}}{L_d}\right)\right)^2}{25\sigma_d^2\log\left(\frac{2\tilde{A}d(d+1)}{\epsilon}\right)}\right\}. \qquad (33)$$

and

$$\tilde{\tilde{D}} := \left[\frac{1}{2}\left(l_1 + x_1^{(0)}\right), \frac{1}{2}\left(x_1^{(n_{\rm gr}-1)} + u_1\right)\right] \times \cdots \times \left[\frac{1}{2}\left(l_d + x_d^{(0)}\right), \frac{1}{2}\left(x_d^{(n_{\rm gr}-1)} + u_d\right)\right]. \qquad (34)$$

Note that, in (32), the region of the integral is slightly different from $\tilde{D}$, the interior of the boundary in the $\vec{x}$ domain. This is just for interpreting the finite-sum approximation of the integral as the midpoint rule (see the proof of Lemma V.1 in [1] for the detail).

V. QUANTUM METHOD FOR DERIVATIVE PRICING BY FDM

In this section, we finally present the quantum method for derivative pricing by FDM. Our idea is calculating the present derivative price V_0 as (27), the expected value of the price at the future time $t_{\rm ter}$. As explained in Section IV, we approximate (27) as (28). In fact, we have to approximate (28) further, since we obtain the derivative prices only on the grid points by solving PDE using FDM. Therefore, we approximate (28) as

$$V_0 \approx e^{-rT} \sum_{k=1}^{N_{\rm gr}} p_k \tilde{Y}_k(\tau_{\rm ter}) = e^{-rT} \vec{p} \cdot \vec{\tilde{Y}}(\tau_{\rm ter}), \qquad (35)$$

where $\vec{p} := (p_1, ..., p_{N_{\rm gr}})^T$, and p_k is the existence probability of $\vec{x}(t_{\rm ter})$ on the k-th grid point and explicitly defined soon. Hereafter, we discuss how to estimate this inner product.

A. Generating the probability vector

Firstly, let us discuss how to generate $\vec{p}$, a vector which represents $\tilde{\phi}(t_{\rm ter}, \vec{x})$, the probability distribution of $\vec{x}(t_{\rm ter})$, as a quantum state. As we will see below, although we aim to generate a quantum state in which the amplitudes of basis states are proportional to $\tilde{\phi}(t_{\rm ter}, \vec{x})$, we can apply the method to generate a state in which amplitudes are square roots of probabilities[4, 18], since $\tilde{\phi}(t_{\rm ter}, \vec{x})$ can be regarded as the square roots of the probability densities under another distribution.

Concretely speaking, we aim to generate the vector

$$\vec{p} := (p_1, ..., p_{N_{\rm gr}})^T, p_k := \tilde{\phi}(t_{\rm ter}, \vec{x}) \prod_{i=1}^{d} h_i \qquad (36)$$

where $\tilde{\phi}(t, \vec{x})$, the probability density of $\vec{x}(t)$, is given as

$$\tilde{\phi}(t, \vec{x}) := \frac{1}{(2\pi t)^{d/2} \left(\prod_{i=1}^{d} \sigma_i\right) \sqrt{\det \rho}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})\right) \qquad (37)$$

with $\vec{\mu} := \left(\left(r - \frac{1}{2}\sigma_1^2\right)t, ..., \left(r - \frac{1}{2}\sigma_d^2\right)t\right)^T$ and $\Sigma := (\sigma_i \sigma_j \rho_{ij})_{1 \le i,j \le d}$, that is, the density of the d-dimensional normal distribution with the mean $\vec{\mu}$ and the covariance matrix Σ. Actually, we generate this vector as a normalized quantum state, that is,

$$|\bar{p}\rangle := \sum_{k=1}^{N_{\rm gr}} \frac{p_k}{P} |k\rangle, P := \|\vec{p}\| = \sqrt{\sum_{k=1}^{N_{\rm gr}} p_k^2}. \qquad (38)$$

Here, note that $(\tilde{\phi}(t, \vec{x}))^2$ is $\varphi(\vec{x})$ times a constant independent of $\vec{x}$, where

$$\varphi(\vec{x}) := \frac{1}{(\pi t)^{d/2} \left(\prod_{i=1}^{d} \sigma_i\right) \sqrt{\det \rho}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T \left(\frac{1}{2}\Sigma\right)^{-1} (\vec{x} - \vec{\mu})\right), \qquad (39)$$

is the probability density for another d-dimensional normal distribution. Therefore, $|\bar{p}\rangle$ is approximately the state

$$|\varphi\rangle := \frac{1}{\sqrt{Q}} \sum_{k=1}^{N_{\rm gr}} \sqrt{q_k} |k\rangle, \qquad (40)$$

where $\varphi(\vec{x})$ is encoded into the square roots of the amplitudes. Here, for $k = \sum_{i=1}^{d} n_{\rm gr}^{d-i} k_i + 1$ with $k_1, ..., k_d \in [n_{\rm gr} - 1]_0$,

$$q_k := \int_{x_1^{(k_1)}}^{x_1^{(k_1+1)}} dx_1 \cdots \int_{x_d^{(k_d)}}^{x_d^{(k_d+1)}} dx_d \varphi(t_{\rm ter}, \vec{x}), \qquad (41)$$

which is close to $\varphi(t_{\rm ter}, \vec{x}^{(k)}) \prod_{i=1}^{d} h_i$, and

$$Q := \int_{x_1^{(0)}}^{x_1^{(n_{\rm gr})}} dx_1 \cdots \int_{x_d^{(0)}}^{x_d^{(n_{\rm gr})}} dx_d \varphi(t_{\rm ter}, \vec{x}), \qquad (42)$$

which is close to 1.

