RESEARCH **Open Access**

Bermudan option pricing by quantum amplitude estimation and Chebyshev interpolation

Koichi Miyamoto[1]* ⓘ

*Correspondence:
koichi.miyamoto@qiqb.osaka-u.ac.jp
[1]Center for Quantum Information
and Quantum Biology, Osaka
University, 1-3 Machikaneyama,
Toyonaka, Osaka, Japan

Abstract

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 $\widetilde{O}(\epsilon^{-1})$, where ϵ 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 $\widetilde{O}(\epsilon^{-2})$.

Keywords: Option pricing; Quantum amplitude estimation; Chebyshev interpolation

1 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 [7–18], risk measurement [19–22], portfolio optimization [23–25], 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

---

[1]As a standard textbook for quantum computing, we refer to [1].

[2]As standard textbooks for financial engineering, especially option pricing, we refer to [5, 6].

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,[3] 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 $\widetilde{O}(N^{-1/2})$[4] when the sample number N increases, regardless of the number of the underlying assets d. In other words, it suffices to take $\widetilde{O}(\epsilon^{-2})$ samples in order to achieve the error tolerance ϵ on the option price. This contrasts to other methods, for example approaches based on solving partial differential equations [27, 28], whose complexity is $\widetilde{O}((1/\epsilon)^{\mathrm{poly}(d)})$. 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 forgoes 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) [29] 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 continuous time period with that at discrete dates with sufficiently small intervals.

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

---

[3] We refer to [26] as a textbook on Monte Carlo simulation and its application to finance.

[4] $\widetilde{O}(\cdot)$ hides logarithmic factors in the ordinary big O notation $O(\cdot)$.

based on quantum amplitude estimation (QAE) [31, 33–42]. 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,[5] and has already been used in some (classical) methods for Bermudan option pricing [44–50]. 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 by 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 ϵ, calling the oracle only $\widetilde{O}(\epsilon^{-1})$ 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 Sect. 2, 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 Sect. 3, we present the general formulation of Bermudan option pricing and explain LSM as a typical classical solutions. In Sect. 4, we explain QAE and QAE-based Monte Carlo integration. Then, in Sect. 5, 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. In Sect. 6, we make some remarks: comparison with existing methods, comments on exponential factors with respect to the exercise date number in the error bound, and the possibility of quantization of LSM. Section 7 summarizes this paper. All proofs are presented in the Appendix.

1.1 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 $[n] := \{i \in \mathbb{N} \mid i \le n\}$ for any $n \in \mathbb{N}$, and $[n]_0 := \{i \in \mathbb{N}_0 \mid i \le n\}$ for any $n \in \mathbb{N}_0$. We also define $\mathbb{N}_{\ge n} := \{i \in \mathbb{N} \mid i \ge n\}$ for $n \in \mathbb{N}$. Similarly, we define $\mathbb{R}_{>a} := \{x \in \mathbb{R} \mid x > a\}$ and $\mathbb{R}_{\ge a} := \{x \in \mathbb{R} \mid x \ge a\}$ 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 $\vec{l}_1, \vec{l}_2 \in \mathbb{N}_0^d$, we also define $\delta_{\vec{l}_1, \vec{l}_2}$, which is 1 if $\vec{l}_1 = \vec{l}_2$ and 0 otherwise.

For a measure space $(\Omega, \mathcal{F}, \mu)$ and $p \in \mathbb{R}_{\ge 1}$, $L^p(\Omega, \mu)$ 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 $\vec{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$, $|\vec{x}\rangle := |x_1\rangle \cdots |x_d\rangle$ is the state on the d-register system.

2 Approximation of functions by Chebyshev interpolation
For $l \in \mathbb{N}_0$, the l-th Chebyshev polynomial (of the first kind) is defined as

$$T_l(x) := \cos\big(l \cdot \arccos(x)\big), \tag{1}$$

---

[5]See [43] as a textbook on this topic.

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

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

Here, $x_{m,j}$ is the *Chebyshev node* defined as

$$x_{m,j} = \cos\left(\frac{j + \frac{1}{2}}{m+1}\pi\right) \tag{3}$$

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

$$\mathcal{D} = [L_1, U_1] \times \cdots \times [L_d, U_d], \tag{4}$$

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

$$\widetilde{T}_{\mathcal{D}, \vec{l}}(\vec{S}) := \prod_{i=1}^{d} T_{l_i}\big(\theta_{\mathcal{D},i}(S_i)\big) \tag{5}$$

for every $\vec{l} = (l_1, \ldots, l_d)^T \in \mathbb{N}_0^d$ and $\vec{S} = (S_1, \ldots, S_d)^T \in \mathcal{D}$, where

$$\theta_{\mathcal{D}}(\vec{S}) := \left(\frac{2S_1 - U_1 - L_1}{U_1 - L_1}, \ldots, \frac{2S_d - U_d - L_d}{U_d - L_d}\right)^T. \tag{6}$$

For the above polynomials, the orthogonality relation is now

$$\sum_{\vec{x} \in \mathcal{G}_{\mathcal{D}}^{d,m}} \widetilde{T}_{\mathcal{D}, \vec{l}_1}(\vec{s}) \widetilde{T}_{\mathcal{D}, \vec{l}_2}(\vec{s}) = \begin{cases} \frac{(m+1)^d}{2^{\aleph(\vec{l}_1)}}; & \text{if } \vec{l}_1 = \vec{l}_2, \\ 0; & \text{if } \vec{l}_1 \neq \vec{l}_2 \end{cases} \tag{7}$$

for every $m \in \mathbb{N}$ and $\vec{l}_1, \vec{l}_2 \in [m]_0^d$, where $\aleph(\vec{l}) := \#\{i \in [d] \mid l_i > 0\}$ for $\vec{l} = (l_1, \ldots, l_d)^T \in \mathbb{N}_0^d$, and $\mathcal{G}_{\mathcal{D}}^{d,m}$ is the set of points $\vec{S}_{\vec{j}}^{\mathcal{D},m} \in \mathcal{D}$ written in the form of

$$\vec{S}_{\vec{j}}^{\mathcal{D},m} := \left(\frac{U_1 - L_1}{2}x_{m,j_1} + \frac{U_1 + L_1}{2}, \ldots, \frac{U_d - L_d}{2}x_{m,j_d} + \frac{U_d + L_d}{2}\right)^T \tag{8}$$

with $\vec{j} = (j_1, \ldots, j_d)^T \in [m]_0^d$.

We can use the above polynomials for function approximation. Given $\mathcal{D}$ as (4) and $m \in \mathbb{N}$, we define the *Chebyshev interpolation* of a function $f : \mathcal{D} \to \mathbb{R}$ as

$$\Pi_{\mathcal{D},m}[f](\vec{S}) := \sum_{\vec{l} \in [m]_0^d} a_{f, \vec{l}} \widetilde{T}_{\mathcal{D}, \vec{l}}(\vec{S}) \tag{9}$$

for every $\vec{S} \in \mathcal{D}$, where the coefficient $a_{f,\vec{l}}$ is calculated by

$$a_{f,\vec{l}} := \frac{2^{\aleph(\vec{l})}}{(m+1)^d} \sum_{\vec{S} \in \mathcal{G}_{\mathcal{D}}^{d,m}} f(\vec{S}) \widetilde{T}_{\mathcal{D},\vec{l}}(\vec{S}) \tag{10}$$

for every $\vec{l} \in [m]_0^d$. This is in fact an interpolation, since $\Pi_{\mathcal{D},m}[f](\vec{S}) = f(\vec{S})$ for every node $\vec{S} \in \mathcal{G}_{\mathcal{D}}^{d,m}$.

The error in the above approximation has been investigated in [47, 51]. They gave the error bound, making an assumption on analyticity of the interpolated function f. We here present the theorem on such a error in the case where we are given the values of f at the Chebyshev nodes with some errors [47]. However, let us make some definitions prior to the theorem. For $\rho \in \mathbb{R}_{>1}$, the Bernstein ellipse $\mathcal{B}_\rho$ is defined as the open region in the complex plane bounded by the ellipse $\{\frac{1}{2}(u + u^{-1}) \mid u \in \mathbb{C}, |u| = \rho\}$. We also define the generalized Bernstein ellipse as $\mathcal{B}_{\mathcal{D},\rho} := (\eta_1 \circ \mathcal{B}_\rho) \times \cdots \times (\eta_d \circ \mathcal{B}_\rho)$, where, for every $i \in [d]$, $\eta_i(z)$ is the map from $\mathbb{C}$ to $\mathbb{C}$ defined as $\eta_i(z) := \frac{U_i - L_i}{2} z + \frac{U_i + L_i}{2}$. Furthermore, we define the multivariate version of the *Lebesgue constant* of the Chebyshev nodes: for every $m \in \mathbb{N}$,

$$\Lambda_{d,m} := \max_{(x_1,\ldots,x_d)^T \in [-1,1]^d} \sum_{(j_1,\ldots,j_d)^T \in [m]_0^d} \prod_{i=1}^d \ell_{j_i}^m(x_i), \tag{11}$$

where

$$\ell_j^m(x) := \prod_{k \in [m]_0 \setminus \{j\}} \frac{x - x_{m,k}}{x_{m,j} - x_{m,k}} \tag{12}$$

for every $j \in [m]_0$. As [47] showed,

$$\Lambda_{d,m} \le \prod_{i=1}^d \left(\frac{2}{\pi} \log(m+1) + 1 \right) \tag{13}$$

holds, which is derived from the well-known upper bound $\Lambda_{1,m} \le \frac{2}{\pi} \log(m+1) + 1$ [43]. Then, the theorem is as follows.[6]

Theorem 1 *Let d and m be positive integers. Let $\mathcal{D}$ be a hyper-rectangle like (4). Let $f : \mathcal{D} \to \mathbb{R}$ be a function that has an analytic extension to $\mathcal{B}_{\mathcal{D},\rho}$ for some $\rho \in \mathbb{R}_{>1}$. Besides, assume that $\sup_{\vec{s} \in \mathcal{B}_{\mathcal{D},\rho}} |f(\vec{s})| \le B$ for some $B \in \mathbb{R}$. Moreover, suppose that we are given a real number $\hat{f}_{\vec{j}}$ for every $\vec{j} \in [m]_0^d$, and that there exists $\epsilon \in \mathbb{R}$ such that*

$$\left| f(\vec{S}_{\vec{j}}^{\mathcal{D},m}) - \hat{f}_{\vec{j}} \right| \le \epsilon \tag{14}$$

holds for every $\vec{j} \in [m]_0^d$. Then,

$$\max_{\vec{S} \in \mathcal{D}} \left| f(\vec{S}) - \tilde{f}(\vec{S}) \right| \le \epsilon^{\mathrm{int}}(\rho, d, m, B) + \Lambda_{d,m} \epsilon \tag{15}$$

---

[6][47–50] considered the more general case, where the values of ρ and m are different for different directions in $\mathbb{R}^d$. In this paper, we take common values of ρ and m for every direction, for simplicity.

