Bermudan option pricing by quantum amplitude estimation and Chebyshev interpolation

Koichi Miyamoto
Center for Quantum Information and Quantum Biology, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan
(Dated: August 23, 2021)

Pricing of financial derivatives, in particular early exercisable options such as Bermudan options, is an important but heavy numerical task in financial institutions, and its speed-up will provide a large business impact. Recently, applications of quantum computing to financial problems have been started to be investigated. In this paper, we first propose a quantum algorithm for Bermudan option pricing. This method performs the approximation of the continuation value, which is a crucial part of Bermudan option pricing, by Chebyshev interpolation, using the values at interpolation nodes estimated by quantum amplitude estimation. In this method, the number of calls to the oracle to generate underlying asset price paths scales as \( \tilde{O}(\epsilon^{-1}) \), where \( \epsilon \) is the error tolerance of the option price. This means the quadratic speed-up compared with classical Monte Carlo-based methods such as least-squares Monte Carlo, in which the oracle call number is \( \tilde{O}(\epsilon^{-2}) \).

I. INTRODUCTION

Following the recent advances of quantum computing technologies\(^1\), many researches have been done for the application to practical problems in various industries. One of the promising targets is finance (see [2–4] for reviews). Financial firms have a lot of heavy computational tasks in their daily business\(^2\), and therefore the speed-up of such tasks by quantum computers are expected to provide a large impact. For example, previous papers studied option pricing\(^3\), risk measurement\(^4\), portfolio optimization\(^5\), and so on.

In this paper, we focus on Bermudan option pricing and consider how to speed it up by quantum algorithms. Let us briefly describe the problem. An option is a financial contract between two parties, the option buyer and seller, which conveys the option buyer the right to buy some underlying assets such as stocks and bonds from or sell them to the option seller, at some specified price on some date. Or, more generally, it can be regarded as a contract, in which the option buyer receives some amount of money (payoff) determined in reference to the underlying asset price, from the option seller.

There are some kinds of options with respect to timing of exercise of right. In an European option, the option buyer can exercise the right at one predetermined date, which is called the maturity. On the other hand, there are early-exercisable options, in which the option buyer can choose the exercise date. In an American option, the option buyer can exercise the right at any time before the final maturity \( T \). In a Bermudan option, exercise of right is possible on any of finite predetermined dates including \( T \). We hereafter call such dates exercise dates.

Major financial firms hold large portfolios of a wide variety of options, and therefore pricing them is an important task for business. However, it is also a difficult task. Basically, the option price is the expected value of the payoff under some stochastic model describing random time evolution of the underlying assets. Although European options can be sometimes priced easily, for example, by some analytic formulas, pricing Bermudan and American options typically involves heavy numerical calculations. The difficulty partly stems from the nature of the problem as dynamic programming. That is, pricing early-exercisable options contains determining the optimal exercise time as a crucial part. Although there are some kinds of methods which aim to reflect early exercise to the option price, each one has pros and cons.

One major category of pricing methods is the Monte Carlo-based method\(^6\), in which we generate many sample paths of evolution of underlying asset prices, and estimate the expected payoff as an averaged payoff over the paths. This approach has an advantage in the case of multiple underlying assets.

Namely, in this approach, the estimation error on the option price decays as \( \tilde{O}(N^{-1/2}) \) when the sample number \( N \) increases, regardless of the number of the underlying assets \( d \). In other words, it suffices to take \( \tilde{O}(\epsilon^{-2}) \) samples in order to achieve the error tolerance \( \epsilon \) on the option price. This contrasts to other methods, for example approaches based on solving partial differential equations\(^7\), whose complexity is \( \tilde{O}((1/\epsilon)^{\text{poly}(d)}) \). On the other hand, in the Monte Carlo-based methods, it is difficult to precisely determine the optimal exercise time, and we have to approximate this in some way. In many cases, this is done through approximation of the continuation value, which is the option price at each exercise date in the case that the option buyer forgos the exercise. The option should be exercised if the payoff is larger than the continuation value, and should not be exercised otherwise.

In this category, the least-squares Monte Carlo (LSM)\(^8\) is widely used. LSM estimates the continuation value at each exercise date by linear regression using the generated sample paths as training data, and then, going backward from the final maturity to the present, finds the present option price.

Note that this method can also price American options approximately, replacing exercisability at any point in the con-

---

\(^1\) koichi.miyamoto@qiqb.osaka-u.ac.jp

\(^2\) As a standard textbook for quantum computing, we refer to [1].

\(^3\) As standard textbooks for financial engineering, especially option pricing, we refer to [5, 6].

\(^4\) \( \tilde{O}(\cdot) \) hides logarithmic factors in the ordinary big \( O \) notation \( O(\cdot) \).

\(^5\) We refer to [26] as a textbook on Monte Carlo simulation and its application to finance.
In this paper, we propose a new method for Bermudan option pricing, combining Chebyshev interpolation and quantum algorithm for Monte Carlo integration \([30, 32]\), which is based on quantum amplitude estimation (QAE) \([31, 33, 34]\). As far as the author knows, this is the first proposal on the quantum method for Bermudan option pricing. Chebyshev interpolation is a widely used method for function approximation \([34, 50]\). In the proposed method, given the access to the quantum circuit (or, the oracle) for time evolution of underlying asset prices, we calculate the continuation values at the interpolation nodes (or the quantum algorithm), and find Chebyshev interpolation using these values. Importantly, this method outputs an estimation of the option price with the error at most \(\epsilon\), calling the oracle only \(\tilde{O}(\epsilon^{-1})\) times. Thus, as we commonly observed in applications of QAE to various kinds of problems, we obtain the quadratic speed-up compared with the classical Monte Carlo-based methods such as LSM and the Chebyshev interpolation-based methods.

The rest of this paper is organized as follows. In Section \([\text{II}]\), we briefly explain Chebyshev polynomials and function approximation by them. We present how to calculate the coefficients of Chebyshev expansion in general and the upper bound for the approximation error. In Section \([\text{III}]\) we present the general formulation of Bermudan option pricing and explain LSM as a typical classical solutions. In Section \([\text{IV}]\) we explain QAE and QAE-based Monte Carlo integration. Then, in Section \([\text{V}]\) we present the new algorithm for Bermudan option pricing based on Chebyshev interpolation and QAE. We also present an upper bound on the price error in the method, and that on the complexity sufficient to achieve the given error tolerance. Section \([\text{VI}]\) summarizes this paper. All proofs are presented in the appendix.

### A. Notations

We here explain the notations used in this paper.

\(\mathbb{N}\) denotes the set of all positive integers, and \(\mathbb{N}_0 := \{0\} \cup \mathbb{N}\). We define \([n]\) := \(\{i \in \mathbb{N} \mid i \leq n\}\) for any \(n \in \mathbb{N}\), and \([n]_0 := \{i \in \mathbb{N}_0 \mid i \leq n\}\) for any \(n \in \mathbb{N}_0\). We also define \(\mathbb{N}_{>n} := \{i \in \mathbb{N} \mid i > n\}\) for \(n \in \mathbb{N}\). Similarly, we define \(\mathbb{R}_{>a} := \{x \in \mathbb{R} \mid x > a\}\) and \(\mathbb{R}_{\geq a} := \{x \in \mathbb{R} \mid x \geq a\}\) for \(a \in \mathbb{R}\). \(\mathbb{R}_+\) denotes the set of all positive real number, that is, \(\mathbb{R}_{>0}\).

For \(a, b \in \mathbb{N}_0\), \(\delta_{a,b}\) denotes the Kronecker delta, which is 1 if \(a = b\) or 0 otherwise. For \(d \in \mathbb{N}\) and \(l_1, l_2 \in \mathbb{N}_0\), we also define \(\delta_{l_1, l_2}\), which is 1 if \(l_1 = l_2\) or 0 otherwise.

For a measure space \((\Omega, \mathcal{F}, \mu)\) and \(p \in \mathbb{R}_{\geq 1}\), \(L^p(\Omega, \mu)\) denotes the \(L^p\) space on it.

The indicator function \(1_{C}\) takes 1 if the condition \(C\) is satisfied, and 0 otherwise.

In this paper, we consider quantum states of systems consisting of some quantum registers with some qubits. For \(x \in \mathbb{R}\), \(|x\rangle\) denotes one of the computational basis states on some register, whose bit string corresponds to the binary representation of \(x\) with truncation at some digit. For \(d \in \mathbb{N}\) and \(\bar{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d\), \(|\bar{x}\rangle := |x_1\rangle \ldots |x_d\rangle\) is the state on the \(d\)-register system.

### II. APPROXIMATION OF FUNCTIONS BY CHEBYSHEV INTERPOLATION

For \(l \in \mathbb{N}_0\), the \(l\)-th Chebyshev polynomial (of the first kind) is defined as

\[
T_l(x) := \cos(l \cdot \arccos(x)),
\]

where \(x \in [-1, 1]\). One of its important properties is the discrete orthogonality: for any \(m \in \mathbb{N}_0\) and \(l_1, l_2 \in [m]_0\),

\[
\sum_{j=0}^{m} T_{l_1}(x_{m,j}) T_{l_2}(x_{m,j}) = \begin{cases} 0 & \text{if } l_1 \neq l_2 \\ m + 1 & \text{if } l_1 = l_2 = 0 \\ \frac{m + 1}{2} & \text{if } l_1 = l_2 > 0 \end{cases}.
\]

Here, \(x_{m,j}\) is the Chebyshev node defined as

\[
x_{m,j} = \cos\left(\frac{j + \frac{1}{2}}{m + 1} \pi\right)
\]

for \(j \in [m]_0\). \(x_{m,0}, \ldots, x_{m,m}\) are the zeros of \(T_{m+1}(x)\).