Then, the task is boiled down to generating $|\varphi\rangle$. This is done by Algorithm 1, the multivariate extension of the method of [18].

Algorithm 1 Generate $|\bar{p}\rangle$

```
1: Prepare $d$ $m_{\rm gr}$-qubit registers and initialize all qubits to $|\bar{0}\rangle$, which
   means the initial state is $|0\rangle^{\otimes d}$.
2: for $i = 1$ to $d$ do
3:     for $j = 1$ to $m_{\rm gr}$ do
4:         Using $k_1, ..., k_{i-1}$ indicated by the first, ..., $(i-1)$-th registers,
           respectively, and $k_i^{[1]}, ..., k_i^{[j-1]}$, the bits on the first, ..., $(j-1)$-th
           qubits of the $i$-th register, respectively, rotate the $j$-th
           qubit in the $i$-th register as
           $$|\bar{0}\rangle \to \sqrt{f_{i,j}(k_1, ..., k_{i-1}; k_i^{[1]}, ..., k_i^{[j-1]})} |\bar{0}\rangle + \sqrt{1 - f_{i,j}(k_1, ..., k_{i-1}; k_i^{[1]}, ..., k_i^{[j-1]})} |\bar{1}\rangle.$$
           This transforms the entire state into
           $$\frac{1}{\sqrt{Q}} \sum_{\substack{k_1,...,k_{i-1} \\ \in [n_{\rm gr}-1]_0}} \sum_{\substack{k_i^{[1]},...,k_i^{[j]} \\ \in \{0,1\}}} \sqrt{q_{i,j}(k_1, ..., k_{i-1}; k_i^{[1]}, ..., k_i^{[j]})} \times |k_1\rangle \cdots |k_{i-1}\rangle |\tilde{k}\rangle |0\rangle^{\otimes(d-i)},$$
           where $\tilde{k}$ is an integer whose $m_{\rm gr}$-bit representation is
           $k_i^{[1]} \cdots k_i^{[j]} \underbrace{0 \cdots 0}_{m_{\rm gr}-j}$.
5:     end for
6: end for
```

Here, note that $|k\rangle$ can be decomposed as

$$|k\rangle = |k_1\rangle \cdots |k_d\rangle, \qquad (43)$$

where each $|k_i\rangle$ is a state on a m_{gr}-qubit register (recall that $n_{\mathrm{gr}} = 2^{m_{\mathrm{gr}}}$), and $|k_i\rangle$ can be further decomposed as

$$|k_i\rangle = \left|\overline{k_i^{[1]}}\right\rangle \cdots \left|\overline{k_i^{[m_{\mathrm{gr}}]}}\right\rangle, \tag{44}$$

where we write the n-bit representation of $i \in \{0, 1, ..., 2^n - 1\}$ as $i^{[1]} \cdots i^{[n]}$ with $i^{[1]}, ..., i^{[n]} \in \{0, 1\}$. Besides, note that Algorithm 1 requires us to compute

$$f_{i,j}(k_1, ..., k_{i-1}; k_i^{[1]}, ..., k_i^{[j-1]}) := \frac{q_{i,j}(k_1, ..., k_{i-1}; k_i^{[1]}, ..., k_i^{[j-1]}, 0)}{q_{i,j-1}(k_1, ..., k_{i-1}; k_i^{[1]}, ..., k_i^{[j-1]})} \tag{45}$$

for $i \in [d]$ and $j \in [m_{\mathrm{gr}}]$, where

$$q_{i,j}(k_1, ..., k_{i-1}; b_1, ..., b_j) := \begin{cases} \int_{x^L_{1,j}(b_1,...,b_j)}^{x^R_{1,j}(b_1,...,b_j)} dx_1 \int_{x_2^{(0)}}^{x_2^{(n_{\mathrm{gr}})}} dx_2 \cdots \int_{x_d^{(0)}}^{x_d^{(n_{\mathrm{gr}})}} dx_d \varphi(t, \vec{x}) & ;\ i = 1 \\ \int_{x_1^{(k_1)}}^{x_1^{(k_1+1)}} dx_1 \cdots \int_{x_{i-1}^{(k_{i-1})}}^{x_{i-1}^{(k_{i-1}+1)}} dx_{i-1} \int_{x^L_{i,j}(b_1,...,b_j)}^{x^R_{i,j}(b_1,...,b_j)} dx_i \int_{x_{i+1}^{(0)}}^{x_{i+1}^{(n_{\mathrm{gr}})}} dx_{i+1} \cdots \int_{x_d^{(0)}}^{x_d^{(n_{\mathrm{gr}})}} dx_d \varphi(t, \vec{x}) & ;\ 2 \le i \le d-1 \\ \int_{x_1^{(k_1)}}^{x_1^{(k_1+1)}} dx_1 \cdots \int_{x_{d-1}^{(k_{d-1})}}^{x_{d-1}^{(k_{d-1}+1)}} dx_{d-1} \int_{x^L_{d,j}(b_1,...,b_j)}^{x^R_{d,j}(b_1,...,b_j)} dx_d \varphi(t, \vec{x}) & ; i = d \end{cases}, \tag{46}$$