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

$$\tilde{f}(\vec{S}) := \sum_{\vec{l} \in [m]_0^d} \tilde{a}_{\vec{l}} \widetilde{T}_{\mathcal{D}, \vec{l}}(\vec{S}), \tag{16}$$

with the coefficients $\tilde{a}_{\vec{l}}$ calculated by

$$\tilde{a}_{\vec{l}} := \frac{2^{\aleph(\vec{l})}}{(m+1)^d} \sum_{\vec{j} \in [m]_0^d} \hat{f}_{\vec{j}} \widetilde{T}_{\mathcal{D}, \vec{l}}(\vec{S}_{\vec{j}}^{\mathcal{D}, m}) \tag{17}$$

for every $\vec{l} \in [m]_0^d$, and

$$\epsilon^{\text{int}}(\rho, d, m, B) := 2^{\frac{d}{2}+1} \sqrt{d} B \rho^{-m} (1 - \rho^{-2})^{-\frac{d}{2}}. \tag{18}$$

3 Bermudan option pricing

3.1 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 $t_0 < t_1 < \cdots < t_K$ with $t_0 := 0$ being the present and $t_K := T \in \mathbb{R}_+$ being the final maturity. This is formulated as follows. Under some filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \leq 0}, \mathbb{P})$, consider the $\mathcal{S}$-valued Markov process $\vec{S}(t) := (S_1(t), \ldots, S_d(t))^T$, where $\mathcal{S}$ is a subset of $\mathbb{R}^d$ equipped with its Borel σ-algebra inherited from $\mathbb{R}^d$, and $\vec{S}_0 := \vec{S}(0)$ is deterministic. $\vec{S}(t)$ corresponds to the values of the underlying asset prices at time t, or transformations of them by some function (for example, the logarithms of the asset prices). We are mainly interested in its values at $t_1, \ldots, t_K$, that is, the discrete-time process $\vec{S}_k = (S_{1,k}, \ldots, S_{d,k}) := \vec{S}(t_k)$, $k \in [K]_0$. We hereafter denote an instance of this process, which is a $(K+1)$-tuple of elements of $\mathcal{S}$, as $\mathbf{S} = (\vec{S}_0, \vec{S}_1, \ldots, \vec{S}_K)$. Besides, suppose that we are given the function $f_k^{\text{pay}} \in L^2(\mathcal{S}, \rho_k)$ for every $k \in [K]$, where ρ_k is the image probability measure on $\mathcal{S}$ induced by $\vec{S}_k$. This corresponds to the payoff which arises by the exercise at t_k. Although we assume that the risk-free rate is 0 for simplicity in this paper, we can consider that f_k^{pay} is the discounted payoff, that is, the product of the payoff and the discount factor. Then, the price of the Bermudan option at t_k with $\vec{S}_k = \vec{s} \in \mathcal{S}$ is given as

$$V_k(\vec{s}) := \sup_{\tau \in \mathcal{T}_k} \mathbb{E}\big[f_\tau^{\text{pay}}(\vec{S}_\tau) \mid \vec{S}_t = \vec{s} \big], \tag{19}$$

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 $\{k, \ldots, K\}$-valued stopping times. In particular, the present option price is

$$V_0(\vec{s}) := \sup_{\tau \in \mathcal{T}} \mathbb{E}\big[f_\tau^{\text{pay}}(\vec{S}_\tau) \big], \tag{20}$$

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

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

$$V_k(\vec{s}) = \begin{cases} f_K^{\text{pay}}(\vec{s}); & k = K, \\ \max\{f_k^{\text{pay}}(\vec{s}), Q_k(\vec{s})\}; & k = 1, \ldots, K-1 \end{cases} \tag{21}$$

for every $\vec{s} \in \mathcal{S}$, and

$$V_0 = \mathbb{E}\big[V_1(\vec{S}_1)\big]. \tag{22}$$

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

$$Q_k(\vec{s}) := \mathbb{E}\big[V_{k+1}(\vec{S}_{k+1}) \mid \vec{S}_k = \vec{s}\big] \tag{23}$$

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 $\vec{S}_k = \vec{s}$.

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

$$\begin{aligned} \tau_K &= K, \\ \tau_k &= k\mathbb{1}_{f_k^{\mathrm{pay}}(\vec{S}_k) \geq Q_k(\vec{S}_k)} + \tau_{k+1}\mathbb{1}_{f_k^{\mathrm{pay}}(\vec{S}_k) < Q_k(\vec{S}_k)}, \quad k \in [K - 1] \end{aligned} \tag{24}$$

and $\tau_{\mathrm{op}} = \tau_1$. Also note that

$$Q_k(\vec{s}) = \mathbb{E}\big[f_{\tau_{k+1}}^{\mathrm{pay}}(\vec{S}_{\tau_{k+1}}) \mid \vec{S}_k = \vec{s}\big], \tag{25}$$

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

3.2 Least squares Monte Carlo

We here explain LSM [29], as an algorithm for Bermudan option pricing. This is one of the widely used methods in practical business, and the theoretical error bound on the price in the method has been investigated [52–61].

Omitting some technical details, we describe the outline of LSM as follows. As a preparation, for every $k \in [K - 1]$, we determine the set of functions $\mathcal{H}_k \subseteq L^2(\mathcal{S}, \rho_k)$ for approximation of the continuation value Q_k. One common choice is $\mathcal{H}_k = \mathcal{R}_{d,m}$, the set of all real-coefficient polynomials on $\mathbb{R}^d$ of degree at most $m \in \mathbb{N}$, and we hereafter consider this. Next, we generate $N_{\mathrm{samp}} \in \mathbb{N}_{\geq 2}$ sample paths of underlying asset prices, which are denoted as $\mathbf{S}_i = (\vec{S}_0, \vec{S}_1^{(i)}, \ldots, \vec{S}_K^{(i)})$, $i \in [N_{\mathrm{samp}}]$. Then, we determine the stopping time, which approximate the optimal one τ_{op}, by the following procedure. First, we set $\widehat{\tau}_{K,i} = K$ for every $i \in [N_{\mathrm{samp}}]$. For $k \in [K - 1]$, given $\widehat{\tau}_{k+1,i}$ for every $i \in [N_{\mathrm{samp}}]$, we determine the approximate continuation value $\widehat{Q}_k$ as the element g_k in $\mathcal{R}_{d,m}$, which minimizes

$$\frac{1}{N_{\mathrm{samp}}} \sum_{i=1}^{N_{\mathrm{samp}}} \big(g_k(\vec{S}_k^{(i)}) - f_{\widehat{\tau}_{k+1,i}}^{\mathrm{pay}}(\vec{S}_{\widehat{\tau}_{k+1,i}}^{(i)})\big)^2, \tag{26}$$

or, in other words, best fits to the realized payoffs under the stopping time $\widehat{\tau}_{k+1,i}$ on the sample paths. It is guaranteed by statistical leaning 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

$$\widehat{\tau}_{k,i} = \begin{cases} k; & \text{if } \widehat{Q}_k(\vec{S}_k^{(i)}) \leq f_k^{\mathrm{pay}}(\vec{S}_k^{(i)}), \\ \widehat{\tau}_{k+1,i}; & \text{otherwise} \end{cases} \tag{27}$$

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

$$\widehat{V}_0 := \frac{1}{N_{\text{samp}}} \sum_{i=1}^{N_{\text{samp}}} f_{\widehat{\tau}_{1,i}}^{\text{pay}} \big(\vec{S}_{\widehat{\tau}_{1,i}}^{(i)} \big) \tag{28}$$

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 $\mathbf{S}_i$. In the usual situation where the stochastic differential equation (SDE) for $\vec{S}(t)$ 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

$$\mathbb{E}_{\text{samp}}\big[|\widehat{V}_0 - V_0|\big] = \widetilde{O}\big((N_{\text{samp}})^{-\frac{n(p-2)}{2n(p+2)+d(p-2)}}\big). \tag{29}$$

Here, $\mathbb{E}_{\text{samp}}[\cdot]$ denotes the expectation with respect to randomness of samples, and n and p are the quantities which characterize smoothness of $Q_1(\vec{S}), \ldots, Q_{K-1}(\vec{S})$ and boundedness of the norms of $f_1^{\text{pay}}(\vec{S}), \ldots, f_K^{\text{pay}}(\vec{S})$, respectively (see [61] for more details). For larger p and n, the RHS of (29) decreases faster against the increase of N_{samp}. In the limit of $n, p \to \infty$, which means that Q_k's are highly smooth and the norms of f_k^{pay}'s are well-bounded, the RHS of (29) becomes $\widetilde{O}(N_{\text{samp}}^{-1/2})$, which coincides with the well-known error decay rate in Monte Carlo integration. Conversely, in this limit, it is sufficient to take $\widetilde{O}(\epsilon^{-2})$ samples in order to achieve the error tolerance ϵ.

4 Quantum amplitude estimation and quantum algorithm for Monte Carlo integration

4.1 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

$$A|0\rangle|0\rangle = \sqrt{a}|\psi_1\rangle|1\rangle + \sqrt{1-a}|\psi_0\rangle|0\rangle =: |\Psi\rangle \tag{30}$$

with some $a \in (0, 1)$. Here, the first and second kets 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 ϵ. There exist some algorithms which output such an estimation with $O(\epsilon^{-1})$ calls to A and its inverse $A^\dagger$ in total [31, 33–36]. 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 [30, 62]. Let us define the $(N_{\text{QAE}}, N_{\text{rep}})$-QAE as the method for estimating a which runs N_{rep} rounds of QAE and makes N_{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 $N_{\mathrm{QAE}}N_{\mathrm{rep}}$ times in total. Then, combining Theorem 12 in [33] and Lemma 6.1 in [62], we obtain the following theorem.