We also define the tensorized Chebyshev polynomials on a general hyperrectangle in \(\mathbb{R}^d\), where \(d \in \mathbb{N}\). That is, given

\[
D = [L_1, U_1] \times \ldots \times [L_d, U_d],
\]

with \(L_1, \ldots, L_d, U_1, \ldots, U_d \in \mathbb{R}\) satisfying \(L_1 < U_1, \ldots, L_d < U_d\), we define

\[
\tilde{T}_{D, \tilde{l}}(\bar{s}) := \prod_{i=1}^{d} T_{l_i}(\theta_{D, l_i}(s_i))
\]

for every \(\bar{l} = (l_1, \ldots, l_d)^T \in [0, \infty]^d\) and \(\bar{s} = (S_1, \ldots, S_d)^T \in D\), where

\[
\theta_{D, \bar{s}}(\bar{l}) := \left(\frac{2S_1 - U_1 - L_1}{U_1 - L_1}, \ldots, \frac{2S_d - U_d - L_d}{U_d - L_d}\right)^T.
\]

For the above polynomials, the orthogonality relation is now

\[
\sum_{\bar{s} \in G_{D, m}^l} \tilde{T}_{D, \tilde{l}}(\bar{s}) \tilde{T}_{D, \tilde{l}}(\bar{s}) = \begin{cases} (m+1)^l / 2^{l(l+1)/2} & \text{if } \tilde{l}_1 = \tilde{l}_2 \\ 0 & \text{if } \tilde{l}_1 \neq \tilde{l}_2 \end{cases}
\]

for every \(m \in \mathbb{N}\) and \(l_1, l_2 \in [m]_0\), where \(N(\bar{l}) := \#\{i \in [d] \mid l_i > 0\}\) for \(\bar{l} = (l_1, \ldots, l_d)^T \in [0, \infty]^d\), and \(G_{D, m}^l\) is the set of points \(S_1^{D, m}_{\bar{l}} \in D\) written in the form of

\[
S_1^{D, m}_{\bar{l}} := \left(\frac{U_1 - L_1}{2}x_{m, l_1} + \frac{U_1 + L_1}{2}, \ldots, \frac{U_d - L_d}{2}x_{m, l_d} + \frac{U_d + L_d}{2}\right)^T
\]
Let \( d \) and \( m \) be positive integers. Let \( \eta \) be a function such that \( \eta : \mathbb{N} \to \mathbb{R} \), where \( \eta_i = i \). Given the values of \( \eta \) at the Chebyshev nodes, we can consider the more general case, where the values of \( \eta \) are different for different nodes. We then define the multivariate version of the Lebesgue constant of the Chebyshev nodes: for every \( m \in \mathbb{N} \),

\[
\Lambda_{d,m} := \max_{(x_1, \ldots, x_d) \in [-1,1]^d} \sum_{j=1}^m \ell_j^{(m)}(x_i),
\]

where

\[
\ell_j^{(m)}(x) := \prod_{k=1}^d \frac{x - x_{m,k}}{x_{m,j} - x_{m,k}}
\]

for every \( j \in [m]_0 \). As [47], [51] showed, \( \Lambda_{d,m} \) is derived from the well-known upper bound \( \Lambda_{d,m} \leq \frac{2}{\pi} \log(m + 1) + 1 \) [43]. Then, the theorem is as follows.

**Theorem 1.** Let \( d \) and \( m \) be positive integers. Let \( D \) be a hyper-rectangle like \( [a,b]^d \). Let \( f : \mathbb{R}^d \to \mathbb{R} \) be a function that has an analytic extension to \( B_{D,p} \) for some \( p \in \mathbb{R} > 1 \). Besides, assume that \( \sup_{\tilde{S} \in B_D} |f(\tilde{S})| \leq B \) for some \( B \in \mathbb{R} \). Moreover, suppose that we are given a real number \( \tilde{f} \), for every \( \tilde{f} \in [m]_0 \), and that there exists \( \epsilon \in \mathbb{R} \) such that

\[
|f(\tilde{S}) - \tilde{f}| \leq \epsilon
\]

holds for every \( \tilde{f} \in [m]_0 \). Then,

\[
\max_{\tilde{S} \in D} |f(\tilde{S}) - \tilde{f}(\tilde{S})| \leq e^{im(\rho, d, m, B)} + \Lambda_{d,m}\epsilon
\]

holds. Here, for every \( \tilde{S} \in D \), \( \tilde{f}(\tilde{S}) \) is defined as

\[
\tilde{f}(\tilde{S}) := \sum_{j \in [m]_0} \tilde{a}_j \tilde{T}_{D,j}(\tilde{S}_j),
\]

with the coefficients \( \tilde{a}_j \) calculated by

\[
\tilde{a}_j := \frac{2n(j)}{(m+1)^d} \sum_{\tilde{S} \in G_D} \tilde{f}(\tilde{S}) \tilde{T}_{D,j}(\tilde{S})
\]

for every \( j \in [m]_0 \) and

\[
e^{im(\rho, d, m, B)} := 2^{3+1} \sqrt{d} B \rho^{-m} \left(1 - \rho^{-2}\right)^{-\frac{1}{2}}.
\]

**III. BERMUDAN OPTION PRICING**

**A. General formulation**

In this paper, we consider pricing a Bermudan option with \( d \in \mathbb{N} \) underlying assets and \( K \in \mathbb{N} \) exercise dates \( t_1, \ldots, t_K \), which satisfy \( t_0 < t_1 < \ldots < t_K \) with \( t_0 := 0 \) being the present and \( t_K := T \) being the final maturity. This is formulated as follows. Under some filtered probability space \((\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \geq 0}, \mathbb{P})\), consider the \( S \)-valued Markov process \( \tilde{S}(t) := (S_1(t), \ldots, S_d(t))^T \), where \( S \) is a subset of \( \mathbb{R}^d \) equipped with its Borel \( \sigma \)-algebra inherited from \( \mathbb{R}^d \), and \( \tilde{S}_0 := S(0) \) is deterministic. \( \tilde{S}(t) \) corresponds to the values of the underlying asset prices at time \( t \). 

6 [42], [58] considered the more general case, where the values of \( \rho \) and \( m \) are different for different directions in \( \mathbb{R}^d \). In this paper, we take common values of \( \rho \) and \( m \) for every direction, for simplicity.
Then, the price of the Bermudan option at $t_k$ with $\hat{S}_k = \hat{s} \in S$ is given as

$$V_k(\hat{s}) := \sup_{\tau \in \mathcal{T}_k} \mathbb{E}[f^\text{pay}(\hat{s}_\tau)|\hat{s}_t = \hat{s}],$$

(19)

where $\mathbb{E}[\cdot]$ denotes the (conditional or unconditional) expected value with respect to $\mathbb{P}$, and $\mathcal{T}_k$, $k \in [K]$ is the set of all $[k, ..., K]$-valued stopping times. In particular, the present option price is

$$V_0(\hat{s}) := \sup_{\tau \in \mathcal{T}} \mathbb{E}[f^\text{pay}(\hat{s}_\tau)],$$

(20)

where $\mathcal{T} = \mathcal{T}_1$.

The problem to find $V_0$ can be written as a kind of dynamic programming, that is,

$$V_k(\hat{s}) = \begin{cases} f^\text{pay}(\hat{s}) & ; k = K \\ \max[f^\text{pay}(\hat{s}), Q_k(\hat{s})] & ; k = 1, ..., K - 1 \end{cases}$$

(21)

for every $\hat{s} \in S$, and

$$V_0 = \mathbb{E}[V_1(\hat{S}_1)].$$

(22)

Here, for every $k \in [K - 1]$ and $\hat{s} \in S$,

$$Q_k(\hat{s}) := \mathbb{E}[V_{k+1}(\hat{S}_{k+1})|\hat{S}_k = \hat{s}],$$

(23)

is called the continuation value. This corresponds to the option price at $t_k$ in the case that the option has not been exercised at that time and that $\hat{S}_k = \hat{s}$.

Note that this problem can be seen as finding the optimal exercise date $\tau_{\text{opt}} \in \mathcal{T}$, which maximizes (20). This can be recursively determined as

$$\tau_K = K,$$

$$\tau_k = k \mathbb{1}_{j_k^\text{pay}(\hat{S}_k) \geq Q_k(\hat{S}_k)} + \tau_{k+1} \mathbb{1}_{j_k^\text{pay}(\hat{S}_k) < Q_k(\hat{S}_k)}, k \in [K - 1]$$

(24)

and $\tau_{\text{opt}} = \tau_1$. Also note that

$$Q_k(\hat{s}) := \mathbb{E}[f^\text{pay}(\hat{s}_\tau)|\hat{s}_k = \hat{s}],$$

(25)

for every $k \in [K - 1]$, which means that $Q_k$ is the expected value of the payoff under the exercise strategy $\tau_{k+1}$.