and

$$x^L_{i,j}(b_1, ..., b_j) := x_i^{(k_L)}, k_L := \begin{cases} 0 & ;\ j = 0 \\ b_1 \cdots b_j \underbrace{0 \cdots 0}_{m_{\mathrm{gr}}-j} & ;\ j \in [m_{\mathrm{gr}}] \end{cases}$$

$$x^R_{i,j}(b_1, ..., b_j) := x_i^{(k_R)}, k_R := \begin{cases} n_{\mathrm{gr}} & ;\ j = 0 \\ b_1 \cdots b_j \underbrace{1 \cdots 1}_{m_{\mathrm{gr}}-j} + 1 & ;\ j \in [m_{\mathrm{gr}}] \end{cases} \tag{47}$$

for $i \in [d]$, $j = 0, 1, ..., m_{\mathrm{gr}}$ and $b_1, ..., b_j \in \{0, 1\}$ (note that $q_{1,0} = Q$). Such a $f_{i,j}$ can be actually computed as follows. Neglecting the contribution from the outside of the boundary, we see that

$$f_{i,j}(k_1, ..., k_{i-1}; k_i^{[1]}, ..., k_i^{[j-1]}) \approx \frac{\int_{x^L_{i,j}(b_1,...,b_j)}^{\frac{1}{2}(x^L_{i,j}(b_1,...,b_j)+x^R_{1,j}(b_1,...,b_j))} dx_i \varphi_i^{\mathrm{mar}}(x_i; k_1, ..., k_{i-1})}{\int_{x^L_{i,j}(b_1,...,b_j)}^{x^R_{i,j}(b_1,...,b_j)} dx_i \varphi_i^{\mathrm{mar}}(x_i; k_1, ..., k_{i-1})}, \tag{48}$$

where

$$\varphi_i^{\mathrm{mar}}(x_i; k_1, ..., k_{i-1}) := \int_{-\infty}^{+\infty} dx_{i+1} \cdots \int_{-\infty}^{+\infty} dx_d \varphi((x_1^{(k_1)}, ..., x_{i-1}^{(k_{i-1})}, x_i, x_{i+1}, ..., x_d)^T) \tag{49}$$

is the marginal density given by integrating out $x_{i+1}, ..., x_d$ and fixing $x_1, ..., x_{i-1}$. We can regard this as an univariate normal distribution density function of x_i (times a constant independent of x_i), and therefore compute (48) by the method presented in [4].

At the end of this subsection, let us evaluate the error of (35) as an approximation for (28). As preparation, we evaluate the normalization factor P as follows:

$$P^2 = \sum_{k=1}^{N_{\mathrm{gr}}} \left(\phi_{\vec{x}}(t, \vec{x}_{\mathrm{gr}}^{(k)})\right)^2 \left(\prod_{i=1}^{d} h_i\right)^2 \approx \frac{\prod_{i=1}^d \Delta_i}{(4\pi)^{d/2} N_{\mathrm{gr}} \sqrt{\det\rho}}, \tag{50}$$

where

$$\Delta_i := \frac{u_i - l_i}{\sigma_i \sqrt{t_{\mathrm{ter}}}}, i \in [d]. \tag{51}$$

Besides, we make an additional assumption.

Assumption V.1. *For $Y(\tau, \vec{x})$, the solution of (5) and (6), and $\tilde{\phi}(t, \vec{x})$, the probability density function of $\vec{x}(t)$ under the BS model (3), there exists $\eta \in \mathbb{R}$ such that*

$$\forall i, j \in [d], \tau \in (0, T), \vec{x} \in \tilde{D}, \left|\frac{\partial^2}{\partial x_i \partial x_j}(\tilde{\phi}(T - \tau, \vec{x}) Y(\tau, \vec{x}))\right| < \eta. \tag{52}$$

Then, we obtain the following lemma, which guarantees us that we can approximate the integral by the finite sum over the grid points.

Lemma V.1. *Consider Problem 1. Under Assumptions II.1, IV.1 and V.1, for a given $\epsilon \in \mathbb{R}_+$ satisfying (30) and (31), if we set*

$$h_i < \tilde{h}_i := \min\left\{\frac{(4\pi)^{d/8}(\det\rho)^{1/8}}{d\sigma_i(\prod_{i=1}^d \Delta_i)^{1/4}} \sqrt{\frac{\epsilon}{2\xi T}}, \frac{(4\pi)^{d/8}(\det\rho)^{1/8}}{\sigma_i(\prod_{i=1}^d \Delta_i)^{1/4}} \sqrt{\frac{\epsilon}{\zeta d T}}, \frac{1}{\left(\prod_{i=1}^d (u_i - l_i)\right)^{1/2}} \sqrt{\frac{24\epsilon}{d\eta}}\right\} \tag{53}$$

for each $i \in [d]$, the following holds

$$\left|e^{-rT} \vec{p} \cdot \tilde{\vec{Y}}(\tau_{\mathrm{ter}}) - V_0\right| < 4\epsilon, \tag{54}$$

where $\vec{p}$ is defined as (36), $\tilde{\vec{Y}}$ is the solution of (8).