Theorem 2 *Suppose that we are given the accesses to A in* (30) *and its inverse* $A^\dagger$. *Then, for any* $\gamma \in (0,1)$ *and* $\epsilon \in (0,0.1)$, *a* $(N_{\mathrm{QAE}}, N_{\mathrm{rep}})$-*QAE, where the positive integers* N_{QAE} *and* N_{rep} *satisfy*

$$N_{\mathrm{QAE}} \geq \frac{7}{\epsilon} \tag{31}$$

and

$$N_{\mathrm{rep}} \geq 12\lceil \log(\gamma^{-1}) \rceil + 1 \tag{32}$$

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

$$|\tilde{a} - a| \leq \frac{7}{N_{\mathrm{QAE}}} \leq \epsilon, \tag{33}$$

where a is defined as (30), *with probability higher than* $1 - \gamma$.

Here, (31) is derived from the inequality in Theorem 12 in [33] with $k = 1$, that is,

$$|\tilde{a} - a| \leq \frac{2\pi\sqrt{a(1-a)}}{M} + \frac{\pi^2}{M^2}, \tag{34}$$

where $M = N_{\mathrm{QAE}}/2$ under the current definition. Using $\sqrt{a(1-a)} \leq \frac{1}{2}$, we see that (34) implies $|\tilde{a} - a| \leq 7/N_{\mathrm{QAE}}$ for $N_{\mathrm{QAE}} \geq 70$, which follows from (31) and $0 < \epsilon < 0.1$. In summary, if N_{QAE} satisfies (31) for $\epsilon \in (0,0.1)$, the error in QAE is suppressed to at most ϵ with high probability. Hereafter, we say that a $(N_{\mathrm{QAE}}, N_{\mathrm{rep}})$-QAE succeeded if it output $\tilde{a}$ such that $|\tilde{a} - a| \leq 7/N_{\mathrm{QAE}}$.

4.2 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 $\mathbb{E}[F(\vec{X})]$, the expected value of $F(\vec{X})$, where $\vec{X}$ is some real vector-valued stochastic variable and F is a real-valued function acting on $\vec{X}$. 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 $O_{\vec{X}}$ and O_F. $O_{\vec{X}}$ is the oracle to generate the state in which the distribution of $\vec{X}$ is encoded. That is, $O_{\vec{X}}$ operates on a quantum register and transform the state with all qubits set to $|0\rangle$ into

$$O_{\vec{X}}|0\rangle = \sum_{i=1}^{N_{\vec{X}}} \sqrt{p_i}|\vec{x}_i\rangle, \tag{35}$$

where $\vec{x}_1, \ldots, \vec{x}_{N_{\vec{X}}}$ are $N_{\vec{X}} \in \mathbb{N}$ possible values of $\vec{X}$ and $p_i, i \in [N_{\vec{X}}]$ is the probability that $\vec{X} = \vec{x}_i$. Here, we assume that the set of all the values that $\vec{X}$ can take is finite. If $\vec{X}$ is continuous, we need some discretization. How to create states corresponding to widely used

distributions such as normal distribution has been investigated [13, 63]. The second oracle O_F operates on a two-register system, and, using the first register as the input $\vec{x}$, outputs $F(\vec{x})$ into the second register. That is, for any $\vec{x}$ in the domain of F,

$$O_F|\vec{x}\rangle|0\rangle = |\vec{x}\rangle|F(\vec{x})\rangle. \tag{36}$$

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

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

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

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

$$\rightarrow \sum_{i=1}^{N_{\vec{X}}} \sqrt{p_i}|\vec{x}_i\rangle|F(\vec{x}_i)\rangle\left(\sqrt{F(\vec{x}_i)}|1\rangle + \sqrt{1 - F(\vec{x}_i)}|0\rangle\right), \tag{37}$$

where the first, second and third kets correspond to R_1, R_2 and R_3, respectively. We use $O_{\vec{X}}$ and O_F at the first and second arrows, respectively. The transformation at the third arrow is done by arithmetic circuits [64] and controlled rotation gates. Note that the probability to obtain 1 in R_3 when we measure the final state in (37) is $\sum_{i=1}^{N_{\vec{X}}} p_i F(\vec{x}_i)$, that is, $\mathbb{E}[F(\vec{X})]$. Therefore, using the whole operation in (37) as the oracle A, we can estimate $\mathbb{E}[F(\vec{X})]$ by QAE.

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

5.1 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 $\vec{S}_{k+1}$ conditional on $\vec{S}_k$. That is, for every $k \in [K - 1]_0$ and $\vec{S} \in \mathcal{S}$,

$$O_{\text{step},k} : |\vec{S}\rangle|0\rangle \mapsto \sum_{\vec{s} \in \widetilde{\mathcal{S}}_{k+1}(\vec{S})} \sqrt{p_{k+1}(\vec{s}; \vec{S})}|\vec{S}\rangle|\vec{s}\rangle, \tag{38}$$

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

$$p_{k+1}(\vec{s}; \vec{S}) := \mathbb{P}(\vec{S}_{k+1} = \vec{s} \mid \vec{S}_k = \vec{S}). \tag{39}$$

We here make comments on how to implement $O_{\text{step},k}$. As mentioned in Sect. 3, usually, following some SDE and some numerical method such as Euler-Maruyama, we can

generate random sample values of $\vec{S}_{k+1}$ with the given value of $\vec{S}_k$ as the initial condition. Implementations of such a calculation on quantum circuits have been discussed in the previous papers [7, 9, 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 $\vec{S}(t)$ from t_k to t_{k+1}. This yields the state like (38). We should also note that, in Assumption 1, it is assumed that $\vec{S}_{k+1}$ can take only a finite number of values for the fixed $\vec{S}_k$. This is not the case in the most models of $\vec{S}(t)$, in which it takes continuous values. However, under the aforementioned implementations for time evolution of $\vec{S}(t)$, in which both time and random variables are discretely approximated, the number of the possible values of $\vec{S}_k$ 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 $\mathcal{D}_k := [L_{1,k}, U_{1,k}] \times \cdots \times [L_{d,k}, U_{d,k}] \subseteq \mathcal{S}$, with $L_{1,k}, \ldots, L_{d,k}, U_{1,k}, \ldots, U_{d,k} \in \mathbb{R}$ satisfying $L_{1,k} < U_{1,k}, \ldots, L_{d,k} < U_{d,k}$,
- $V_k^{\text{OB}} \in \mathcal{I}_{\mathcal{S} \setminus \mathcal{D}_k}$

such that the following (i) and (ii) are satisfied.

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

$$\left| V_k^{\text{OB}}(\vec{s}) - V_k(\vec{s}) \right| < \epsilon_k^{\text{OB}} \tag{40}$$

or

$$\left| \mathbf{F}_k[V_k](\vec{s}) - V_k(\vec{s}) \right| < \epsilon_k^{\text{OB}} \tag{41}$$

is satisfied for any $\vec{s} \in \mathcal{S} \setminus \mathcal{D}_k$. Here, $\mathbf{F}_k[\cdot]$ is the 'flat extrapolation operator' defined as

$$\mathbf{F}_k[F](\vec{s}) := F\big(b_k(\vec{s})\big), \tag{42}$$

$$b_k(\vec{s}) := \big(\min\{U_{1,k}, \max\{L_{1,k}, s_1\}\}, \ldots, \min\{U_{d,k}, \max\{L_{d,k}, s_d\}\}\big)^T \tag{43}$$

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

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

$$O_G|\vec{s}\rangle|0\rangle = |\vec{s}\rangle\big|G(\vec{s})\big\rangle \tag{44}$$

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

$$\widetilde{O}_G|\vec{s}\rangle|0\rangle = |\vec{s}\rangle\big|\mathbf{G}_k[G](\vec{s})\big\rangle. \tag{45}$$

Here, $\mathbf{G}_k[\cdot]$ is defined as

$$
\mathbf{G}_k[H](\vec{s}) := \begin{cases} V_k^{\mathrm{OB}}(\vec{s}); & \text{if } \vec{s} \in \mathcal{A}_k, \\ \mathbf{F}_k[H](\vec{s}); & \text{otherwise} \end{cases} \tag{46}
$$

for any $H \in \mathcal{I}_{\mathcal{D}_k}$ and $\vec{s} \in \mathcal{S}$, where $\mathcal{A}_k$ is a subset of $\mathcal{S} \setminus \mathcal{D}_k$ such that (40) and (41) hold for any $\vec{s} \in \mathcal{A}_k$ and any $\vec{s} \in (\mathcal{S} \setminus \mathcal{D}_k) \setminus \mathcal{A}_k$, respectively.

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

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_k^{OB} or the flat extrapolation of V_k from moderate underlying asset prices. Postponing explanation on why this assumption is necessary to Sect. 5.2, 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 $f_k^{\mathrm{pay}}((s_1, \ldots, s_d)^T) = \max\{\kappa - s_1 - \cdots - s_d, 0\}$ with some $\kappa \in \mathbb{R}$ 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, (40) or (41) holds.

- If some of $S_{1,k}, \ldots, S_{d,k}$ 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 $V_k(\vec{S}_k) = f_k^{\mathrm{pay}}(\vec{S}_k)$.

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

Assumption 3 For every $k \in [K-1]$, $Q_k(\vec{S})$ has an analytic extension to $\mathcal{B}_{\mathcal{D}_k, \rho_k}$, where $\mathcal{D}_k$ is given in Assumption 2 and ρ_k is some real number greater than 1, and

$$
\sup_{\vec{s} \in \mathcal{B}_{\mathcal{D}_k, \rho_k}} \left| Q_k(\vec{S}) \right| \leq B_k \tag{47}
$$

holds, where B_k is some positive real number.

5.2 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 similarly to LSM; we sequentially calculate the approximate continuation value $\widetilde{Q}_k$ and option price $\widetilde{V}_k$ 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 $\mathcal{D}_k = [L_{1,k}, U_{1,k}] \times \cdots \times [L_{d,k}, U_{d,k}] \subseteq \mathcal{S}$. We begin the iterative calculation by setting $\widetilde{V}_K(\vec{S}) := f_K^{\mathrm{pay}}(\vec{S})$ for every $\vec{S} \in \mathcal{S}$. Then, for $k \in [K-1]$, given $\widetilde{V}_{k+1}$, we estimate the expected value of $\widetilde{V}_{k+1}(\vec{S}_{k+1})$ under the condition that $\vec{S}_k = \vec{S}_j^{\mathcal{D}_k, m_k}$ for every Chebyshev node $\vec{S}_j^{\mathcal{D}_k, m_k}$ by QAE, and denote the estimation

as $\widehat{Q}_{k,\vec{j}}^{\text{QAE}}$. Using these, we construct $\widetilde{Q}_k$, the Chebyshev interpolation of the approximate continuation value, and set $\widetilde{V}_k(\vec{S}) = \max\{\widetilde{Q}_k(\vec{S}), f_k^{\text{pay}}(\vec{S})\}$ for every $\vec{S} \in \mathcal{S}$. We repeat these steps until we reach $k = 1$. Finally, we estimate the expected value of $\widetilde{V}_1(\vec{S}_1)$ by QAE again, and let the result be an approximation of V_0.