Next, we generate $N_{\text{samp}} \in \mathbb{N}_{>2}$ sample paths of underlying asset prices, which are denoted as $\mathcal{S}_i = (\tilde{S}_0, \tilde{S}_1^{(i)}, ..., \tilde{S}_K^{(i)})$, $i \in [N_{\text{samp}}]$. Then, we determine the stopping time, which approximate the optimal one $\tau_{\text{op}}$, by the following procedure. First, we set $\mathcal{T}_{K,i} = K$ for every $i \in [N_{\text{samp}}]$. For $k \in [K - 1]$, given $\tau_{k+1,i}$ for every $i \in [N_{\text{samp}}]$, we determine the approximate continuation value $\hat{Q}_k$ as the element $g_k$ in $\mathcal{R}_{d,m}$, which minimizes

$$\frac{1}{N_{\text{samp}}} \sum_{i=1}^{N_{\text{samp}}} (g_k(\tilde{S}_k^{(i)}) - f^\text{pay}(\hat{s}_k^{(i)}))^2,$$

(26)

or, in other words, best fits to the realized payoffs under the stopping time $\tau_{k+1,i}$ on the sample paths. It is guaranteed by statistical learning theory that fitting to the sample values of the payoff, which distribute around the continuation value, yields the approximation of the continuation value $[55, 57, 59, 61]$.

Then, we set

$$\tilde{\tau}_{k,i} = \begin{cases} k & ; \text{if } \hat{Q}_k(\tilde{S}_k^{(i)}) \leq f^\text{pay}(\tilde{S}_k^{(i)}) \\ \tau_{k+1,i} & ; \text{otherwise} \end{cases}$$

(27)

for every $i \in [N_{\text{amp}}]$. By repeating this until we reach $k = 1$, we get $\tilde{\tau}_{1,i}$, and finally

$$\tilde{V}_0 := \frac{1}{N_{\text{amp}}} \sum_{i=1}^{N_{\text{amp}}} f^\text{pay}(\hat{s}_1^{(i)}),$$

as an approximation of $V_0$.

Let us make some comments on the procedure. First, note that it is assumed that we can generate sample paths $\mathcal{S}_i$. In the usual situation where the stochastic differential equation (SDE) for $\tilde{S}(t)$ is given, we can use some method for numerical simulation of SDEs such as Euler-Maruyama method. Second, we mention how to find $g_k$ minimizing (26). Note that this is just least-squares linear regression, since $\mathcal{R}_{d,m}$ is a vector space. Therefore, we can solve this by various methods, for example, solving the normal equation of linear regression, some numerical optimization, and so on.

Then, let us mention the relationship between the error and the sample number in LSM. According to [61], under some technical assumptions, taking appropriately large $m$, we obtain the error bound on the option price which scales as

$$\mathbb{E}_{\text{amp}}[|\tilde{V}_0 - V_0|] = \tilde{O}\left(N_{\text{amp}}^{\frac{p-2}{2(p+2)}} \right).$$

(28)

Here, $\mathbb{E}_{\text{amp}}[\cdot]$ denotes the expectation with respect to random samples, and $n$ and $p$ are the quantities which characterize smoothness of $Q_1(\hat{S}), ..., Q_{K-1}(\hat{S})$ and boundedness of the norms of $f^\text{pay}(\hat{s}), ..., f^\text{pay}(\hat{s})$, respectively (see [61] for more details). For larger $p$ and $n$, the RHS of (28) decreases faster against the increase of $N_{\text{amp}}$. In the limit of $n, p \to \infty$, which means that $Q_k$’s are highly smooth and the norms of $f^\text{pay}$’s are well-bounded, the RHS of (28) becomes $\tilde{O}(N_{\text{amp}}^{-1/2})$, which coincides with the well-known error decay rate in Monte Carlo integration. Conversely, in this limit, it is sufficient to take $\tilde{O}(\varepsilon^{2})$ samples in order to achieve the error tolerance $\varepsilon$.
IV. QUANTUM AMPLITUDE ESTIMATION AND QUANTUM ALGORITHM FOR MONTE CARLO INTEGRATION

A. Quantum amplitude estimation (QAE)

We here briefly review QAE. Consider the system consisting of a quantum register $R_1$ and a qubit $R_2$. Suppose that we are given the oracle $A$, which transforms $|0\rangle |0\rangle$, the state in which all qubits in $R_1$ and $R_2$ are set to $|0\rangle$, into

$$A |0\rangle |0\rangle = \sqrt{a} |\psi_1\rangle |1\rangle + \sqrt{1-a} |\psi_0\rangle |0\rangle =: |\Psi\rangle$$  \hspace{1cm} (29)

with some $a \in (0,1)$. Here, the first and second qets correspond to $R_1$ and $R_2$ respectively, and $|\psi_0\rangle, |\psi_1\rangle$ are some quantum states. Then, our goal is estimating $a$, which is the probability to obtain 1 in $R_2$ when we measure $|\Psi\rangle$, with the error tolerance $\epsilon$. There exist some algorithms which output such an estimation with $O(\epsilon^{-1})$ calls to $A$ and its inverse $A^\dagger$ in total\cite{13,63}. Although these QAE algorithms output a number close to $a$ not with certainty but with some probability, we can enhance the success probability by running QAE many times and take the median of the outputs\cite{13,63}. Let us define the $(N_{\text{QAE}}, N_{\text{rep}})$-QAE as the method for estimating $a$ which runs $N_{\text{rep}}$ rounds of QAE and makes $N_{\text{QAE}}$ calls to $A$ and $A^\dagger$ in total in each round. Obviously, in the $(N_{\text{QAE}}, N_{\text{rep}})$-QAE, $A$ and $A^\dagger$ are called $N_{\text{QAE}}/N_{\text{rep}}$ times in total. Then, combining Theorem 12 in \cite{62} and Lemma 6.1 in \cite{62}, we obtain the following theorem.

**Theorem 2.** Suppose that we are given the accesses to $A$ in (29) and its inverse $A^\dagger$. Then, for any $y \in (0,1)$ and $\epsilon \in (0,0.1)$, a $(N_{\text{QAE}}, N_{\text{rep}})$-QAE, where the positive integers $N_{\text{QAE}}$ and $N_{\text{rep}}$ satisfy

$$N_{\text{QAE}} \geq \frac{7}{\epsilon}$$  \hspace{1cm} (30)

and

$$N_{\text{rep}} \geq 12\lfloor \log(y^{-1}) \rfloor + 1$$  \hspace{1cm} (31)

respectively, outputs $\tilde{a} \in \mathbb{R}$ such that

$$|\tilde{a} - a| \leq \frac{7}{N_{\text{QAE}}} \leq \epsilon,$$  \hspace{1cm} (32)

where $\tilde{a}$ is derived as (29), with probability higher than $1-y$.

Here, (30) is derived from the inequality in Theorem 12 in \cite{63} with $k = 1$, that is,

$$|\tilde{a} - a| \leq \frac{2\pi \sqrt{a(1-a)}}{M} + \frac{\pi^2}{M^2},$$  \hspace{1cm} (33)

where $M = N_{\text{QAE}}/2$ under the current definition. Using $\sqrt{a(1-a)} \leq \frac{1}{4}$, we see that (33) implies $|\tilde{a} - a| \leq 7/N_{\text{QAE}}$ for $N_{\text{QAE}} \geq 70$, which follows from (30) and $0 < \epsilon < 0.1$. In summary, if $N_{\text{QAE}}$ satisfies (30) for $\epsilon \in (0,0.1)$, the error in QAE is suppressed to at most $\epsilon$ with high probability. Hereafter, we say that a $(N_{\text{QAE}}, N_{\text{rep}})$-QAE succeeded if it output $\tilde{a}$ such that $|\tilde{a} - a| \leq 7/N_{\text{QAE}}$.

B. Quantum algorithm for Monte Carlo integration

One application of QAE is the algorithm for Monte Carlo integration, that is, the method to calculate expected values. Suppose that we want to calculate $\mathbb{E}[F(\tilde{X})]$, the expected value of $F(\tilde{X})$, where $\tilde{X}$ is some real vector-valued stochastic variable and $F$ is a real-valued function acting on $\tilde{X}$. We also assume that the range of $F$ is in $[0,1]$, and, if not, we make this hold by adding and/or multiplying some constants to $F$. Furthermore, suppose that we can use the following oracles $O_X$ and $O_F$. $O_X$ is the oracle to generate the state in which the distribution of $\tilde{X}$ is encoded. That is, $O_X$ operates on a quantum register and transform the state with all qubits set to $|0\rangle$ into

$$O_X |0\rangle = \sum_{i=1}^{N_\tilde{x}} \sqrt{p_i} |\tilde{x}_i\rangle,$$  \hspace{1cm} (34)

where $\tilde{x}_1, ..., \tilde{x}_{N_\tilde{x}}$ are $N_\tilde{x} \in \mathbb{N}$ possible values of $\tilde{X}$ and $p_i, i \in [N_\tilde{x}]$ is the probability that $\tilde{X} = \tilde{x}_i$. Here, we assume that the set of all the values that $\tilde{X}$ can take is finite. If $\tilde{X}$ is continuous, we need some discretization. How to create states corresponding to widely used distributions such as normal distribution has been investigated\cite{13,62}. The second oracle $O_F$ operates on a two-register system, and, using the first register as the input $\tilde{x}$, outputs $F(\tilde{x})$ into the second register. That is, for any $\tilde{x}$ in the domain of $F$,

$$O_F |\tilde{x}\rangle = |\tilde{x}\rangle |F(\tilde{x})\rangle.$$  \hspace{1cm} (35)

By these oracles, the following computation is possible. Preparing two registers $R_1, R_2$ and a qubit $R_3$, and initializing all of them to $|0\rangle$, we perform

$$|0\rangle |0\rangle$$

$$\rightarrow \sum_{i=1}^{N_\tilde{x}} \sqrt{p_i} |\tilde{x}_i\rangle |0\rangle |0\rangle$$

$$\rightarrow \sum_{i=1}^{N_\tilde{x}} \sqrt{p_i} |\tilde{x}_i\rangle |F(\tilde{x}_i)\rangle |0\rangle$$

$$\rightarrow \sum_{i=1}^{N_\tilde{x}} \sqrt{p_i} |\tilde{x}_i\rangle |F(\tilde{x}_i)\rangle \left( \sqrt{F(\tilde{x}_i)} |1\rangle + \sqrt{1-F(\tilde{x}_i)} |0\rangle \right),$$  \hspace{1cm} (36)

where the first, second and third qets correspond to $R_1, R_2$ and $R_3$, respectively. We use $O_X$ and $O_F$ at the first and second arrows, respectively. The transformation at the third arrow is done by arithmetic circuits\cite{64} and controlled rotation gates. Note that the probability to obtain 1 in $R_3$ when we measure the final state in (36) is $\sum_{i=1}^{N_\tilde{x}} p_i F(\tilde{x}_i)$, that is, $\mathbb{E}[F(\tilde{X})]$. Therefore, using the whole operation in (36) as the oracle $A$, we can estimate $\mathbb{E}[F(\tilde{X})]$ by QAE.
V. BERMUDAN OPTION PRICING BY CHEBYSHEV INTERPOLATION AND QAE

Now, let us present the method for Bermudan option pricing by Chebyshev interpolation and QAE.