B. Generating the derivative price vector

Next, let us consider how to generate $\vec{V}(t_{\mathrm{ter}})$, the vector which encodes the grid derivative prices at t_{ter}. Precisely speaking, since we solve (8), we actually obtain the vector $\vec{\tilde{Y}}(\tau_{\rm ter})$, which encodes the approximations of $Y(\tau_{\rm ter},\vec{x})$ on the grid points. Furthermore, by the algorithm presented in Section III, we obtain not $\vec{\tilde{Y}}$ itself but some quantum state like (25), which contains a state corresponding to $\vec{\tilde{Y}}$ along with a garbage state.

For the precise discussion, let us firstly make some assumptions in order to satisfy preconditions to use the quantum algorithm. The first one is as follows:

Assumption V.2. *$\vec{C}(\tau)$ in (8) is independent of τ.*

Then, hereafter, we simply write $\vec{C}(\tau)$ as $\vec{C}$. We make this assumption in order to fit the current setting to [10], which considered solving (16) for constant A and $\vec{b}$ (note that F in (8) is constant). Although $\vec{C}(\tau)$ is not generally time-independent, the assumption is satisfied in some cases. For example, if a boundary corresponds to a knock-out barrier, $V(t,\vec{S})=0$ on it. Of course, there are many cases where $\vec{C}(\tau)$ is time-dependent, and it is desirable to expend our method to such cases. We leave this as a future work.

The second assumption is as follows:

Assumption V.3. *For F in (10), the following oracles $O_{F,1}$ and $O_{F,2}$ are available:*

$$O_{F,1}:|j\rangle|l\rangle\mapsto|j\rangle|\nu_F(j,l)\rangle, \tag{55}$$

where $j\in[N_{\rm gr}]$, $l\in[s_F]$, s_F is the sparsity of F, and $\nu(j,l)$ is the column index of the l-th nonzero entry in the j-th row,

$$O_{F,2}:|j\rangle|k\rangle|z\rangle\mapsto|j\rangle|k\rangle|z\oplus F_{jk}\rangle, \tag{56}$$

where $j,k\in[N_{\rm gr}]$ and $z\in\mathbb{R}$. Besides, for $\vec{\tilde{f}}_{\rm pay}$ in (9) and $\vec{C}$ in (12), we know their norms and the following oracles $O_{\vec{\tilde{f}}_{\rm pay}}$ and $O_{\vec{C}}$ are available:

$$O_{\vec{\tilde{f}}_{\rm pay}}:\begin{cases}|\bar{0}\rangle|0\rangle\mapsto\frac{1}{\|\vec{\tilde{f}}_{\rm pay}\|}|\bar{0}\rangle|\vec{\tilde{f}}_{\rm pay}\rangle\\ |\bar{1}\rangle|\psi\rangle\mapsto|\bar{1}\rangle|\psi\rangle\ \text{for any}\ |\psi\rangle\end{cases}, \tag{57}$$

$$O_{\vec{C}}:\begin{cases}|\bar{0}\rangle|\psi\rangle\mapsto|\bar{0}\rangle|\psi\rangle\ \text{for any}\ |\psi\rangle\\ |\bar{1}\rangle|0\rangle\mapsto\frac{1}{\|\vec{C}\|}|\bar{1}\rangle|\vec{C}\rangle\end{cases}, \tag{58}$$

for $\vec{C}\neq0$ and $O_{\vec{C}}$ is an identity operator for $\vec{C}=0$.

Since F is explicitly given as (10), the sum of the Kronecker products of tridiagonal matrices, construction of $O_{F,1}$ and $O_{F,2}$ is straightforward. On the other hand, $\vec{\tilde{f}}_{\rm pay}$ and $\vec{C}$ are highly problem-dependent, and so are $O_{\vec{\tilde{f}}_{\rm pay}}$ and $O_{\vec{C}}$. Therefore, we just assume their availability in this paper, referring to some specific cases. Although the gate complexity for preparing a state in which a general vector is amplitude-encoded is exponential in the qubit number [19], it can be efficiently performed in the following cases.

- By the analogy with preparation of $|\vec{p}\rangle$, we see that we can prepare $\frac{|\vec{\tilde{f}}_{\rm pay}\rangle}{\|\vec{\tilde{f}}_{\rm pay}\|}$ if we can analytically calculate the integral of the square of $\tilde{f}_{\rm pay}(\vec{x})$. For example, it is possible if $\tilde{f}_{\rm pay}(\vec{x})$ depends on only one of $x_1,...,x_d$ (say x_1) and has a simple form such as call-option-like $\tilde{f}_{\rm pay}(\vec{x})=\max\{e^{x_1}-K,0\}$.
- If all boundaries correspond to knock-out barriers, $\vec{C}=\vec{0}$, and therefore $O_{\vec{C}}$ is just an identity operator.

Then, we obtain the following lemma.