The fully detailed procedure is shown in Algorithm 1.

Some additional explanations should be made. The first one is on $|\Psi_{k,\vec{j}}\rangle$ in Step 4. For every $k \in [K-1]$ and $\vec{j} \in \mathcal{J}_k$, given the approximation $\widetilde{V}_{k+1} \in \mathcal{I}_{\mathcal{D}_{k+1}}$ of V_{k+1}, we generate the state $|\Psi_{k,\vec{l}}\rangle$ on the appropriate multi-register system with the last one being single-qubit,

---

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

Require:

- $m_k \in \mathbb{N}$ for every $k \in [K-1]$, the degree of Chebyshev polynomials.
- $N_k^{\text{QAE}} \in \mathbb{N}$ for every $k \in [K-1]_0$, the iteration number in each run of QAE in calculating $\widehat{Q}_{k,\vec{j}}^{\text{QAE}}$ or $\widetilde{V}_0$.
- $N_k^{\text{rep}} \in \mathbb{N}$ for every $k \in [K-1]_0$, the number of rounds of QAE in calculating $\widehat{Q}_{k,\vec{j}}^{\text{QAE}}$ or $\widetilde{V}_0$.
- $\widetilde{V}_k^{\max} \in \mathbb{R}_+$ for every $k \in [K]$, the upper bound of $\{|\mathbf{G}_k[\hat{V}_k](\vec{s})| \mid \vec{s} \in \mathcal{S}\}$.
- $\mathcal{D}_k = [L_{1,k}, U_{1,k}] \times \cdots \times [L_{d,k}, U_{d,k}] \subseteq \mathcal{S}$ for every $k \in [K-1]$, the hyper-rectangle for Chebyshev interpolation. Here, $L_{1,k}, \ldots, L_{d,k}, U_{1,k}, \ldots, U_{d,k} \in \mathcal{S}$ satisfy $L_{1,k} < U_{1,k}, \ldots, L_{d,k} < U_{d,k}$.

1: Set $\widetilde{V}_K(\vec{S}) := f_K^{\text{pay}}(\vec{S})$ for every $\vec{S} \in \mathcal{S}$.

2: **for** $k = K - 1$ to 1 **do**

3:     **for all** $\vec{j} \in \mathcal{J}_k := [m_k]_0^d$ **do**

4:         Using $(N_k^{\text{QAE}}, N_k^{\text{rep}})$-QAE, obtain an estimation $\widetilde{P}_{k,\vec{j}}$ of the probability $P_{k,\vec{j}}$ to observe 1 on the last qubit in measuring $|\Psi_{k,\vec{j}}\rangle$ in (50), and let $(2\widetilde{P}_{k,\vec{j}} - 1)\widetilde{V}_k^{\max}$ be $\widehat{Q}_{k,\vec{j}}^{\text{QAE}}$.

5:     **end for**

6:     Set

$$\widetilde{Q}_k(\vec{S}) := \sum_{\vec{l} \in \mathcal{J}_k} \tilde{a}_{k,\vec{l}} \, \widetilde{T}_{\mathcal{D}_k,\vec{l}}(\vec{S}) \tag{48}$$

for every $\vec{S} \in \mathcal{D}_k$, with $\tilde{a}_{k,\vec{l}}$ calculated as

$$a_{k,\vec{l}} := \frac{2^{\aleph(\vec{l})}}{(m_k+1)^d} \sum_{\vec{j} \in \mathcal{J}_k} \widehat{Q}_{k,\vec{j}}^{\text{QAE}} \widetilde{T}_{\mathcal{D}_k,\vec{l}} \left(\vec{S}_{\vec{j}}^{\mathcal{D}_k,m_k}\right) \tag{49}$$

for $\vec{l} \in \mathcal{J}_k$.

7:     Set $\widetilde{V}_k(\vec{S}) := \max\{f_k^{\text{pay}}(\vec{S}), \widetilde{Q}_k(\vec{S})\}$ for any $\vec{S} \in \mathcal{D}_k$.

8: **end for**

9: Using $(N_0^{\text{QAE}}, N_0^{\text{rep}})$-QAE, obtain an estimation $\widetilde{P}_0$ of the probability P_0 to observe 1 on the last qubit in measuring $|\Psi_0\rangle$ in (53), and output $(2\widetilde{P}_0 - 1)\widetilde{V}_1^{\max}$ as $\widetilde{V}_0$.

by the following operation:

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

$$\rightarrow \big|\vec{S}_j^{\mathcal{D}_k,m_k}\big\rangle|0\rangle|0\rangle|0\rangle$$

$$\rightarrow \big|\vec{S}_j^{\mathcal{D}_k,m_k}\big\rangle \sum_{\vec{s}\in\widetilde{\mathcal{S}}_{k+1}(\vec{S}_j^{\mathcal{D}_k,m_k})} \sqrt{p_{k+1}\big(\vec{s};\vec{S}_j^{\mathcal{D}_k,m_k}\big)}|\vec{s}\rangle|0\rangle|0\rangle$$

$$\rightarrow \big|\vec{S}_j^{\mathcal{D}_k,m_k}\big\rangle \sum_{\vec{s}\in\widetilde{\mathcal{S}}_{k+1}(\vec{S}_j^{\mathcal{D}_k,m_k})} \sqrt{p_{k+1}\big(\vec{s};\vec{S}_j^{\mathcal{D}_k,m_k}\big)}|\vec{s}\rangle\big|\mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{s})\big\rangle|0\rangle$$

$$\rightarrow \big|\vec{S}_j^{\mathcal{D}_k,m_k}\big\rangle \sum_{\vec{s}\in\widetilde{\mathcal{S}}_{k+1}(\vec{S}_j^{\mathcal{D}_k,m_k})} \sqrt{p_{k+1}\big(\vec{s};\vec{S}_j^{\mathcal{D}_k,m_k}\big)}|\vec{s}\rangle\big|\mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{s})\big\rangle$$

$$\otimes \left(\sqrt{\frac{1}{2}+\frac{\mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{s})}{2\widetilde{V}_{k+1}^{\max}}}|1\rangle + \sqrt{\frac{1}{2}-\frac{\mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{s})}{2\widetilde{V}_{k+1}^{\max}}}|0\rangle\right)$$

$$=: |\Psi_{k,\vec{j}}\rangle, \tag{50}$$

where $O_{\text{step},k}$ in Assumption 1 and $\widetilde{O}_{\widetilde{V}_{k+1}}$ in Assumption 2 are used at the second and third arrows, respectively. Note that the probability to obtain 1 on the last qubit in measuring $|\Psi_{k,\vec{l}}\rangle$ is

$$P_{k,\vec{j}} = \frac{1}{2} + \frac{\widehat{Q}_k(\vec{S}_j^{\mathcal{D}_k,m_k})}{2\widetilde{V}_{k+1}^{\max}}, \tag{51}$$

where

$$\widehat{Q}_k(\vec{S}) := \mathbb{E}\big[\mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{S}_{k+1}) \mid \vec{S}_k = \vec{S}\big]$$

$$= \sum_{\vec{s}\in\widetilde{\mathcal{S}}_{k+1}(\vec{S})} p_{k+1}(\vec{s};\vec{S})\mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{s}). \tag{52}$$

Therefore, as long as $\mathbf{G}_{k+1}[\widetilde{V}_{k+1}]$ is close to V_{k+1}, $(2P_{k,\vec{j}}-1)\widetilde{V}_{k+1}^{\max} = \widehat{Q}_k(\vec{S}_j^{\mathcal{D}_k,m_k})$ is close to $Q_k(\vec{S}_j^{\mathcal{D}_k,m_k})$. This is why we can obtain approximations of the continuation values at Chebyshev nodes by Step 4, with the errors from QAEs being also small.

Second, let us explain the state $|\Psi_0\rangle$ in Step 9. Given $\widetilde{V}_1$, we can generate $|\Psi_0\rangle$ similarly to $|\Psi_{k,\vec{j}}\rangle$ as

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

$$\rightarrow |\vec{S}_0\rangle|0\rangle|0\rangle|0\rangle$$

$$\rightarrow |\vec{S}_0\rangle \sum_{\vec{s}\in\widetilde{\mathcal{S}}_1(\vec{S}_0)} \sqrt{p_1(\vec{s};\vec{S}_0)}|\vec{s}\rangle|0\rangle|0\rangle$$

$$\rightarrow |\vec{S}_0\rangle \sum_{\vec{s}\in\widetilde{\mathcal{S}}_1(\vec{S}_0)} \sqrt{p_1(\vec{s};\vec{S}_0)}|\vec{s}\rangle\big|\mathbf{G}_1[\widetilde{V}_1](\vec{s})\big\rangle|0\rangle$$

$$\rightarrow |\vec{S}_0\rangle \sum_{\vec{s} \in \widetilde{\mathcal{S}}_1(\vec{S}_0)} \sqrt{p_1(\vec{s}; \vec{S}_0)} |\vec{s}\rangle \big| \mathbf{G}_1[\widetilde{V}_1](\vec{s}) \big\rangle$$

$$\otimes \left(\sqrt{\frac{1}{2} + \frac{\mathbf{G}_1[\widetilde{V}_1](\vec{s})}{2\widetilde{V}_1^{\max}}} |1\rangle + \sqrt{\frac{1}{2} - \frac{\mathbf{G}_1[\widetilde{V}_1](\vec{s})}{2\widetilde{V}_1^{\max}}} |0\rangle \right)$$

$$=: |\Psi_0\rangle, \tag{53}$$

where the last ket corresponds to a single-qubit register. Since the probability P_0 to obtain 1 on the last qubit in measuring $|\Psi_0\rangle$ satisfies

$$(2P_0 - 1)\widetilde{V}_1^{\max} = \widehat{V}_0, \tag{54}$$

where

$$\widehat{V}_0 := \mathbb{E}\big[\mathbf{G}_1[\widetilde{V}_1](\vec{S}_1)\big] = \sum_{\vec{s} \in \widetilde{\mathcal{S}}_1(\vec{S}_0)} p_1(\vec{s}; \vec{S}_0) \mathbf{G}_1[\widetilde{V}_1](\vec{s}), \tag{55}$$

we can obtain an approximation of V_0 by Step 9, as long as $\mathbf{G}_1[\widetilde{V}_1]$ is close to V_1 and the QAE error is small.