A. Assumptions

We begin with making some assumptions necessary to execute the proposed method. The first one is as follows.

**Assumption 1.** We are given the access to the oracle $O_{\text{step}, k}$, which generates the state corresponding to the probability distribution of $\tilde{S}_{k+1}$ conditional on $\tilde{S}_{k}$. That is, for every $k \in [K-1]_0$ and $\tilde{S} \in \mathcal{S}$,

\[
O_{\text{step}, k} : |\tilde{S}\rangle |0\rangle \mapsto \sum_{\tilde{s} \in \mathcal{S}_{k+1}(\tilde{S})} \sqrt{p_{k+1}(\tilde{s}, \tilde{S})} |\tilde{s}\rangle |\tilde{S}\rangle, \tag{37}
\]

where $\mathcal{S}_{k+1}(\tilde{S})$ is the set of possible values of $\tilde{S}_{k+1}$ under the condition that $\tilde{S}_{k} = \tilde{S}$, and

\[
\begin{align*}
p_{k+1}(\tilde{s}, \tilde{S}) := & \mathbb{P}\left(\tilde{S}_{k+1} = \tilde{s} \mid \tilde{S}_{k} = \tilde{S}\right). \tag{38}
\end{align*}
\]

We here make comments on how to implement $O_{\text{step}, k}$. As mentioned in Section III, usually, following some SDE and some numerical method such as Euler-Maruyama, we can generate random sample values of $\tilde{S}_{k+1}$ with the given value of $\tilde{S}_{k}$ as the initial condition. Implementations of such a calculation on quantum circuits have been discussed in the previous papers [2] [3] [13]. That is, we can prepare the states corresponding to some (discretely approximated) random variables (e.g. standard normal) on the other registers, and, using them at discretized time steps, generate the path of $\tilde{S}(t)$ from $t_k$ to $t_{k+1}$. This yields the state like $|\tilde{S}\rangle$. We should also note that, in Assumption 1, it is assumed that $\tilde{S}_{k+1}$ can take only a finite number of values for the fixed $\tilde{S}_{k}$. This is not the case in the most models of $\tilde{S}(t)$, in which it takes continuous values. However, under the aforementioned implementations for time evolution of $\tilde{S}(t)$, in both time and random variables are discretely approximated, the number of the possible values of $\tilde{S}_{k}$ necessarily becomes finite.

Hereafter, we are mainly interested in the number of calls to $O_{\text{step}, k}$ in calculating the option price as a measure of complexity, since calculation for time evolution of underlying asset prices is typically the most time-consuming part in option pricing.

The second assumption is as follows. Here, $\mathcal{I}_{\mathcal{A}}$ denotes the set of all real-valued functions on a given subset $\mathcal{A} \subseteq \mathbb{R}^d$.

**Assumption 2.** For every $k \in [K-1]$, we are given the following

- the hyper-rectangle $\mathcal{D}_k := [L_{1,k}, U_{1,k}] \times \ldots \times [L_{d,k}, U_{d,k}] \subseteq \mathcal{S}$, with $L_{1,k}, \ldots, L_{d,k}, U_{1,k}, \ldots, U_{d,k} \in \mathbb{R}$ satisfying $L_{1,k} < U_{1,k}, \ldots, L_{d,k} < U_{d,k}$,

- $V^\text{OB}_k \in \mathcal{I}_{\mathcal{S} \setminus \mathcal{D}_k}$ such that the following (i) and (ii) are satisfied.

  (i) There exists $\epsilon^\text{OB}_k \in \mathbb{R}_+$ such that either

  \[
  |V^\text{OB}_k(\tilde{s}) - V_k(\tilde{s})| < \epsilon^\text{OB}_k \tag{39}
  \]

  or

  \[
  |F_k(\tilde{s}) - V_k(\tilde{s})| < \epsilon^\text{OB}_k \tag{40}
  \]

  is satisfied for any $\tilde{s} \in \mathcal{S} \setminus \mathcal{D}_k$. Here, $F_k[.]$ is the ‘flat extrapolation operator’ defined as

  \[
  F_k(F)(\tilde{s}) := F(b_k(\tilde{s})) \tag{41}
  \]

  \[
  b_k(\tilde{s}) := (\min[L_{1,k}, \max[L_{1,k}, s_1]], \ldots, \min[U_{d,k}, \max[L_{d,k}, s_d]])^T \tag{42}
  \]

  for any $F \in \mathcal{I}_{\mathcal{D}_k}$ and $\tilde{s} = (s_1, \ldots, s_d)^T \in \mathcal{S}$.

  (ii) If, for some $G \in \mathcal{I}_{\mathcal{D}_k}$, we have the access to the oracle $O_G$ such that

  \[
  O_G[\tilde{s}] |0\rangle = |\tilde{s}\rangle |G(\tilde{s})\rangle \tag{43}
  \]

  for any $\tilde{s} \in \mathcal{D}_k$, we also have the access to the oracle $\tilde{O}_G$, which acts as

  \[
  \tilde{O}_G[\tilde{s}] |0\rangle = |\tilde{s}\rangle |G_k[\tilde{G}(\tilde{s})]\rangle. \tag{44}
  \]

Here, $G_k[\cdot]$ is defined as

\[
G_k[H](\tilde{s}) := \begin{cases} V^\text{OB}_k(\tilde{s}) & \text{if } \tilde{s} \in \mathcal{A}_k \\ F_k[H](\tilde{s}) & \text{otherwise} \end{cases} \tag{45}
\]

for any $H \in \mathcal{I}_{\mathcal{D}_k}$ and $\tilde{s} \in \mathcal{S}$, where $\mathcal{A}_k$ is a subset of $\mathcal{S} \setminus \mathcal{D}_k$ such that (39) and (40) hold for any $\tilde{s} \in \mathcal{A}_k$ and any $\tilde{s} \in (\mathcal{S} \setminus \mathcal{D}_k) \setminus \mathcal{A}_k$, respectively.

We also define $G_k[\tilde{H}](\tilde{s}) := H(\tilde{s})$ for any $H \in \mathcal{I}_{\mathcal{S}}$ and $\tilde{s} \in \mathcal{S}$.

Roughly speaking, this assumption means that, when some of underlying asset prices are extremely large or small, we can approximate the option value $V_k$ by some known and easily computable function $V^\text{OB}_k$ or the flat extrapolation of $V_k$ from moderate underlying asset prices. Postponing explanation on why this assumption is necessary to Section VIII we here see that it is actually satisfied in some typical settings in option pricing. For example, let us consider a basket put option, whose payoff function is $\int_0^\infty (s_1, \ldots, s_d)^T = \max[k - s_1 - \ldots - s_d, 0]$ with some $k \in \mathbb{R}$ for every $k \in [K]$, under some model in which $S_1(t), \ldots, S_d(t)$ are unbounded from above but bounded from below, say, by $0$, as the Black-Scholes model. Then, in each of the following situations, (39) or (40) holds.

- If some of $S_1(t), \ldots, S_d(t)$ are extremely large, the option is far out-of-money, and therefore its price is almost 0.
• If some of $S_{1,k}, \ldots, S_{d,k}$ are smaller than the sufficiently small thresholds $L_{1,k}, \ldots, L_{d,k} \in \mathbb{R}_+$ respectively, but the others are not, setting the former to the thresholds hardly affects the option price.

• If all of $S_{1,k}, \ldots, S_{d,k}$ are sufficiently close to 0, the option is exercised at $t_k$, and therefore $V_k(S_k) = f_k^{\text{pay}}(S_k)$.

Thirdly, we make the following assumption, which is necessary for bounding the interpolation error in the proposed method.

**Assumption 3.** For every $k \in [K-1]$, $Q_k(S)$ has an analytic extension to $D_k$, where $D_k$ is given in Assumption 2 and $\rho_k$ is some real number greater than 1, and

$$\sup_{\bar{S} \in D_k \cap \mathbb{R}_+} |Q_k(S)| \leq B_k$$  \hspace{1cm} (46)

holds, where $B_k$ is some positive real number.