Lemma V.2. *Consider the ODE system (8). Assume that Assumptions II.1, IV.1, V.1, V.2 and V.3 are satisfied. Let ϵ be any positive real number satisfying (30) and (31), and ϵ' be any positive real number. Then, there exists a quantum algorithm that produces a state $|\tilde{\Psi}\rangle$ ϵ'-close to*

$$|\Psi\rangle:=\frac{1}{\sqrt{\langle\Psi_{\rm gar}|\Psi_{\rm gar}\rangle+(p+1)\|\vec{\tilde{Y}}(\tau_{\rm ter})\|^2}}\left(|\Psi_{\rm gar}\rangle+\sum_{j=p(k+1)}^{p(k+2)}|j\rangle|\vec{\tilde{Y}}(\tau_{\rm ter})\rangle\right), \tag{59}$$

where $\vec{\tilde{Y}}(\tau_{\rm ter})$ is a vector satisfying (54), making

$$O\left(C\times{\rm poly}\left(\log\left(\frac{C}{\epsilon'}\right)\right)\right) \tag{60}$$

queries to $O_{F,1}$, $O_{F,2}$, $O_{\vec{\tilde{f}}_{\rm pay}}$, and $O_{\vec{C}}$. Here,

$$C:=\max\left\{\frac{\sqrt{\prod_{i=1}^d\Delta_i}d^2\Xi\sigma^2_{\max}\tau_{\rm ter}}{(4\pi)^{d/4}(\det\rho)^{1/4}},d\eta\prod_{i=1}^d(u_i-l_i)\right\}\times\frac{\kappa_Vd^4\sigma^2_{\max}\tau_{\rm ter}}{\epsilon}, \tag{61}$$

$\kappa_V=\|V\|\cdot\|V^{-1}\|$ is the condition number of V which diagonalizes F (i.e. VFV^{-1} is a diagonal matrix), $\sigma_{\max}:=\max_{i\in[d]}\sigma_i$, $\Xi:=\max\{\xi,\zeta/d\}$, $\tau_{\rm ter}:=T-t_{\rm ter}$, $t_{\rm ter}$ is defined as (33), $p:=\lceil\tau_{\rm ter}\|F\|\rceil$, $k:=\lfloor2\log\Omega/\log(\log\Omega)\rfloor$, $\Omega=70g\kappa_Vp^{3/2}(\|\vec{\tilde{f}}_{\rm pay}\|+T\|\vec{C}\|)/\epsilon\|\vec{\tilde{Y}}(\tau_{\rm ter})\|$, and $|\Psi_{\rm gar}\rangle$ is an unnormalized state which takes the form of $|\Psi_{\rm gar}\rangle=\sum_{j=0}^{p(k+1)-1}|j\rangle|\psi_j\rangle$ with some unnormalized states $|\psi_0\rangle,|\psi_1\rangle,...,|\psi_{p(k+1)-1}\rangle$ and satisfies

$$\langle\Psi_{\rm gar}|\Psi_{\rm gar}\rangle=O(g^2(p+1)\|\vec{\tilde{Y}}(\tau_{\rm ter})\|^2) \tag{62}$$

with $g:=\max_{\tau\in[0,\tau_{\rm ter}]}\|\vec{\tilde{Y}}(\tau)\|/\|\vec{\tilde{Y}}(\tau_{\rm ter})\|$.

C. Proposed algorithm

Finally, based on the above discussions, we present the quantum method to calculate the present derivative price V_0. Our strategy is calculating this as (35). More concretely, we aim to subtract the information of $\vec{p}\cdot\vec{\tilde{Y}}(\tau_{\rm ter})$ from $|\Psi\rangle$ in (59), the output state of the algorithm of [10].

In order to do this, we first modify the algorithm slightly. That is, we aim to solve not (18) but the following one by the QLS algorithm:

$$\tilde{C}_{m,k,p}(Fh_t)\vec{X} = \vec{e}_0 \otimes \vec{\tilde{f}}_{\rm pay} + h_t \sum_{i=0}^{m-1} \vec{e}_{i(k+1)+1} \otimes \vec{C} + \sum_{i=1}^{p+1} \vec{e}_{m(k+1)+p+i} \otimes \vec{\gamma}. \tag{63}$$

Here, m, p, k are integers defined in the statement of Lemma V.2, $q := m(k+1)+2p+1$, $h_t = \tau_{\rm ter}/m$, $\vec{X} \in \mathbb{R}^{N_{\rm gr}(q+1)}$, $\{\vec{e}_i\}_{i=0,1,...,q}$ is an orthonormal basis of $\mathbb{R}^{q+1}$, and $\vec{\gamma} := (\gamma, ..., \gamma)^T \in \mathbb{R}^{N_{\rm gr}}$ for some $\gamma \in \mathbb{R}_+$. Hereafter, we make the following assumption on γ:

Assumption V.4. *We are given $\gamma \in \mathbb{R}_+$ satisfying*

$$\frac{1}{2}\bar{Y}(\tau_{\rm ter}) < \gamma < 2\bar{Y}(\tau_{\rm ter}), \tag{64}$$

with $\bar{Y}(\tau_{\rm ter}) := \sqrt{\frac{1}{N_{\rm gr}}\sum_{k=1}^{N_{\rm gr}}(Y(\tau_{\rm ter}, \vec{x}^{(k)}))^2}$.