Lastly, let us comment on the reason why Assumption 2 is necessary. This is because we have to handle underlying asset prices out of $\mathcal{D}_{k+1}$ in Steps 4 and 9, or, more specifically, in generating $|\Psi_{k,\vec{j}}\rangle$ and $|\Psi_0\rangle$. In fact, when we generate $|\Psi_{k,\vec{j}}\rangle$, $\vec{S}_{k+1}$ can be out of $\mathcal{D}_{k+1}$ with some probability. In particular, when $|\Psi_{k,\vec{j}}\rangle$ corresponds to a Chebyshev node $\vec{S}_{\vec{j}}^{\mathcal{D}_k, m_k}$ close to the boundary of $\mathcal{D}_k$, or, in other words, the condition that $\vec{S}_k$ is close to the boundary of $\mathcal{D}_k$ is imposed, such a probability becomes non-negligible.

5.3 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* 1 *to* 3, *consider Algorithm* 1. *Suppose that, for every* $k \in [K-1]$ *and* $\vec{j} \in \mathcal{J}_k$,

$$\left| \widehat{Q}_k\big(\vec{S}_{\vec{j}}^{\mathcal{D}_k, m_k}\big) - \widehat{Q}_{k,\vec{j}}^{\mathrm{QAE}} \right| \le \epsilon_k^{\mathrm{QAE}} \tag{56}$$

is satisfied, where $\epsilon_k^{\mathrm{QAE}}$ *is some positive real number. Moreover, suppose that*

$$|\widehat{V}_0 - \widetilde{V}_0| \le \epsilon_0^{\mathrm{QAE}} \tag{57}$$

is satisfied for some $\epsilon_0^{\mathrm{QAE}} \in \mathbb{R}_+$. *Then,*

$$|V_0 - \widetilde{V}_0| \le \sum_{k=1}^{K-1} \widetilde{\Lambda}_{1,k-1} \epsilon_k^{\mathrm{int}} + \sum_{k=1}^{K-1} \widetilde{\Lambda}_{1,k-1} \epsilon_k^{\mathrm{OB}} + \sum_{k=0}^{K-1} \widetilde{\Lambda}_{1,k} \epsilon_k^{\mathrm{QAE}} \tag{58}$$

holds, where, for $k \in [K-1]$ *and* $k' \in [K-1]_0$,

$$\epsilon_k^{\mathrm{int}} := \epsilon_{\mathrm{int}}(\rho_k, d, m_k, B_k) \tag{59}$$

and

$$\widetilde{\Lambda}_{k,k'} := \begin{cases} \prod_{i=k}^{k'} \Lambda_i; & \text{if } k \leq k' \\ 1; & \text{otherwise} \end{cases} \tag{60}$$

$$\Lambda_k := \left(\frac{2}{\pi} \log(m_k + 1) + 1 \right)^d. \tag{61}$$

The proof is presented in Appendix A.1.

5.4 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 ϵ be a real number in $(0, 0.1)$. Under Assumptions 1 to 3, consider Algorithm 1 with the following parameters*:

(i) $m_k, k \in [K-1]$ *satisfying* $m_k \geq m_k^{\text{th}}$ *with*

$$m_1^{\text{th}} = \left\lceil \frac{1}{\log \rho_1} \log \left(\frac{2^{d/2+2}\sqrt{d}(K-1)(1-\rho_1^{-2})^{-d/2}B_1}{\epsilon \widetilde{V}_1^{\max}} \right) \right\rceil$$

$$m_k^{\text{th}} = \left\lceil \frac{1}{\log \rho_k} \log \left(\frac{2^{d/2+2}\sqrt{d}(K-1)(1-\rho_k^{-2})^{-d/2}\widetilde{\Lambda}_{1,k-1}^{\text{th}}B_k}{\epsilon \widetilde{V}_1^{\max}} \right) \right\rceil \tag{62}$$

for $k = 2, \dots, K-1$,

where $\widetilde{\Lambda}_{1,k-1}^{\text{th}}$ is determined as $\widetilde{\Lambda}_{1,k-1}$ in (60) with $m_1 = m_1^{\text{th}}, \dots, m_{k-1} = m_{k-1}^{\text{th}}$.

(ii) $N_k^{\text{QAE}}, k \in [K-1]_0$ *set as*

$$N_k^{\text{QAE}} = \left\lceil \frac{7}{\bar{\epsilon}_k} \right\rceil. \tag{63}$$

Here, $\bar{\epsilon}_0, \dots, \bar{\epsilon}_{K-1}$ are given by

$$\bar{\epsilon}_0 = \frac{1}{1 + \sum_{k'=1}^{K-1} \sqrt{(m_{k'}+1)^d \widetilde{\Lambda}_{1,k'}}} \cdot \frac{\epsilon}{4}$$

$$\bar{\epsilon}_k = \frac{\sqrt{(m_k+1)^d/\widetilde{\Lambda}_{1,k}}}{1 + \sum_{k'=1}^{K-1} \sqrt{(m_{k'}+1)^d \widetilde{\Lambda}_{1,k'}}} \cdot \frac{\widetilde{V}_1^{\max}\epsilon}{4\widetilde{V}_k^{\max}} \quad \text{for } k = 1, \dots, K-1, \tag{64}$$

where $m_0, \dots, m_{K-1}$ are set as (i) and $\widetilde{\Lambda}_{1,1}, \dots, \widetilde{\Lambda}_{1,K-1}$ are given as (60) with such $m_0, \dots, m_{K-1}$.

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

$$N_{\text{rep}} := 12 \left\lceil \log \left(\frac{N_{\text{est}}}{0.01} \right) \right\rceil + 1, \tag{65}$$

for every $k \in [K-1]_0$. Here, $N_{\text{est}} := 1 + \sum_{k'=1}^{K-1}(m_{k'}+1)^d$ with $\{m_k\}$ set as (i).

Moreover, suppose that $\epsilon_1^{\mathrm{OB}},\ldots,\epsilon_{K-1}^{\mathrm{OB}}$ are 0. Then, Algorithm 1 outputs $\widetilde{V}_0$ satisfying $|V_0 - \widetilde{V}_0| \le \epsilon \widetilde{V}_1^{\max}$ with probability higher than 0.99.

The proof is presented in Appendix A.2.

We here explain why the parameters are set as above. As we see in the proof in Appendix A.2, $m_1,\ldots,m_{K-1}$ satisfying (62) make the first term in the RHS in (58) smaller than $\epsilon \widetilde{V}_1^{\max}/2$. Then, for such $\{m_k\}_{k=1,\ldots,K-1}$, $\{N_k^{\mathrm{QAE}}\}_{k=0,\ldots,K-1}$ are determined as (63) so that

$$\frac{N_{\mathrm{tot}}}{N_{\mathrm{rep}}} = N_0^{\mathrm{QAE}} + \sum_{k=1}^{K-1}(m_k+1)^d N_k^{\mathrm{QAE}}, \tag{66}$$

that is, the total number N_{tot} of calls to $\{O_{\mathrm{step},k}\}_{k=0,\ldots,K-1}$ divided by the QAE repetition number N^{rep}, is minimized under the constraint that, if all the QAEs in Algorithm 1 succeed, the third term in the RHS in (58) is smaller than $\epsilon \widetilde{V}_1^{\max}/2$. Finally, $\{N_k^{\mathrm{rep}}\}_{k=0,\ldots,K-1}$ 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 $\epsilon \widetilde{V}_1^{\max}$ with probability higher than 0.99.

Note that, in reality, it is difficult to set m_k to m_k^{th}, since ρ_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 ϵ 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 $\{N_k^{\mathrm{rep}}\}$ compensating the change of $\{m_k\}$.

Let us consider the dependency of the total complexity on the error tolerance ϵ. We see that

$$m_k = O\big(\log(\epsilon^{-1})\mathrm{polyloglog}(\epsilon^{-1})\big), \tag{67}$$

$$N_k^{\mathrm{QAE}} = O\big(\epsilon^{-1} \times \mathrm{polyloglog}(\epsilon^{-1})\big) \tag{68}$$

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

$$N_0^{\mathrm{QAE}} = O\big(\epsilon^{-1}\log^{d/2}(\epsilon^{-1})\,\mathrm{polyloglog}(\epsilon^{-1})\big), \tag{69}$$

where $\mathrm{polyloglog}(\cdot)$ means $\mathrm{polylog}(\log(\cdot))$. Combining these with (66), we obtain

$$N_{\mathrm{tot}} = O\big(\epsilon^{-1}\log^d(\epsilon^{-1})\,\mathrm{polyloglog}(\epsilon^{-1})\big), \tag{70}$$

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

6 Discussion
6.1 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.

6.2 Exponential factor with respect to the number of exercise dates in the error bound

Now, let us make a comment on the factor $\widetilde{\Lambda}_{1,K-1}$ in (58), which is reflected into the polyloglog factors in (70). This exponentially depends on the number of exercise dates K. Therefore, it seems that the error on the option price exponentially grows as K increase, and so does the complexity sufficient to achieve a given error tolerance. Similar situations arose in the error analyses for LSM [53, 55, 57, 59–61] and classical Chebyshev interpolation-based methods [48].

However, we should note that (58) is an upper bound on the error, and that the actual error might not necessarily grows exponentially against K. In fact, in the numerical experiment in [48], where American options were approximately priced as Bermudan options with a small exercise date interval, the error was suppressed even if hundreds or thousands of exercise dates were set.

Let us now consider why the factor exponentially depending on K appears in (58). In the derivation of (58) described in Appendix A.1, we make Assumption 3 on the analyticity and boundedness of the continuation values Q_k, and apply Theorem 1 to Chebyshev interpolation of Q_k in Algorithm 1. Since we use not $Q_k(\vec{S}_{\vec{j}}^{\mathcal{D}_k,m_k})$, the values of Q_k at the Chebyshev nodes, but $\widehat{Q}_{k,\vec{j}}^{\mathrm{QAE}}$, estimates on $Q_k(\vec{S}_{\vec{j}}^{\mathcal{D}_k,m_k})$ with some errors, in interpolation, the term like the second term in the RHS in (1) arises in the upper bound of the difference between Q_k and the interpolant $\widetilde{Q}_k$, and is amplified at every later interpolation.

On the other hand, we can consider that the actual target of Chebyshev interpolation is not Q_k but $\widehat{Q}_k$ in (52). That is, we can regard $\widehat{Q}_{k,\vec{j}}^{\mathrm{QAE}}$ as not an estimate on $Q_k(\vec{S}_{\vec{j}}^{\mathcal{D}_k,m_k})$ but that on $\widehat{Q}_k(\vec{S}_{\vec{j}}^{\mathcal{D}_k,m_k})$. Then, we can make an assumption not on analyticity and boundedness

of Q_k but those of $\widehat{Q}_k$. This leads to a different error bound than Theorem 3. Actually, if we make Assumption 4 instead of Assumption 3, we have an error bound with no exponential factor as shown in Theorem 4.