### B. The proposed method

Under these assumptions, we can construct the procedure for Bermudan option pricing based on QAE and Chebyshev interpolation. This is also a backward calculation similar to LSM; we sequentially calculate the approximate continuation value $\bar{Q}_k$ and option price $\bar{V}_k$ at $t_k$, going from the final maturity to the present. Roughly, the outline is as follows. As preparation, for every $k \in [K-1]$, we set $m_k \in \mathbb{N}$, the degree of Chebyshev polynomials used for the approximation, and the hyper-rectangle $D_k = [L_{1,k}, U_{1,k}] \times \cdots \times [L_{d,k}, U_{d,k}] \subseteq S$. We begin the iterative calculation by setting $\bar{V}_k(S) := f_k^{\text{pay}}(S)$ for every $S \in S$. Then, for $k \in [K-1]$, given $\bar{V}_{k+1}$, we estimate the expected value of $\bar{V}_{k+1}(S_{k+1})$ under the condition that $S_{k+1} = S_{k+1}^{D_k,m_k}$ for every Chebyshev node $S_{k+1}^{D_k,m_k}$ by QAE, and denote the estimation as $\bar{Q}_k^{QAE}$. Using these, we construct $\bar{Q}_k$, the Chebyshev interpolation of the approximate continuation value, and set $\bar{V}_k(S) = \max\{\bar{Q}_k(S), f_k^{\text{pay}}(S)\}$ for every $S \in S$. We repeat these steps until we reach $k = 1$. Finally, we estimate the expected value of $\bar{V}_1(S_1)$ by QAE again, and let the result be an approximation of $V_0$.

The fully detailed procedure is shown in Algorithm 1.

---

**Algorithm 1** The method for Bermudan option pricing based on Chebyshev interpolation and QAE

**Input:**

- $m_k \in \mathbb{N}$ for every $k \in [K-1]$, the degree of Chebyshev polynomials.
- $N_k^{QAE} \in \mathbb{N}$ for every $k \in [K-1]$, the iteration number in each run of QAE in calculating $\hat{V}_k^{QAE}$ or $\bar{V}_0$.
- $N_k^{\text{stop}} \in \mathbb{N}$ for every $k \in [K-1]$, the number of rounds of QAE in calculating $\hat{V}_k^{QAE}$ or $\bar{V}_0$.
- $\tilde{V}_k^{\text{max}} \in \mathbb{R}_+$ for every $k \in [K]$, the upper bound of $|G_k(\bar{V}_k(S))|$ for Chebyshev interpolation. Here, $U_{1,k}, \ldots, U_{d,k} \in S$ satisfy $L_{1,k} < U_{1,k}, \ldots, L_{d,k} < U_{d,k}$.
- $D_k = [L_{1,k}, U_{1,k}] \times \cdots \times [L_{d,k}, U_{d,k}] \subseteq S$ for every $k \in [K-1]$, the hyper-rectangle for Chebyshev interpolation.
- Here, $L_{1,k}, \ldots, L_{d,k}, U_{1,k}, \ldots, U_{d,k} \in S$ satisfy $L_{1,k} < U_{1,k}, \ldots, L_{d,k} < U_{d,k}$.

**Steps:**

1: Set $\bar{V}_k(S) := f_k^{\text{pay}}(S)$ for every $S \in S$.
2: for $k = K - 1$ to 1 do
3: for all $j \in \mathcal{F}_k$ do
4: Using $(N_k^{QAE}, N_k^{\text{stop}})$-QAE, obtain an estimation $\hat{P}_{k,j}$ of the probability $P_{k,j}$ to observe 1 on the last qubit in measuring $|\Psi_{k,j}\rangle$ in (49), and let $(2\hat{P}_{k,j} - 1)\tilde{V}_k^{\text{max}}$ be $\hat{V}_k^{QAE}$.
5: end for
6: Set
$$\tilde{V}_k(S) := \sum_{j \in \mathcal{F}_k} \tilde{a}_{k,j} \tilde{T}_{D_k,j}(S)$$  \hspace{1cm} (47)

for every $S \in D_k$, with $\tilde{a}_{k,j}$ calculated as
$$\tilde{a}_{k,j} := \frac{2^{m_j}}{(m_j + 1)^2} \sum_{j \in \mathcal{F}_k} \tilde{Q}_{k,j}^{QAE} \tilde{T}_{D_k,j}(S_j^{D_k,m_j})$$  \hspace{1cm} (48)

for $j \in \mathcal{F}_k$.
7: Set $\tilde{V}_k(S) := \max\{\tilde{V}_k(S), \tilde{Q}_k(S)\}$ for any $S \in D_k$.
8: end for
9: Using $(N_0^{QAE}, N_0^{\text{stop}})$-QAE, obtain an estimation $\tilde{P}_0$ of the probability $P_0$ to observe 1 on the last qubit in measuring $|\Psi_0\rangle$ in (52), and output $(2\tilde{P}_0 - 1)\tilde{V}_0^{\text{max}}$ as $\bar{V}_0$.

Some additional explanations should be made. The first one is on $|\Psi_{k,j}\rangle$ in Step 4. For every $k \in [K-1]$ and $j \in \mathcal{F}_k$, given the approximation $V_{k+1} \in I_{D_k}$ of $V_{k+1}$, we generate the state $|\Psi_{k,j}\rangle$ on the appropriate multi-register system with the last one being single-qubit, by the following operation:

- $|0\rangle|0\rangle|0\rangle|0\rangle$
- $\rightarrow |S_j^{D_k,m_j}\rangle|0\rangle|0\rangle|0\rangle$
- $\rightarrow |S_j^{D_k,m_j}\rangle \sum_{\tilde{S} \in S_{k+1}^{D_k,m_j}} \sqrt{P_{k+1}(\tilde{S}, S_j^{D_k,m_j})} |0\rangle|0\rangle|0\rangle|0\rangle$
- $\rightarrow |S_j^{D_k,m_j}\rangle \sum_{\tilde{S} \in S_{k+1}^{D_k,m_j}} \sqrt{P_{k+1}(\tilde{S}, S_j^{D_k,m_j})} |\tilde{S} G_{k+1}(\tilde{V}_{k+1}(\tilde{S}))|0\rangle
where \( O_{k,j} \) is imposed, such a probability becomes non-negligible.

### C. Evaluation of the error

Then, let us consider the error on the present option price in the proposed method. First, we have the following theorem.

**Theorem 3.** Under Assumptions 7 to 10 consider Algorithm 1. Suppose that, for every \( k \in \{K - 1\} \) and \( j \in J_k \),

\[
\left| \overline{Q}_k \left( \tilde{S}^{D_{k,m_j}}_j \right) - \overline{Q}^{\text{QAE}}_k \right| \leq \epsilon_k^{\text{QAE}}
\]

is satisfied, where \( \epsilon_k^{\text{QAE}} \) is some positive real number. Moreover, suppose that

\[
|V_0 - \overline{V}_0| \leq \epsilon_0^{\text{QAE}}
\]

is satisfied for some \( \epsilon_0^{\text{QAE}} \in \mathbb{R}_+ \). Then,

\[
|V_0 - \overline{V}_0| \leq \sum_{k=1}^{K-1} \Lambda_{1,k-1} \epsilon_k^{\text{int}} + \sum_{k=1}^{K-1} \Lambda_{1,k-1} \epsilon_k^{\text{OB}} + \sum_{k=0}^{K-1} \Lambda_{1,k} \epsilon_k^{\text{QAE}}
\]

holds, where, for \( k \in \{K - 1\} \) and \( k' \in \{K - 1\}_0 \),

\[
\epsilon_k^{\text{int}} := \epsilon_{\text{int}}(\rho_k, d, m_k, B_k)
\]

and

\[
\Lambda_{k,k'} := \begin{cases} 
\prod_{i=1}^{k'} \Lambda_i & \text{if } k \leq k' \\
1 & \text{otherwise}
\end{cases}
\]

and

\[
\Lambda_k := \left( \frac{2}{\pi} \log(m_k + 1) + 1 \right)^d.
\]

The proof is presented in Appendix A.

### D. Complexity

Based on Theorem 3, we can evaluate the complexity, that is, the number of calls to \( O_{\text{step},k} \) sufficient to achieve the desired level of the error on the present option price.

**Corollary 1.** Let \( \epsilon \) be a real number in \((0, 0.1)\). Under Assumptions 7 to 10 consider Algorithm 1 with the following parameters:
(i) $m_k, k \in [K-1]$ satisfying $m_k \geq m_k^{th}$ with

$$m_1^{th} = \frac{1}{\log \rho_1} \log \left( \frac{2^{d/2+2} \sqrt{d}(K-1)(1-\rho_1^{-2})^{-d/2}B_1}{V_1^{\max}} \right)$$

$$m_k^{th} = \frac{1}{\log \rho_k} \log \left( \frac{2^{d/2+2} \sqrt{d}(K-1)(1-\rho_k^{-2})^{-d/2} \tilde{N}^{th}_{1,k-1}B_k}{V_1^{\max}} \right)$$

for $k = 2, \ldots, K-1$, (61)

where $\tilde{N}^{th}_{1,k-1}$ is determined as $\tilde{N}_{1,k-1}$ in (59) with $m_1 = m_1^{th}, \ldots, m_{k-1} = m_{k-1}^{th}$.

(ii) $N_k^{QAE}$, $k \in [K-1]$ set as

$$N_k^{QAE} = \left\lceil \frac{7}{\epsilon_k} \right\rceil.$$ (62)

Here, $\epsilon_0, \ldots, \epsilon_{K-1}$ are given by

$$\epsilon_0 = \frac{1}{1 + \sum_{k'=1}^{K-1} (m_{k'} + 1)^d \Lambda_{1,k'}} \cdot \frac{\epsilon}{4}$$

and

$$\epsilon_k = \frac{\sqrt{(m_{k} + 1)^d / \Lambda_{1,k}}}{1 + \sum_{k'=1}^{K-1} (m_{k'} + 1)^d \Lambda_{1,k'}} \cdot \frac{V_1^{\max} \epsilon}{4V_1^{\max}},$$

for $k = 1, \ldots, K-1$, (63)

where $m_0, \ldots, m_{K-1}$ are set as (i) and $\tilde{N}_{1,1}, \ldots, \tilde{N}_{1,K-1}$ are given as (59) with such $m_0, \ldots, m_{K-1}$.