This means that γ is comparable with the root mean square of $Y(\tau_{\rm ter}, \vec{x})$ on the grid points. Besides, the $N_{\rm gr}(q+1) \times N_{\rm gr}(q+1)$ matrix $\tilde{C}_{m,k,p}(Fh_t)$ is now defined as

$$\begin{aligned}&\tilde{C}_{m,k,p}(Fh_t) := \\ &\sum_{j=0}^{q} \vec{e}_j \vec{e}_j^T \otimes I_{N_{\rm gr}} - \sum_{i=0}^{m-1}\sum_{j=1}^{k} \vec{e}_{i(k+1)+j}\vec{e}_{i(k+1)+j-1}^T \otimes \frac{1}{j}Fh_t \\ &- \sum_{i=0}^{m-1}\sum_{j=0}^{k} \vec{e}_{(i+1)(k+1)}\vec{e}_{i(k+1)+j}^T \otimes I_{N_{\rm gr}} - \sum_{j=m(k+1)+1}^{m(k+1)+p} \vec{e}_j \vec{e}_{j-1}^T \otimes I_{N_{\rm gr}}.\end{aligned} \tag{65}$$

The solution of (63) is

$$\vec{X} = \sum_{i=0}^{m-1}\sum_{j=1}^{k} \vec{e}_{i(k+1)+j} \otimes \vec{\tilde{\tilde{Y}}}_{i,j} + \sum_{j=0}^{p} \vec{e}_{m(k+1)+j} \otimes \vec{\tilde{\tilde{Y}}}(\tau_{\rm ter}) + \sum_{j=1}^{p+1} \vec{e}_{m(k+1)+p+j} \otimes \vec{\gamma}, \tag{66}$$

for some vectors $\vec{\tilde{\tilde{Y}}}_{i,j}, \vec{\tilde{\tilde{Y}}}(\tau_{\rm ter}) \in \mathbb{R}^{N_{\rm gr}}$, and $\vec{\tilde{\tilde{Y}}}(\tau_{\rm ter})$ becomes close to $\vec{\tilde{Y}}(\tau_{\rm ter})$. Note that, in $\vec{X}$, $\vec{\tilde{\tilde{Y}}}(\tau_{\rm ter})$ and $\vec{\gamma}$ are repeated $(p+1)$-times. Then, applying the quantum algorithm, we can generate the quantum state $|\tilde{\Psi}_{\rm mod}\rangle$ ϵ-close to

$$|\Psi_{\rm mod}\rangle := \frac{1}{Z}\left(|\Psi_{\rm gar}\rangle + \sum_{j=p(k+1)}^{p(k+2)} |j\rangle |\vec{\tilde{Y}}(\tau_{\rm ter})\rangle + \sum_{j=p(k+2)+1}^{p(k+3)+1} |j\rangle |\vec{\gamma}\rangle\right),$$

$$Z := \sqrt{\langle\Psi_{\rm gar}|\Psi_{\rm gar}\rangle + (p+1)\|\vec{\tilde{Y}}(\tau_{\rm ter})\|^2 + (p+1)N_{\rm gr}\gamma^2}. \tag{67}$$

Note that the query complexity for generating $|\tilde{\Psi}_{\rm mod}\rangle$ is (60), similarly to $|\tilde{\Psi}\rangle$. This is because the complexity of the QLS algorithm depends only on the condition number and sparsity of the matrix and the tolerance[15], and the condition number and sparsity of $\tilde{C}_{m,k,p}(Fh_t)$ is same as $C_{m,k,p}(Fh_t)$.

Using $|\tilde{\Psi}_{\rm mod}\rangle$, we can estimate $\vec{p} \cdot \vec{\tilde{Y}}(\tau_{\rm ter})$. The outline is as follows. First, we estimate the inner product

$$\langle\Pi|\Psi_{\rm mod}\rangle = \frac{\sqrt{p+1}}{PZ}\vec{p} \cdot \vec{\tilde{Y}}(\tau_{\rm ter}), \tag{68}$$

where

$$|\Pi\rangle := \frac{1}{\sqrt{p+1}}\sum_{j=p(k+1)}^{p(k+2)} |j\rangle |\bar{p}\rangle, \tag{69}$$

by estimating the amplitude of $|0\rangle|0\rangle$ in $U_\Pi^\dagger U_{\tilde{\Psi}_{\rm mod},\epsilon}|0\rangle|0\rangle$ using QAE. Here, $U_{\tilde{\Psi}_{\rm mod}}$ and U_Π are the unitary operators such that

$$U_{\tilde{\Psi}_{\rm mod}}|0\rangle|0\rangle = |\tilde{\Psi}_{\rm mod}\rangle, \tag{70}$$

and

$$U_\Pi|0\rangle|0\rangle = |\Pi\rangle, \tag{71}$$

respectively. Note that, if we can generate $|\bar{p}\rangle$, we can also generate $|\Pi\rangle$, since this is just a tensor product of $\frac{1}{\sqrt{p+1}}\sum_{j=p(k+1)}^{p(k+2)}|j\rangle$ and $|\bar{p}\rangle$. Next, by QAE, we estimate the probability that we obtain $j \in \{p(k+2)+1, ..., p(k+3)+1\}$ in the first register when we measure $|\tilde{\Psi}_{\rm mod}\rangle$, and then obtain an estimation of $\gamma\sqrt{(p+1)N_{\rm gr}}/Z$. Finally, using E_1 and E_2, the outputs of the first and second estimations, respectively, we calculate

$$\frac{e^{-rT}\gamma\sqrt{N_{\rm gr}}PE_1}{E_2} \tag{72}$$

as an estimation of $e^{-rT}\vec{p} \cdot \vec{\tilde{Y}}(\tau_{\rm ter})$. We present the detailed procedure is described as Algorithm 2.