Assumption 4 For every $k \in [K - 1]$, $\widehat{Q}_k(\vec{S})$ has an analytic extension to $\mathcal{B}_{\mathcal{D}_k, \widehat{\rho}_k}$, where $\mathcal{D}_k$ is given in Assumption 2 and $\widehat{\rho}_k$ is some real number greater than 1, and

$$\sup_{\vec{s} \in \mathcal{B}_{\mathcal{D}_k, \widehat{\rho}_k}} \left| \widehat{Q}_k(\vec{S}) \right| \le \widehat{B}_k \tag{71}$$

holds, where $\widehat{B}_k$ 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 $\vec{j} \in \mathcal{J}_k$, (56) is satisfied for some $\epsilon_k^{\mathrm{QAE}} \in \mathbb{R}_+$, and that (57) is satisfied for some $\epsilon_0^{\mathrm{QAE}} \in \mathbb{R}_+$. Then,*

$$|V_0 - \widetilde{V}_0| \le \sum_{k=1}^{K-1} \epsilon_k^{\mathrm{OB}} + \sum_{k=1}^{K-1} \widetilde{\epsilon}_k^{\mathrm{int}} + \sum_{k=0}^{K-1} \Lambda_k \epsilon_k^{\mathrm{QAE}} \tag{72}$$

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

$$\widetilde{\epsilon}_k^{\mathrm{int}} := \epsilon_{\mathrm{int}}(\widetilde{\rho}_k, d, m_k, \widetilde{B}_k), \tag{73}$$

and Λ_k is defined as (61).

The proof is presented in Appendix A.3. Note that a similar point has been made for LSM in [59] (Theorem 3.1).

Of course, $\widehat{Q}_k$ is defined as (52) with $\widetilde{V}_{k+1}$, 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 Q_k and/or other quantities determined independently from pricing algorithms. We leave considering whether we can obtain an error bound similar to Theorem 4 under such assumptions or not as a future work.

6.3 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 $\widetilde{O}(D^{7/2}\kappa^4/\epsilon)$ with the tolerance ϵ, the explanatory variable number D, and the condition number κ of the design matrix, from which we expect the quadratic speed-up of LSM with respect to ϵ. 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 κ, which might make the algorithm disadvantageous. Therefore, it will be crucial to evaluate these, especially κ, in

addition to finding the basis function set which makes κ as small as possible. We will consider this direction in the future work.

7 Conclusions

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 ϵ. As expected, this method has the complexity depending on ϵ as $\widetilde{O}(\epsilon^{-1})$, 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 Sect. 6.3. 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.

Appendix
A.1 Proof of Theorem 3

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

$$\epsilon_k \leq \tilde{\epsilon}_k \tag{74}$$

holds, where

$$\epsilon_k := \max_{\vec{S} \in \mathcal{D}_k} \left| V_k(\vec{S}) - \widetilde{V}_k(\vec{S}) \right|, \tag{75}$$

and

$$\tilde{\epsilon}_k := \begin{cases} \epsilon_{K-1}^{\mathrm{int}} + \Lambda_k \epsilon_{K-1}^{\mathrm{QAE}}; & \text{for } k = K-1, \\ \epsilon_k^{\mathrm{int}} + \Lambda_k (\epsilon_{k+1} + \epsilon_{k+1}^{\mathrm{OB}} + \epsilon_k^{\mathrm{QAE}}); & \text{for } k = 1, \ldots, K-2. \end{cases} \tag{76}$$

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

$$\left| V_k(\vec{S}_k) - \mathbf{G}_k[\widetilde{V}_k](\vec{S}_k) \right| = \left| V_k(\vec{S}_k) - \widetilde{V}_k(\vec{S}_k) \right| \leq \epsilon_k \tag{77}$$

holds if $\vec{S}_k \in \mathcal{D}_k$, and, under Assumption 2, either

$$\left| V_k(\vec{S}_k) - \mathbf{G}_k[\widetilde{V}_k](\vec{S}_k) \right| = \left| V_k(\vec{S}_k) - V_k^{\mathrm{OB}}(\vec{S}_k) \right| \leq \epsilon_k^{\mathrm{OB}} \tag{78}$$

or

$$
\begin{aligned}
&\left| V_k(\vec{S}_k) - \mathbf{G}_k[\widetilde{V}_k](\vec{S}_k) \right| \\
&= \left| V_k(\vec{S}_k) - \mathbf{F}_k[\widetilde{V}_k](\vec{S}_k) \right| \\
&\le \left| V_k(\vec{S}_k) - \mathbf{F}_k[V_k](\vec{S}_k) \right| + \left| \mathbf{F}_k[V_k](\vec{S}_k) - \mathbf{F}_k[\widetilde{V}_k](\vec{S}_k) \right| \\
&= \left| V_k(\vec{S}_k) - \mathbf{F}_k[V_k](\vec{S}_k) \right| + \left| V_k\big(b_k(\vec{S}_k)\big) - \widetilde{V}_k\big(b_k(\vec{S}_k)\big) \right| \\
&\le \epsilon_k^{\mathrm{OB}} + \epsilon_k
\end{aligned}
\tag{79}
$$

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

$$
\left| V_k(\vec{S}_k) - \mathbf{G}_k[\widetilde{V}_k](\vec{S}_k) \right| \le \epsilon_k + \epsilon_k^{\mathrm{OB}}
\tag{80}
$$

for any $\vec{S}_k \in \mathcal{S}$. This leads to

$$
\begin{aligned}
&\left| Q_k(\vec{S}) - \widehat{Q}_k(\vec{S}) \right| \\
&= \left| \mathbb{E}\big[V_{k+1}(\vec{S}_{k+1}) - \mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{S}_{k+1}) \mid \vec{S}_k = \vec{S} \big] \right| \\
&\le \mathbb{E}\big[\left| V_{k+1}(\vec{S}_{k+1}) - \mathbf{G}_{k+1}[\widetilde{V}_{k+1}](\vec{S}_{k+1}) \right| \mid \vec{S}_k = \vec{S} \big] \\
&\le \epsilon_{k+1} + \epsilon_{k+1}^{\mathrm{OB}}
\end{aligned}
\tag{81}
$$

for any $k \in [K-2]$ and $\vec{S} \in \mathcal{D}_k$. Thus, with (56), we obtain

$$
\begin{aligned}
&\left| Q_k\big(\vec{S}_{\vec{j}}^{\mathcal{D}_k, m_k}\big) - \widehat{Q}_{k,\vec{j}}^{\mathrm{QAE}} \right| \\
&\le \left| Q_k\big(\vec{S}_{\vec{j}}^{\mathcal{D}_k, m_k}\big) - \widehat{Q}_k\big(\vec{S}_{\vec{j}}^{\mathcal{D}_k, m_k}\big) \right| + \left| \widehat{Q}_k\big(\vec{S}_{\vec{j}}^{\mathcal{D}_k, m_k}\big) - \widehat{Q}_{k,\vec{j}}^{\mathrm{QAE}} \right| \\
&\le \epsilon_{k+1} + \epsilon_{k+1}^{\mathrm{OB}} + \epsilon_k^{\mathrm{QAE}}
\end{aligned}
\tag{82}
$$

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

$$
\left| Q_{K-1}\big(\vec{S}_{\vec{j}}^{\mathcal{D}_{K-1}, m_{K-1}}\big) - \widehat{Q}_{K-1,\vec{j}}^{\mathrm{QAE}} \right| \le \epsilon_{K-1}^{\mathrm{QAE}}
\tag{83}
$$

for any $\vec{j} \in \mathcal{J}_{K-1}$. Then, under Assumption 3, invoking Theorem 1, we obtain

$$
\left| Q_k(\vec{S}) - \widetilde{Q}_k(\vec{S}) \right| \le \tilde{\epsilon}_k
\tag{84}
$$

for any $k \in [K-1]$ and $\vec{S} \in \mathcal{D}_k$, which immediately leads to (74) as

$$
\begin{aligned}
&\left| V_k(\vec{S}) - \widetilde{V}_k(\vec{S}) \right| \\
&= \left| \max\big\{ f_k^{\mathrm{pay}}(\vec{S}), Q_k(\vec{S}) \big\} - \max\big\{ f_k^{\mathrm{pay}}(\vec{S}), \widetilde{Q}_k(\vec{S}) \big\} \right| \\
&\le \left| Q_k(\vec{S}) - \widetilde{Q}_k(\vec{S}) \right| \\
&\le \tilde{\epsilon}_k.
\end{aligned}
\tag{85}
$$

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

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

$$\epsilon_k \leq \sum_{k'=k}^{K-1} \widetilde{\Lambda}_{k,k'-1}\epsilon_{k'}^{\text{int}} + \sum_{k'=k+1}^{K-1} \widetilde{\Lambda}_{k,k'-1}\epsilon_{k'}^{\text{OB}} + \sum_{k'=k}^{K-1} \widetilde{\Lambda}_{k,k'}\epsilon_{k'}^{\text{QAE}}, \tag{86}$$

holds. We prove this by induction. For $k = K-2$, (74) implies that

$$\epsilon_{K-2}$$
$$\leq \epsilon_{K-2}^{\text{int}} + \Lambda_{K-2}\big(\epsilon_{K-1} + \epsilon_{K-1}^{\text{OB}} + \epsilon_{K-2}^{\text{QAE}}\big)$$
$$\leq \epsilon_{K-2}^{\text{int}} + \Lambda_{K-2}\big(\epsilon_{K-1}^{\text{int}} + \Lambda_{K-1}\epsilon_{K-1}^{\text{QAE}} + \epsilon_{K-1}^{\text{OB}} + \epsilon_{K-2}^{\text{QAE}}\big)$$
$$= \sum_{k'=K-2}^{K-1} \widetilde{\Lambda}_{K-2,k'-1}\epsilon_{k'}^{\text{int}} + \sum_{k'=K-1}^{K-1} \widetilde{\Lambda}_{K-2,k'-1}\epsilon_{k'}^{\text{OB}} + \sum_{k'=K-2}^{K-1} \widetilde{\Lambda}_{K-2,k'}\epsilon_{k'}^{\text{QAE}}. \tag{87}$$

Similarly, if (86) hold for $k \in \{2,\ldots,K-2\}$, (74) implies that

$$\epsilon_{k-1}$$
$$\leq \epsilon_{k-1}^{\text{int}} + \Lambda_{k-1}\big(\epsilon_k + \epsilon_k^{\text{OB}} + \epsilon_{k-1}^{\text{QAE}}\big)$$
$$\leq \epsilon_{k-1}^{\text{int}} + \Lambda_{k-1}\left(\sum_{k'=k}^{K-1} \widetilde{\Lambda}_{k,k'-1}\epsilon_{k'}^{\text{int}} + \sum_{k'=k+1}^{K-1} \widetilde{\Lambda}_{k,k'-1}\epsilon_{k'}^{\text{OB}} + \sum_{k'=k}^{K-1} \widetilde{\Lambda}_{k,k'}\epsilon_{k'}^{\text{QAE}} + \epsilon_k^{\text{OB}} + \epsilon_{k-1}^{\text{QAE}}\right)$$
$$= \sum_{k'=k-1}^{K-1} \widetilde{\Lambda}_{k-1,k'-1}\epsilon_{k'}^{\text{int}} + \sum_{k'=k}^{K-1} \widetilde{\Lambda}_{k-1,k'-1}\epsilon_{k'}^{\text{OB}} + \sum_{k'=k-1}^{K-1} \widetilde{\Lambda}_{k-1,k'}\epsilon_{k'}^{\text{QAE}}. \tag{88}$$

Therefore, (86) is proved for every $k \in [K-2]$.