(iii) $N_k^{rep}$ set to

$$N_k^{rep} := 12 \left\lceil \log \left( \frac{N_{est}}{0.01} \right) \right\rceil + 1$$

for every $k \in [K-1]$. Here, $N_{est} := 1 + \sum_{k'=1}^{K-1} (m_{k'} + 1)^d$ with $m_1$ set as (i).

Moreover, suppose that $\epsilon_0^{OB}, \ldots, \epsilon_{K-1}^{OB}$ are 0. Then, Algorithm 7 outputs $V_0$ satisfying $|V_0 - \bar{V}_0| \leq \epsilon V_1^{\max}$ with probability higher than 0.99.

The proof is presented in Appendix B.

We here explain why the parameters are set as above. As we see in the proof in Appendix B, $m_1, \ldots, m_{K-1}$ satisfying (61) make the first term in the RHS in (57) smaller than $\epsilon V_1^{\max}/2$. Then, for such $m_k = 0, \ldots, K-1$, $N_k^{QAE}$, $k = 0, \ldots, K-1$ are determined as (62) so that

$$\frac{N_{tot}}{N_k^{rep}} = N_k^{QAE} + \sum_{k=1}^{K-1} (m_k + 1)^d N_k^{QAE},$$

(65)

that is, the total number $N_{tot}$ of calls to $\{O_{step,k}\}_{k=0,\ldots,K-1}$ divided by the QAE repetition number $N_k^{rep}$ is minimized under the constraint that, if all the QAEs in Algorithm 1 succeed, the third term in the RHS in (52) is smaller than $\epsilon V_1^{\max}/2$. Finally, $N_k^{rep}$, $k = 0, \ldots, K-1$ are determined so that the probability that these QAEs all succeed becomes higher than 0.99 $= 1 - 0.01$. In total, Algorithm 1 with the setting in Corollary 1 gives an approximation of $V_0$ with an error at most $\epsilon V_1^{\max}$ with probability higher than 0.99.

Note that, in reality, it is difficult to set $m_k$ to $m_k^{th}$, since $\rho_k$ and $B_k$ are usually unknown. In practice, we might set them to some conservatively large values, based on, for example, the calculation results of some benchmark pricing problems for various $(m_k)_{k=1,\ldots,K-1}$. Besides, note that, in the above setting, the half of the error tolerance $\epsilon$ is assigned to the interpolation error and another half is assigned to the QAE error. Although we can of course change this assignment ratio, it affects the complexity only logarithmically, since the sufficient levels of $(m_k)_{k=1,\ldots,K-1}$ are logarithmically affected by such a change and so are $N_k^{rep}$ compensating the change of $(m_k)$.

Let us consider the dependency of the total complexity on the error tolerance $\epsilon$. We see that

$$m_k = O\left( \log \left( \epsilon^{-1} \log \log \log (\epsilon^{-1}) \right) \right),$$

(66)

$$N_k^{QAE} = O\left( \epsilon^{-1} \times \log \log (\epsilon^{-1}) \right),$$

(67)

for every $k \in [K-1]$, and that

$$N_0^{QAE} = O\left( \epsilon^{-1} \log (\epsilon^{-1}) \log \log (\epsilon^{-1}) \right),$$

(68)

where $\log \log (\cdot)$ means $\log (\log (\cdot))$. Combining these with (65), we obtain

$$N_{tot} = O\left( \epsilon^{-1} \log (\epsilon^{-1}) \log \log (\epsilon^{-1}) \right),$$

(69)

which eventually beats LSM’s complexity $\tilde{O}(\epsilon^{-2})$ for small $\epsilon$.

E. Comparison with existing Chebyshev interpolation-based methods

In fact, the idea that we approximate the continuation value by Chebyshev interpolation is not novel. There are some classical methods for Bermudan option pricing based on Chebyshev interpolation [44-50]. However, in addition to whether we use QAE or other classical methods for calculating the nodal continuation values, there are the following differences between the above proposed method and the existing methods.

First, we note that we do not have to use Monte Carlo for calculating the continuation value, and [44-45] actually used other methods. These papers considered the situation where the transition probability of the underlying asset prices can be easily calculated, and computed the continuation value by the numerical integration of the product of the transition probability and the option value at the next exercise date. Note that this way is possible only for some simple models for underlying asset evolution such as the Black-Scholes model. On the other hand, in more complicated settings where, for example, we price a multi-asset option under the stochastic local volatility model, Monte Carlo can be the sole solution, and such a time-consuming situation is a meaningful target for quantum speed-up. However, combining Chebyshev interpolation with various methods might be an interesting possibility also in the quantum setup, and worth to be investigated as a future work.
Let us also mention the differences from [48]. The major difference is that, in the method in [48], the continuation value is not the target of either Monte Carlo integration or Chebyshev interpolation. Instead, the method calculates the conditional expectations of Chebyshev polynomials by Monte Carlo or other methods, and find the Chebyshev interpolation of not the continuation value but the option price at each exercise date. This approach saves the computational time when we price many options under a same model, since we can reuse the conditional expectations for interpolations in pricing different options. Considering the quantum version of this approach might be interesting too.

\[ \sup_{\mathbb{R}^D, \hat{\rho}} \left| \hat{Q}_k(S) \right| \leq \hat{B}_k \]  

holds, where \( \hat{B}_k \) is some positive real number.

**Theorem 4.** Under Assumptions 1, 2 and 4 consider Algorithm 1. Suppose that, for every \( k \in [K - 1] \) and \( j \in \mathcal{J}_k \), (53) is satisfied for some \( \epsilon_{k, \text{QAE}} \in \mathbb{R}_+ \), and that (56) is satisfied for some \( \epsilon_{k, \text{QAE}} \in \mathbb{R}_+ \). Then,

\[ |V_0 - \hat{V}_0| \leq \sum_{k=1}^{K-1} \epsilon_{k, \text{OB}} + \sum_{k=1}^{K-1} \epsilon_{k, \text{int}} + \sum_{k=0}^{K-1} \Lambda_k \epsilon_{k, \text{QAE}} \]

holds, where, for every \( k \in [K - 1] \),

\[ \epsilon_{k, \text{int}} := \epsilon_{\text{int}}(\hat{\rho}_k, d, m_k, \hat{B}_k), \]

and \( \Lambda_k \) is defined as (60).

The proof is presented in Appendix C. Note that a similar point has been made for LSM in [59] (Theorem 3.1).

Of course, \( \hat{Q}_k \) is defined as (51) with \( \hat{V}_{k+1} \), which is the intermediate output in Algorithm 1, and making assumptions on such a thing does not lead to self-contained discussion. It is more desirable to derive the error bound under assumptions on \( \hat{Q}_k \) and/or other quantities determined independently from pricing algorithms. We leave considering whether we can obtain an error bound similar to Theorem 3 under such assumptions or not as a future work.

**G. Quantization of LSM**

Lastly, we make a comment on whether we can consider the quantum algorithm for LSM. Since there are some quantum algorithms for linear regression [65, 71], we naturally wonder that we can apply these to LSM and then obtain speed-up. However, this is not so straightforward, since most of these algorithms output the regression result as a quantum state, in which the regression coefficients are amplitude-encoded. Fortunately, some algorithms [67, 71] output the regression coefficient as classical data. In particular, the algorithm in [71] has the complexity of \( O(D^{1/2}k^3/\epsilon) \) with the tolerance \( \epsilon \), the explanatory variable number \( D \), and the condition number \( k \) of the design matrix, from which we expect the quadratic speed-up of LSM with respect to \( \epsilon \). Nevertheless, applying this algorithm to LSM is not immediate either, because of some points to be considered. For example, the complexity has strong dependence on \( D \) and \( k \), which might make the algorithm disadvantageous. Therefore, it will be crucial to evaluate these, especially \( k \), in addition to finding the basis function set which makes \( k \) as small as possible. We will consider this direction in the future work.
VI. SUMMARY

In this paper, we considered application of quantum algorithms to Bermudan option pricing. Since there are QAE-based algorithms for Monte Carlo integration, which provide quadratic speed-up compared with the classical counterparts, and applications of them to some option pricing problems have been investigated, it is natural to consider to apply them to Bermudan option pricing. One crucial issue in this problem is how to approximate the continuation value $Q_k$, which determines the optimal exercise date. In order to cope with this, we considered combination of QAE and Chebyshev interpolation. That is, the proposed method estimates the values of $Q_k$ on the interpolation nodes by QAE, and, using such estimates, find a Chebyshev interpolation as an approximation of $Q_k$. We presented the calculation procedure in detail, along with the error bound and the complexity, which corresponds to the number of calls to the oracle for underlying asset price evolution, sufficient to achieve the desired error tolerance $\epsilon$. As expected, this method has the complexity depending on $\epsilon$ as $O(\epsilon^{-1})$, which means the quadratic speed-up compared with LSM, the typical classical algorithm for Bermudan option pricing.

As a future work, it is interesting to consider the quantum version of LSM, as mentioned in Section $\nabla \mathcal{G}$. Besides, it is also meaningful to extend the proposed method to other types of dynamic programming, which is ubiquitous in many fields of science and industry.

ACKNOWLEDGMENT

This work was supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant Number JP-MXS0120319794.