Algorithm 2 Calculate $e^{-rT}\vec{p} \cdot \vec{\tilde{Y}}(\tau_{\rm ter})$

```
Require:
    $\gamma \in \mathbb{R}_+$ satisfying (64).
    $\epsilon \in \mathbb{R}_+$ satisfying (30) and (31).
    $\epsilon_1, \epsilon_2 \in \mathbb{R}_+$ satisfying (73). $\epsilon_{\tilde{\Psi}_{\rm mod}} \in \mathbb{R}_+$ satisfying (75).
    Accesses to the oracle $U_{\tilde{\Psi}_{\rm mod}}$ such that (70) and (74) and its inverse.
    Accesses to the oracle $U_\Pi$ such that (71) and its inverse.
1: Estimate the amplitude of $|0\rangle|0\rangle$ in the state $U_\Pi^\dagger U_{\tilde{\Psi}_{\rm mod}}|0\rangle|0\rangle$ by QAE with tolerance $\epsilon_1$. Let the output be $E_1$.
2: Estimate the square root of the probability that we obtain either of $p(k+2)+1, ..., p(k+3)+1$ when we measure the first register of $|\tilde{\Psi}_{\rm mod}\rangle$ by QAE with tolerance $\epsilon_2$. Let the output be $E_2$.
3: Output $e^{-rT}\frac{\gamma\sqrt{N_{\rm gr}}PE_1}{E_2} =: \omega$, where $P$ is given by (50).
```

Here, taking some $\epsilon \in \mathbb{R}_+$, we require the tolerances ϵ_1 and ϵ_2 in calculating E_1 and E_2 be

$$\epsilon_1 = O\left(\frac{(2\pi)^{d/2}\sqrt{\det\rho}\epsilon}{g\left(\prod_{i=1}^d \Delta_i\right)\bar{V}}\right), \epsilon_2 = O\left(\frac{\epsilon}{gV_0}\right) \tag{73}$$

respectively, where $\bar{V}(t_{\rm ter}) := \sqrt{\frac{1}{N_{\rm gr}}\sum_{k=1}^{N_{\rm gr}}(V(t_{\rm ter}, \vec{S}^{(k)}))^2}$ is the root mean square of the derivative prices on the grid points at time $t_{\rm ter}$, and $\vec{S}^{(k)} := (S_1^{(k)}, ..., S_d^{(k)})^T := (\exp(x_1^{(k_1)}), ..., \exp(x_d^{(k_d)}))^T$ for $k = \sum_{i=1}^d n_{\rm gr}^{d-i}k_i + 1$ with $k_1, ..., k_d \in [n_{\rm gr}-1]_0$. Besides, we require that

$$\| |\tilde{\Psi}_{\rm mod}\rangle - |\Psi_{\rm mod}\rangle \| < \epsilon_\Psi, \tag{74}$$

where

$$\epsilon_\Psi = O\left(\max\{\epsilon_1, \epsilon_2\}\right). \tag{75}$$

These requirements guarantee the overall error to be smaller than ϵ. We formally state these points along with the complexity of the procedure in Theorem V.1.

Theorem V.1. *Consider Problem 1. Assume that Assumptions II.1, IV.1, V.1, V.2, V.3 and V.4 are satisfied. Then, for any $\epsilon \in \mathbb{R}_+$ satisfying (30) and (31), Algorithm 2 outputs the real number ω such that $|\omega - V_0| = O(\epsilon)$ with a probability higher than a specified value (say, 0.99). In this procedure, $O\left(\mathcal{D} \times \mathrm{poly}\left(\log \mathcal{D}\right)\right)$ queries to $O_{F,1}$, $O_{F,2}$, $O_{\vec{f}_{\mathrm{pay}}}$, and $O_{\vec{C}}$ are made, where*

$$\mathcal{D} := \max\left\{\frac{\sqrt{\prod_{i=1}^d \Delta_i} d^2 \Xi \sigma_{\max}^2 \tau_{\mathrm{ter}}}{(4\pi)^{d/4}(\det\rho)^{1/4}}, d\eta \prod_{i=1}^d (u_i - l_i)\right\}$$
$$\times \max\left\{\frac{\left(\prod_{i=1}^d \Delta_i\right)\bar{V}}{(2\pi)^{d/2}\sqrt{\det\rho}}, V_0\right\} \times \frac{g\kappa_V d^4 \sigma_{\max}^2 \tau_{\mathrm{ter}}}{\epsilon^2}, \tag{76}$$

$\sigma_{\max} := \max_{i\in[d]} \sigma_i$, $\Xi := \max\{\xi, \zeta/d\}$, $\tau_{\mathrm{ter}} := T - t_{\mathrm{ter}}$, t_{ter} is defined as (33), Δ_i is defined as (51), $g := \max_{\tau\in[0,\tau_{\mathrm{ter}}]} \|\vec{\tilde{Y}}(\tau)\|/\|\vec{\tilde{Y}}(\tau_{\mathrm{ter}})\|$, $\bar{V}(t_{\mathrm{ter}}) := \sqrt{\frac{1}{N_{\mathrm{gr}}}\sum_{k=1}^{N_{\mathrm{gr}}}(V(t_{\mathrm{ter}}, \vec{S}^{(k)}))^2}$, and $\kappa_V = \|V\| \cdot \|V^{-1}\|$ is the condition number of V, which diagonalizes F.