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

$$|V_0 - \widehat{V}_0|$$
$$= \big|\mathbb{E}\big[V_1(\vec{S}_1)\big] - \mathbb{E}\big[\mathbf{G}_1[\tilde{V}_1](\vec{S}_1)\big]\big|$$
$$\leq \mathbb{E}\big[\big|V_1(\vec{S}_1) - \mathbf{G}_1[\tilde{V}_1](\vec{S}_1)\big|\big]$$
$$\leq \epsilon_1 + \epsilon_1^{\text{OB}}$$
$$\leq \sum_{k=1}^{K-1} \widetilde{\Lambda}_{1,k-1}\epsilon_k^{\text{int}} + \sum_{k=1}^{K-1} \widetilde{\Lambda}_{1,k-1}\epsilon_k^{\text{OB}} + \sum_{k=1}^{K-1} \widetilde{\Lambda}_{1,k}\epsilon_k^{\text{QAE}}, \tag{89}$$

where we used (80) and (86) at the second and last inequalities, respectively. Combining this and (57), we obtain

$$|V_0 - \widetilde{V}_0|$$
$$\leq |V_0 - \widehat{V}_0| + |\widehat{V}_0 - \widetilde{V}_0|$$
$$= \sum_{k=1}^{K-1} \widetilde{\Lambda}_{1,k-1}\epsilon_k^{\text{int}} + \sum_{k=1}^{K-1} \widetilde{\Lambda}_{1,k-1}\epsilon_k^{\text{OB}} + \sum_{k=0}^{K-1} \widetilde{\Lambda}_{1,k}\epsilon_k^{\text{QAE}}. \tag{90}$$

$$\square$$

A.2 Proof of Corollary 1

Proof By simple algebra, we see that $m_1, \ldots, m_{K-1}$ satisfying (62) lead to

$$\widetilde{\Lambda}_{1,k-1}\epsilon_k^{\mathrm{int}} \leq \frac{\epsilon \widetilde{V}_1^{\max}}{2(K-1)} \tag{91}$$

for every $k \in [K-1]$.

On the other hand, Theorem 2 implies that, for every $k \in [K-1]$ and $\vec{j} \in \mathcal{J}_k$, using N_k^{QAE} set as (63), Step 4 in Algorithm 1 gives us $\widetilde{P}_{k,\vec{j}}$, an estimate on $P_{k,\vec{j}}$ in (51), satisfying

$$|P_{k,\vec{j}} - \widetilde{P}_{k,\vec{j}}| \leq \bar{\epsilon}_k, \tag{92}$$

and then $\widehat{Q}_{k,\vec{j}}^{\mathrm{QAE}}$ satisfying

$$
\begin{aligned}
&\left|\widehat{Q}_k\big(\vec{S}_{\vec{j}}^{\mathcal{D}_k, m_k}\big) - \widehat{Q}_{k,\vec{j}}^{\mathrm{QAE}}\right| \\
&= \left|(2P_{k,\vec{j}} - 1)\widetilde{V}_k^{\max} - (2\widetilde{P}_{k,\vec{j}} - 1)\widetilde{V}_k^{\max}\right| \\
&\leq 2\widetilde{V}_k^{\max}\bar{\epsilon}_k \\
&\leq \frac{\sqrt{(m_k+1)^d/\widetilde{\Lambda}_{1,k}}}{1 + \sum_{k'=1}^{K-1}\sqrt{(m_{k'}+1)^d\widetilde{\Lambda}_{1,k'}}} \cdot \frac{\widetilde{V}_1^{\max}\epsilon}{2} =: \widetilde{\epsilon}_k^{\mathrm{QAE}},
\end{aligned}
\tag{93}
$$

with some probability. Similarly, it is implied that, with N_0^{QAE} set as (63), Step 9 gives us $\widetilde{P}_0$, an estimation on P_0, satisfying

$$|P_0 - \widetilde{P}_0| \leq \bar{\epsilon}_0, \tag{94}$$

and then $\widetilde{V}_0$ satisfying

$$
\begin{aligned}
&|\widehat{V}_0 - \widetilde{V}_0| \\
&= \left|(2P_0 - 1)\widetilde{V}_0^{\max} - (2\widetilde{P}_0 - 1)\widetilde{V}_0^{\max}\right| \\
&\leq 2\widetilde{V}_0^{\max}\bar{\epsilon}_0 \\
&\leq \frac{1}{1 + \sum_{k'=1}^{K-1}\sqrt{(m_{k'}+1)^d\widetilde{\Lambda}_{1,k'}}} \cdot \frac{\widetilde{V}_1^{\max}\epsilon}{2} =: \widetilde{\epsilon}_0^{\mathrm{QAE}},
\end{aligned}
\tag{95}
$$

with some probability. Therefore, when all of these succeed,

$$
\begin{aligned}
&|V_0 - \widetilde{V}_0| \\
&\leq \sum_{k=1}^{K-1}\widetilde{\Lambda}_{1,k-1}\epsilon_k^{\mathrm{int}} + \sum_{k=1}^{K-1}\widetilde{\Lambda}_{1,k-1}\epsilon_k^{\mathrm{OB}} + \sum_{k=0}^{K-1}\widetilde{\Lambda}_{1,k}\widetilde{\epsilon}_k^{\mathrm{QAE}} \\
&\leq \frac{\epsilon \widetilde{V}_1^{\max}}{2} + 0 + \frac{\epsilon \widetilde{V}_1^{\max}}{2} \\
&= \epsilon \widetilde{V}_1^{\max}
\end{aligned}
\tag{96}
$$

holds, according to Theorem 3. Here, we used (91), (93), (95), and the assumption that $\epsilon_1^{\mathrm{OB}}, \ldots, \epsilon_{K-1}^{\mathrm{OB}}$ are 0, along with simple algebra.

The remaining task is proving that the probability P_{all} that all of the estimations in Steps 4 and 9 succeed is larger than 0.99 under the setting of N_k^{rep} as (65). Note that, according to Theorem 2, with the setting as (65), the probability that Step 4 for a set of $k \in [K-1]$ and $\vec{j} \in \mathcal{J}_k$ outputs $\widetilde{P}_{k,\vec{j}}$ satisfying (92) is higher than

$$1 - \frac{0.01}{N_{\mathrm{est}}}. \tag{97}$$

Similarly, the probability that Step 9 outputs $\widetilde{P}_0$ satisfying (94) is also higher than (97). Besides, the total number of these estimations is N_{est}. Combining these, we obtain a lower bound of P_{all} as

$$P_{\mathrm{all}} \geq \left(1 - \frac{0.01}{N_{\mathrm{est}}}\right)^{N_{\mathrm{est}}} \geq 1 - 0.01 = 0.99, \tag{98}$$

which completes the proof.　　　　　　　　　　　　　　　　　　　　　　　　　　　□

A.3 Proof of Theorem 4

Proof Because of (56), Theorem 1 implies that

$$\left|\widehat{Q}_k(\vec{S}) - \widetilde{Q}_k(\vec{S})\right| \leq \widetilde{\epsilon}_k^{\mathrm{int}} + \Lambda_k \epsilon_k^{\mathrm{QAE}} \tag{99}$$

for every $k \in [K-1]$ and $\vec{S} \in \mathcal{D}_k$. Besides, for every $k \in [K-2]$ and $\vec{S} \in \mathcal{D}_k$, $|Q_k(\vec{S}) - \widehat{Q}_k(\vec{S})| \leq \epsilon_{k+1} + \epsilon_{k+1}^{\mathrm{OB}}$ holds as (81), where ϵ_k is defined as (75) for every $k \in [K-1]$, whereas $Q_{K-1}(\vec{S}) = \widehat{Q}_{K-1}(\vec{S})$ by definition of $\mathbf{G}_K[\cdot]$. Combining these, we see that, for every $k \in [K-1]$ and $\vec{S} \in \mathcal{D}_k$,

$$\left|Q_k(\vec{S}) - \widetilde{Q}_k(\vec{S})\right| \leq \left|Q_k(\vec{S}) - \widehat{Q}_k(\vec{S})\right| + \left|\widehat{Q}_k(\vec{S}) - \widetilde{Q}_k(\vec{S})\right|$$
$$\leq \epsilon_{k+1} + \epsilon_{k+1}^{\mathrm{OB}} + \widetilde{\epsilon}_k^{\mathrm{int}} + \Lambda_k \epsilon_k^{\mathrm{QAE}} \tag{100}$$

holds with $\epsilon_K = 0$ and $\epsilon_K^{\mathrm{OB}} = 0$, which leads to

$$\left|V_k(\vec{S}) - \widetilde{V}_k(\vec{S})\right| \leq \epsilon_{k+1} + \epsilon_{k+1}^{\mathrm{OB}} + \widetilde{\epsilon}_k^{\mathrm{int}} + \Lambda_k \epsilon_k^{\mathrm{QAE}} \tag{101}$$

similarly to (85). Therefore, for every $k \in [K-1]$,

$$\epsilon_k = \max_{\vec{S} \in \mathcal{D}_k} \left|V_k(\vec{S}) - \widetilde{V}_k(\vec{S})\right| \leq \epsilon_{k+1} + \epsilon_{k+1}^{\mathrm{OB}} + \widetilde{\epsilon}_k^{\mathrm{int}} + \Lambda_k \epsilon_k^{\mathrm{QAE}}. \tag{102}$$

This implies

$$\epsilon_1 \leq \sum_{k=2}^{K-1} \epsilon_k^{\mathrm{OB}} + \sum_{k=1}^{K-1} \widetilde{\epsilon}_k^{\mathrm{int}} + \sum_{k=1}^{K-1} \Lambda_k \epsilon_k^{\mathrm{QAE}}. \tag{103}$$

Finally, combining (103) with (57) and $|V_0 - \widehat{V}_0| \leq \epsilon_1 + \epsilon_1^{\mathrm{OB}}$, which we can see as (89), we obtain

$$
\begin{aligned}
|V_0 - \widetilde{V}_0| \\
\leq |V_0 - \widehat{V}_0| + |\widehat{V}_0 - \widetilde{V}_0| \\
\leq \epsilon_1 + \epsilon_1^{\mathrm{OB}} + \epsilon_0^{\mathrm{QAE}} \\
\leq \sum_{k=1}^{K-1} \epsilon_k^{\mathrm{OB}} + \sum_{k=1}^{K-1} \widetilde{\epsilon}_k^{\mathrm{int}} + \sum_{k=0}^{K-1} \Lambda_k \epsilon_k^{\mathrm{QAE}}.
\end{aligned}
\tag{104}
$$
□

Abbreviations
LSM, least-squares Monte Carlo; QAE, quantum amplitude estimation; SDE, stochastic differential equation; RHS, right hand side.