Appendix A: Proof of Theorem $\clubsuit$

Proof. First, we note that, for every $k \in [K - 1],$

$$\epsilon_k \leq \tilde{\epsilon}_k$$

(A1)

holds, where $$\epsilon_k := \max_{S \in D_k} \left| V_k(S) - \bar{V}_k(S) \right|,
$$

(A2)

and

$$\tilde{\epsilon}_k := \begin{cases} \epsilon_{k-1}^{\text{int}} + \Lambda_k \epsilon_{k-1}^{\text{QAE}}; & \text{for } k = K - 1 \\ \epsilon_k^{\text{int}} + \Lambda_k \left( \epsilon_{k+1}^{\text{QAE}} + \epsilon_{k+1}^{\text{OB}} + \epsilon_k^{\text{QAE}} \right); & \text{for } k = 1, ..., K - 2 \end{cases}
$$

(A3)

The proof of this is as follows. We see that, for any $k \in [K - 1],$

$$\left| V_k(S_k) - G_k[\bar{V}_k](S_k) \right| = \left| V_k(S_k) - \bar{V}_k(S_k) \right| \leq \epsilon_k
$$

(A4)

holds if $S_k \in \mathcal{D}_k,$ and, under Assumption $\heartsuit$ either

$$\left| V_k(S_k) - G_k[\bar{V}_k](S_k) \right| = \left| V_k(S_k) - V_k^{\text{OB}}(S_k) \right| \leq \epsilon_k^{\text{OB}}
$$

(A5)

or

$$\left| V_k(S_k) - G_k[\bar{V}_k](S_k) \right| = \left| V_k(S_k) - F_k[\bar{V}_k](S_k) \right| \leq \left| V_k(S_k) - F_k[\bar{V}_k](S_k) \right| + \left| F_k[V_k](S_k) - F_k[\bar{V}_k](S_k) \right| = \left| V_k(S_k) - F_k[V_k](S_k) \right| + \left| V_k(b_k(S_k)) - \bar{V}_k(b_k(S_k)) \right| \leq \epsilon_k^{\text{OB}} + \epsilon_k
$$

or

$$\left| V_k(S_k) - G_k[\bar{V}_k](S_k) \right| \leq \epsilon_k^{\text{OB}} + \epsilon_k
$$

(A6)

holds if $S_k \in \mathcal{S} \setminus \mathcal{D}_k$. Combining these, we obtain

$$\left| V_k(S_k) - G_k[\bar{V}_k](S_k) \right| \leq \epsilon_k^{\text{OB}} + \epsilon_k
$$

(A7)

for any $S_k \in S$. This leads to

$$|Q_k(S) - \bar{Q}_k(S)| = \left| \mathbb{E} \left( V_{k+1}(S_{k+1}) - G_{k+1}[\bar{V}_{k+1}](S_{k+1}) \right) \right|
$$

(A8)

for every $k \in [K - 2]$ and $S \in \mathcal{D}_k$. Thus, with $\clubsuit$, we obtain

$$\left| Q_k \left( \frac{v_{\mathcal{D}_j,m}}{j} \right) - \bar{Q}_k \left( \frac{v_{\mathcal{D}_j,m}}{j} \right) \right| \leq \epsilon_k^{\text{OB}} + \epsilon_k^{\text{QAE}}
$$

(A9)

for any $k \in [K - 2]$ and $j \in \mathcal{J}_k$. On the other hand, for $k = K - 1$, noting that $Q_{K-1}(S) = \bar{Q}_{K-1}(S)$ for any $S \in S$ by definition, we see that

$$\left| Q_{K-1} \left( \frac{v_{\mathcal{D}_j,m_k}}{j} \right) - \bar{Q}_{K-1} \left( \frac{v_{\mathcal{D}_j,m_k}}{j} \right) \right| \leq \epsilon_k^{\text{QAE}}
$$

(A10)

for any $j \in \mathcal{J}_{K-1}$. Then, under Assumption $\heartsuit$ invoking Theorem $\clubsuit$, we obtain

$$|Q_k(S_k) - \bar{Q}_k(S_k)| \leq \epsilon_k \quad (A11)
$$

for any $k \in [K - 1]$ and $S \in \mathcal{D}_k$, which immediately leads to (A1) as

$$\left| V_k(S_k) - \bar{V}_k(S_k) \right| = \max \{|f_k^{\text{pay}}(S_k), \bar{Q}_k(S_k)| - \max \{|f_k^{\text{pay}}(S_k), \bar{Q}_k(S_k)| \}
$$

(A12)

Here, we used $\max |a, b| - \max |a, c| \leq |b - c|$, which holds for any $a, b, c \in \mathbb{R}$.

Next, let us note that, for $k \in [K - 2],$

$$\epsilon_k \leq \sum_{k=k}^{K-1} \bar{\Lambda}_{k,k-1} \epsilon_k^{\text{int}} + \sum_{k=k+1}^{K-1} \bar{\Lambda}_{k,k-1} \epsilon_k^{\text{OB}} + \sum_{k=k}^{K-1} \bar{\Lambda}_{k,k} \epsilon_k^{\text{QAE}},
$$

(A13)
holds. We prove this by induction. For \( k = K - 2 \), \((A1)\) implies that

\[
\varepsilon_{K-2} 
\leq \varepsilon_{K-2}^{\text{int}} + \Lambda_{K-2} \left( \varepsilon_{K-1} + \varepsilon_{K-1}^{\text{OB}} + \varepsilon_{K-1}^{\text{QAE}} \right) 
\leq \varepsilon_{K-2}^{\text{int}} + \Lambda_{K-2} \left( \varepsilon_{K-1}^{\text{int}} + \Lambda_{K-1} \varepsilon_{K-1} + \varepsilon_{K-1}^{\text{OB}} + \varepsilon_{K-1}^{\text{QAE}} \right) 
= \sum_{k=K-2}^{K-1} \Lambda_{K-2,k-1} \varepsilon_{k}^{\text{int}} + \sum_{k=K-1}^{K-1} \Lambda_{K-2,k-1} \varepsilon_{k}^{\text{OB}} + \sum_{k=K-1}^{K-1} \Lambda_{K-2,k} \varepsilon_{k}^{\text{QAE}}. 
\]

\[(A14)\]

Similarly, if \((A13)\) hold for \( k \in \{2, \ldots, K-2\} \), \((A1)\) implies that

\[
\varepsilon_{k-1} 
\leq \varepsilon_{k-1}^{\text{int}} + \Lambda_{k-1} \left( \varepsilon_{k} + \varepsilon_{k}^{\text{OB}} + \varepsilon_{k}^{\text{QAE}} \right) 
\leq \varepsilon_{k-1}^{\text{int}} + \Lambda_{k-1} \left( \sum_{k'=k}^{k-1} \Lambda_{k,k'-1} \varepsilon_{k'}^{\text{int}} + \sum_{k'=k}^{k-1} \Lambda_{k,k'-1} \varepsilon_{k'}^{\text{QAE}} + \sum_{k'=k}^{k-1} \Lambda_{k,k} \varepsilon_{k}^{\text{QAE}} \right) 
+ \varepsilon_{k}^{\text{OB}} + \varepsilon_{k} + \varepsilon_{k}^{\text{QAE}} 
= \sum_{k'=k}^{k-1} \Lambda_{k,k'-1} \varepsilon_{k'}^{\text{int}} + \sum_{k'=k}^{k-1} \Lambda_{k,k'-1} \varepsilon_{k'}^{\text{OB}} + \sum_{k'=k}^{k-1} \Lambda_{k,k} \varepsilon_{k}^{\text{QAE}}. 
\]

\[(A15)\]

Therefore, \((A13)\) is proved for every \( k \in \{K-1\} \).

Finally, the claim is proved as follows. We see that

\[
|V_0 - \bar{V}_0| 
\leq |\mathbb{E}[V_1(S_1)] - \mathbb{E}[G_1(V_1)[\mathcal{S}_1]]| 
\leq \mathbb{E} \left| |V_1(S_1)| - G_1(V_1)[S_1]| \right| 
\leq \mathbb{E} \left| \Lambda_{1,1} \varepsilon_{1}^{\text{int}} + \mathbb{E} \left( \Lambda_{1,k-1} \varepsilon_{k}^{\text{QAE}} \right) \right| 
\leq \sum_{k=1}^{K-1} \Lambda_{1,k} \varepsilon_{k}^{\text{int}} + \sum_{k=1}^{K-1} \Lambda_{1,k} \varepsilon_{k}^{\text{QAE}}. 
\]

(A16)

where we used \((A7)\) and \((A13)\) at the second and last inequalities, respectively. Combining this and \((56)\), we obtain

\[
\left| V_0 - \bar{V}_0 \right| 
\leq \left| V_0 - \bar{V}_0 \right| + \left| \bar{V}_0 - \bar{V}_0 \right| 
= \sum_{k=1}^{K-1} \Lambda_{1,k} \varepsilon_{k}^{\text{int}} + \sum_{k=1}^{K-1} \Lambda_{1,k} \varepsilon_{k}^{\text{QAE}}. 
\]

(A17)

\[\square\]

Appendix B: Proof of Corollary [1]

**Proof.** By simple algebra, we see that \( m_1, \ldots, m_{K-1} \) satisfying \((61)\) lead to

\[
\Lambda_{1,k-1} \varepsilon_{k}^{\text{int}} \leq \frac{\varepsilon_{\text{max}}}{2(K-1)}. 
\]

(B1)

for every \( k \in \{K-1\} \).