Note that the upper bound of the complexity does not have any factor like $(1/\epsilon)^{\mathrm{poly}(d)}$, which means the tremendous speedup with respect to ϵ and d compared with the classical FDM. On the other hand, the exponential dependence on d has not completely disappeared. In fact, there are factors in the form of the d-times product of some numbers such as $\prod_{i=1}^d (u_i - l_i)$ and $\prod_{i=1}^d \Delta_i$. Recall that $u_i - l_i = \log(U_i/L_i)$ is the width between boundaries in the direction of x_i, the logarithm of the i-th underlying asset price, and Δ_i is that divided by $\sigma_i\sqrt{t_{\mathrm{ter}}}$, which roughly measures the extent of the probability distribution of x_i at time t_{ter}. Therefore, these factors are just logarithmic factors to the power of d. Also note that there is a factor of $O(d^6)$, which is polynomial but rather strongly dependent on d.

VI. SUMMARY

In this paper, we studied how to apply the quantum algorithm of [10] for solving linear differential equations to pricing multi-asset derivatives by FDM. FDM is an appropriate method for pricing some types of derivatives such as barrier options, but suffers from the so-called curse of dimensionality, which makes FDM infeasible for large d, the number of underlying assets, since the dimension of the corresponding ODE system grows as $(1/\epsilon)^{\mathrm{poly}(d)}$ for the tolerance ϵ, and so does the complexity. We saw that the quantum algorithm for solving ODE systems, which provides the exponential speedup with respect to the dimensionality compared with classical methods, is beneficial also for derivative pricing. In order to address the specific issue for derivative pricing, that is, extracting the present price from the output state of the quantum algorithm, we adopted the strategy that we calculate the present price as the expected value of the price at some appropriate future time t_{ter}. Then, we constructed the concrete calculation procedure, which is combination of the algorithm of [10] and QAE. We also estimated the query complexity of our method, which does not have any dependence like $(1/\epsilon)^{\mathrm{poly}(d)}$ and shows tremendous speedup with respect to ϵ and d.

We believe that this paper is the first step for the research in this direction, but there remains many points to be improved. For example, we should consider whether the assumptions we made can be mitigated. For instance, although we assume that $\vec{C}(\tau)$ is time-independent (Assumption V.2), some products do not fit to this condition: e.g., when we consider the upper boundary condition in the case of the European-call-like payoff $f_{\mathrm{pay}}(S) = \max\{S - K, 0\}$ with some constant K, $V(t, S) \approx S - e^{-r(T-t)}K$ and therefore $Y(\tau, \vec{x}) = e^{r\tau}V(t, \vec{S})$ cannot be regarded as constant for large S. In order to omit this assumption, we might be able to extend the algorithm of [10] so that it can be applied to time-dependent $\vec{C}(\tau)$. As a future work, we will investigate the possibility of the quantum FDM for the wider range of derivatives.

[1] K. Miyamoto and K. Kubo, IEEE Trans. on Quantum Engineering 3, 3100225 (2022)
[2] P. Rebentrost et al., Phys. Rev. A 98, 022321 (2018)
[3] N. Stamatopoulos et al., Quantum 4, 291 (2020)
[4] K. Kaneko et al., arXiv:2007.01467 (2020)
[5] S. Chakrabarti et al., Quantum 5, 463 (2021)
[6] D. J. Egger et al., IEEE Trans. on Quantum Engineering 1, 3101724 (2020)
[7] A. Bouland et al., arXiv:2011.06492 (2020)
[8] D. J. Duffy, "Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach", Wiley (2006)
[9] D. W. Berry, Journal of Physics A 47, 10, 105301 (2014)
[10] D. W. Berry et al., Commun. Math. Phys. 356, 1057 (2017)
[11] T. Xin et al., Phys. Rev. A 101, 032307 (2020)
[12] A. M. Childs and J.-P. Liu, Commun. Math. Phys. 375, 1427 (2020)
[13] A. W. Harrow et al., Phys. Rev. Lett. 103, 150502 (2009)
[14] A. Ambainis, 29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012), pp. 636 (2012)
[15] A. M. Childs et al., SIAM J. Comput. 46, 1920 (2017)
[16] G. Brassard et. al., Contemporary Mathematics, 305, 53 (2002)
[17] Y. Suzuki et al., Quantum Inf. Process. 19, 75 (2020)
[18] L. Grover and T. Rudolph, arXiv:quant-ph/0208112 (2002)
[19] M. Mottonen et al., Quant. Inf. Comp. 5, 467 (2005)