Availability of data and materials
Not applicable.

Declarations

Competing interests
The author declares that he has no competing interests.

Authors' contributions
KM as the sole author of the manuscript, conceived, designed and performed the analysis; he also wrote and reviewed the paper. The author read and approved the final manuscript.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References
1. Nielsen MA, Chuang IL. Quantum computation and quantum information. Cambridge: Cambridge University Press; 2010.
2. Orus R et al. Quantum computing for finance: overview and prospects. Rev Phys. 2019;4:100028.
3. Egger DJ et al. Quantum computing for finance: state of the art and future prospects. IEEE Trans Quantum Eng. 2020;1:3101724.
4. Bouland A, et al. Prospects and challenges of quantum finance. arXiv:2011.06492.
5. Hull JC. Options, futures, and other derivatives. New York: Prentice Hall; 2012.
6. Shreve S. Stochastic calculus for finance I & II. Berlin: Springer; 2004.
7. Rebentrost P et al. Quantum computational finance: Monte Carlo pricing of financial derivatives. Phys Rev A. 2018;98:022321.
8. Martin A et al. Towards pricing financial derivatives with an IBM quantum computer. Phys Rev Res. 2021;3:013167.
9. Stamatopoulos N et al. Option pricing using quantum computers. Quantum. 2020;4:291.
10. Ramos-Calderer S et al. Quantum unary approach to option pricing. Phys Rev A. 2021;103:032414.
11. Fontanela F, et al. A quantum algorithm for linear PDEs arising in finance. arXiv:1912.02753.
12. Vazquez AC, Woerner S. Efficient state preparation for quantum amplitude estimation. Phys Rev Appl. 2021;15:034027.
13. Kaneko K, et al. Quantum pricing with a smile: implementation of local volatility model on quantum computer. arXiv:2007.01467.
14. Tang H, et al. Quantum computation for pricing the collateralized debt obligations. arXiv:2008.04110.
15. Chakrabarti S et al. A threshold for quantum advantage in derivative pricing. Quantum. 2021;5:463.
16. An D et al. Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance. Quantum. 2021;5:481.
17. Gonzalez-Conde J, et al. Pricing financial derivatives with exponential quantum speedup. arXiv:2101.04023.
18. Radha SK. Quantum option pricing using Wick rotated imaginary time evolution. arXiv:2101.04280.

19. Woerner S, Egger DJ. Quantum risk analysis. npj Quantum Inf. 2019;5(1):1.
20. Egger DJ et al. Credit risk analysis using quantum computers. IEEE Trans Comput. 2021;70:2136.
21. Miyamoto K, Shiohara K. Reduction of qubits in a quantum algorithm for Monte Carlo simulation by a pseudo-random-number generator. Phys Rev A. 2020;102:022424.
22. Kaneko K et al. Quantum speedup of Monte Carlo integration with respect to the number of dimensions and its application to finance. Quantum Inf Process. 2021;20:185.
23. Rebentrost P, Lloyd S. Quantum computational finance: quantum algorithm for portfolio optimization. arXiv:1811.03975.
24. Kerenidis I et al. Quantum algorithms for portfolio optimization. In: Proceedings of the 1st ACM conference on advances in financial technologies. 2019. p. 147.
25. Hodson M, et al. Portfolio rebalancing experiments using the quantum alternating operator ansatz. arXiv:1911.05296.
26. Glasserman P. Monte Carlo methods in financial engineering. Berlin: Springer; 2003.
27. Tavella D, Randall C. Pricing financial instruments: the finite difference method. New York: Wiley; 2000.
28. Duffy DJ. Finite difference methods in financial engineering: a partial differential equation approach. New York: Wiley; 2006.
29. Longstaff FA, Schwartz ES. Valuing American options by simulation: a simple least-squares approach. Rev Financ Stud. 2001;14:113.
30. Montanaro A. Quantum speedup of Monte Carlo methods. Proc R Soc Ser A. 2015;471:2181.
31. Suzuki Y et al. Amplitude estimation without phase estimation. Quantum Inf Process. 2020;19:75.
32. Herbert S. Quantum Monte-Carlo integration: the full advantage in minimal circuit depth. arXiv:2105.09100.
33. Brassard G et al. Quantum amplitude amplification and estimation. Contemp Math. 2002;305:53.
34. Aaronson S, Rall P. Quantum approximate counting, simplified. In: Symposium on simplicity in algorithms. Philadelphia: SIAM; 2020. p. 24–32.
35. Grinko D et al. Iterative quantum amplitude estimation. npj Quantum Inf. 2021;7:52.
36. Nakaji K. Faster amplitude estimation. Quantum Inf Comput. 2020;20:1109.
37. Brown EG, et al. Quantum amplitude estimation in the presence of noise. arXiv:2006.14145.
38. Tanaka T, et al. Amplitude estimation via maximum likelihood on noisy quantum computer. arXiv:2006.16223.
39. Kerenidis I, Prakash A. A method for amplitude estimation with noisy intermediate-scale quantum computers. U.S. Patent Application No. 16/892,229. 2020.
40. Uno S, et al. Modified Grover operator for amplitude estimation. arXiv:2010.11656.
41. Giurgica-Tiron T, et al. Low depth algorithms for quantum amplitude estimation. arXiv:2012.03348.
42. Wang G et al. Bayesian inference with engineered likelihood functions for robust amplitude estimation. PRX Quantum. 2021;2:010346.
43. Trefethen LN. Approximation theory and approximation practice. Philadelphia: SIAM; 2013.
44. Sullivan MA. Valuing American put options using Gaussian quadrature. Rev Financ Stud. 2000;13:75.
45. Lim H et al. Efficient pricing of Bermudan options using recombining quadratures. J Comput Appl Math. 2014;271:195.
46. Mahlstedt M. Complexity reduction for option pricing. Ph.D. thesis. Technische Universität München; 2017.
47. Gaß M et al. Chebyshev interpolation for parametric option pricing. Finance Stoch. 2018;22:701.
48. Glau K et al. A new approach for American option pricing: the dynamic Chebyshev method. SIAM J Sci Comput. 2019;41(1):B153.
49. Glau K, et al. Fast calculation of credit exposures for barrier and Bermudan options using Chebyshev interpolation. arXiv:1905.00238.
50. Glau K et al. Speed-up credit exposure calculations for pricing and risk management. Quant Finance. 2021;21:481.
51. Sauter S, Schwab C. Boundary element methods. Berlin: Springer; 2010.
52. Clement E et al. An analysis of a least squares regression method for American option pricing. Finance Stoch. 2002;6(4):449.
53. Glasserman P, Yu B. Number of paths vs. number of basis functions in American option pricing. Ann Appl Probab. 2004;14(4):2090.
54. Stentoft L. Convergence of the least squares Monte Carlo approach to American option valuation. Manag Sci. 2004;50(9):1193.
55. Egloff D. Monte Carlo algorithms for optimal stopping and statistical learning. Ann Appl Probab. 2005;15:1396.
56. Gobet E et al. A regression-based Monte Carlo method to solve backward stochastic differential equations. Ann Appl Probab. 2005;15(3):2172.
57. Zanger DZ. Convergence of a least-squares Monte Carlo algorithm for bounded approximating sets. Appl Math Finance. 2009;16:123.
58. Gerhold S. The Longstaff–Schwartz algorithm for Levy models: results on fast and slow convergence. Ann Appl Probab. 2011;21(2):589.
59. Zanger DZ. Quantitative error estimates for a least-squares Monte Carlo algorithm for American option pricing. Finance Stoch. 2013;17(3):503.
60. Zanger DZ. Convergence of a least-squares Monte Carlo algorithm for American option pricing with dependent sample data. Math Finance. 2018;28(1):447.
61. Zanger DZ. General error estimates for the Longstaff–Schwartz least-squares Monte Carlo algorithm. Math Oper Res. 2020;45(3):923.
62. Jerrum M et al. Random generation of combinatorial structures from a uniform distribution. Theor Comput Sci. 1986;43:169.
63. Grover L, Rudolph T. Creating superpositions that correspond to efficiently integrable probability distributions. arXiv:quant-ph/0208112.
64. Haner T, et al. Optimizing quantum circuits for arithmetic. arXiv:1805.12445.
65. Wiebe N et al. Quantum data fitting. Phys Rev Lett. 2012;109:050505.
66. Schuld M et al. Prediction by linear regression on a quantum computer. Phys Rev A. 2016;94:022342.
67. Wang G. Quantum algorithm for linear regression. Phys Rev A. 2017;96:012335.

68. Yu C-H et al. Quantum algorithms for ridge regression. IEEE Trans Knowl Data Eng. 2019;29:37491.
69. Chakraborty S. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. In: Proceedings of the 46th international colloquium on automata, languages, and programming (ICALP). 2019. p. 33:1–33:14.
70. Kerenidis I, Prakash A. Quantum gradient descent for linear systems and least squares. Phys Rev A. 2020;101:022316.
71. Kaneko K, et al. Linear regression by quantum amplitude estimation and its extension to convex optimization. arXiv:2105.13511.