On the other hand, Theorem [2] implies that, for every \( k \in \{K-1\} \) and \( j \in \mathcal{J}_k \), using \( N_{\text{QAE}}^{\text{QAE}} \) set as \((62)\). Step 4 in Algorithm [1] gives us \( \bar{P}_{k,j} \) an estimate on \( P_{k,j} \) in \((50)\), satisfying

\[
\left| P_{k,j} - \bar{P}_{k,j} \right| \leq \tilde{e}_k, 
\]

(B2)

and then \( \bar{Q}_{\text{QAE}}^{\text{QAE}} \) satisfying

\[
\left| \bar{Q}_{\text{QAE}}(S_{\text{D},m_j}) - \bar{Q}_{\text{QAE}}^{\text{QAE}} \right| 
= \left| (2P_{k,j} - 1) \bar{V}_{\text{max}} - (2 \bar{P}_{k,j} - 1) \bar{V}_{\text{max}} \right| 
\leq 2 \bar{V}_{\text{max}} \tilde{e}_k 
\leq \frac{\sqrt{(m_k + 1)^{d} \Lambda_{1,k}}} {1 + \sum_{k'=1}^{K-1} (m_{k'} + 1)^{d} \Lambda_{1,k'}} \frac{\varepsilon_{\text{max}}}{2} =: \varepsilon_{\text{QAE}}, \]

(B3)

with some probability. Similarly, it is implied that, with \( N_{\text{QAE}}^{\text{QAE}} \) set as \((62)\). Step 9 gives us \( \bar{P}_0 \), an estimation on \( P_0 \), satisfying

\[
\left| P_0 - \bar{P}_0 \right| \leq \tilde{e}_0, 
\]

(B4)

and then \( \bar{V}_0 \) satisfying

\[
\left| \bar{V}_0 - \bar{V}_0 \right| 
= \left| (2P_0 - 1) \bar{V}_{\text{max}} - (2 \bar{P}_0 - 1) \bar{V}_{\text{max}} \right| 
\leq 2 \bar{V}_{\text{max}} \tilde{e}_0 
\leq \frac{1}{1 + \sum_{k'=1}^{K-1} (m_{k'} + 1)^{d} \Lambda_{1,k'}} \frac{\varepsilon_{\text{max}}}{2} =: \varepsilon_{\text{QAE}}. \]

(B5)

with some probability. Therefore, when all of these succeed,\n
\[
\left| V_0 - \bar{V}_0 \right| 
\leq \sum_{k=1}^{K-1} \Lambda_{1,k} \varepsilon_{k}^{\text{int}} + \sum_{k=1}^{K-1} \Lambda_{1,k} \varepsilon_{k}^{\text{QAE}} \leq \sum_{k=1}^{K-1} \Lambda_{1,k} \varepsilon_{k}^{\text{QAE}} \leq \frac{\varepsilon_{\text{max}}}{2} + \epsilon_{\text{OB}}^{\text{OB}} \leq \varepsilon_{\text{QAE}}^{\text{QAE}}. \]

(B6)

holds, according to Theorem [3]. Here, we used \((B1), (B3), (B5)\), and the assumption that \( \epsilon_{1}^{\text{OB}}, \ldots, \epsilon_{K-1}^{\text{OB}} \) are 0, along with simple algebra.

The remaining task is proving that the probability \( P_{\text{all}} \) that all of the estimations in Steps 4 and 9 succeed is larger than 0.99 under the setting of \( N_{\text{QAE}}^{\text{QAE}} \) as \((64)\). Note that, according to Theorem [2] with the setting as \((64)\), the probability that Step 4 for a set of \( k \in \{K-1\} \) and \( j \in \mathcal{J}_k \) outputs \( \bar{P}_{k,j} \) satisfying \((B2)\) is higher than

\[
1 - \frac{0.01}{N_{\text{est}}} \]

(B7)

Similarly, the probability that Step 9 outputs \( \bar{P}_0 \) satisfying \((B4)\) is also higher than \((B7)\). Besides, the total number of
these estimations is $N_{est}$. Combining these, we obtain a lower bound of $P_{all}$ as

$$P_{all} \geq \left(1 - \frac{0.01}{N_{est}}\right)^{N_{est}} \geq 1 - 0.01 = 0.99,$$

(B8)

which completes the proof. \hspace{1cm} \Box

Appendix C: Proof of Theorem 4

Proof. Because of (55), Theorem 1 implies that

$$|\tilde{Q}_k(S') - \tilde{Q}_k(S)| \leq \epsilon_k^\text{int} + \Lambda_k \epsilon_k^\text{QAE}$$

(C1)

for every $k \in [K-1]$ and $S' \in D_k$. Besides, for every $k \in [K-2]$ and $S' \in D_k$, $|Q_k(S') - \tilde{Q}_k(S')| \leq \epsilon_{k+1} + \epsilon_k^\text{QAE}$ holds as (A8), where $\epsilon_k$ is defined as (A2) for every $k \in [K-1]$, whereas $Q_{K-1}(S') = \tilde{Q}_{K-1}(S')$ by definition of $\mathbf{G}_k[\cdot]$. Combining these, we see that, for every $k \in [K-1]$ and $S' \in D_k$,

$$|Q_k(S') - \tilde{Q}_k(S')| \leq |Q_k(S') - \tilde{Q}_k(S')| + |\tilde{Q}_k(S') - \tilde{Q}_k(S)|

\leq \epsilon_{k+1} + \epsilon_k^\text{QAE}$$

(C2)

holds with $\epsilon_k = 0$ and $\epsilon_k^\text{QAE} = 0$, which leads to

$$|V_k(S') - \tilde{V}_k(S')| \leq \epsilon_{k+1} + \epsilon_k^\text{QAE}$$

(C3)

similarly to (A12). Therefore, for every $k \in [K-1],$

$$\epsilon_k = \max_{S' \in D_k} |V_k(S') - \tilde{V}_k(S')| \leq \epsilon_{k+1} + \epsilon_k^\text{QAE} + \epsilon_k^\text{int} + \Lambda_k \epsilon_k^\text{QAE}.$$  

(C4)

This implies

$$\epsilon_1 \leq \sum_{k=2}^{K-1} \epsilon_k^\text{QAE} + \sum_{k=1}^{K-1} \epsilon_k^\text{int} + \sum_{k=0}^{K-1} \Lambda_k \epsilon_k^\text{QAE}.$$  

(C5)

Finally, combining (C5) with (56) and $|V_0 - \tilde{V}_0| \leq \epsilon_1$, which we can see as (A16), we obtain

$$|V_0 - \tilde{V}_0| \leq |V_0 - \tilde{V}_0| + |\tilde{V}_0 - \tilde{V}_0| \leq \epsilon_1 + \epsilon_1^\text{QAE} + \epsilon_1^\text{QAE}$$

(C6)

\hspace{1cm} \Box

[1] M. A. Nielsen and I. L. Chuang, “Quantum Computation and Quantum Information”, Cambridge University Press (2010)
[2] R. Orus et al., “Quantum computing for finance: overview and prospects”, Reviews in Physics 4, 100028 (2019)
[3] D. J. Egger et al., “Quantum computing for finance: state of the art and future prospects”, IEEE Transactions on Quantum Engineering, 1, 3101724 (2020)
[4] A. Bouland et al., “Prospects and challenges of quantum finance”, arXiv:2011.06492
[5] J. C. Hull, “Options, Futures, and Other Derivatives”, Prentice Hall (2012)
[6] S. Shreve, “Stochastic Calculus for Finance I & II”, Springer (2004)
[7] P. Rebentrost et al., “Quantum computational finance: Monte Carlo pricing of financial derivatives”, Phys. Rev. A 98, 022321 (2018)
[8] A. Martin et al., “Towards Pricing Financial Derivatives with an IBM Quantum Computer”, Phys. Rev. Research 3, 013167 (2021)
[9] N. Stamatopoulos et al., “Option Pricing using Quantum Computers”, Quantum 4, 291 (2020)
[10] S. Ramos-Calderer et al., “Quantum unary approach to option pricing”, Phys. Rev. A 103, 032414 (2021)
[11] F. Fontanela et al., “A Quantum algorithm for linear PDEs arising in Finance”, arXiv:1912.02753
[12] A. C. Vazquez and S. Woerner, “Efficient state preparation for quantum amplitude estimation”, Phys. Rev. Applied 15, 034027 (2021)
[13] K. Kaneko et al., “Quantum pricing with a smile: Implementation of local volatility model on quantum computer”, arXiv:2007.01467
[14] H. Tang et al., “Quantum Computation for Pricing the Collateralized Debt Obligations”, arXiv:2008.04110
[15] S. Chakraborti et al., “A Threshold for Quantum Advantage in Derivative Pricing”, Quantum 5, 463 (2021)
[16] D. An et al., “Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance”, Quantum 5, 481 (2021)
[17] J. Gonzalez-Conde et al., “Pricing Financial Derivatives with Exponential Quantum Speedup”, arXiv:2101.04023
[18] S. K. Radha, “Quantum option pricing using Wick rotated imaginary time evolution”, arXiv:2101.04320
[19] S. Woerner and D. J Egger, “Quantum risk analysis”, npj Quantum Information, 5(1), 1 (2019)
[20] D. J Egger et al., “Credit risk analysis using quantum computers”, IEEE Transactions on Computers (2020)
[21] 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 102, 022424 (2020)
[22] K. Kaneko et al., “Quantum speedup of Monte Carlo integration with respect to the number of dimensions and its application to finance”, Quantum Information Processing, 20, 185 (2021)
[23] P. Rebentrost and S. Lloyd, “Quantum computational finance: quantum algorithm for portfolio optimization”, arXiv:1811.03975
