

# Quantum Algorithm for Stochastic Optimal Stopping Problems with Applications in Finance

João F. Doriguello ✉ 

Centre for Quantum Technologies, National University of Singapore, Singapore

Alessandro Luongo ✉

Centre for Quantum Technologies, National University of Singapore, Singapore

Jinge Bao ✉

Centre for Quantum Technologies, National University of Singapore, Singapore

Patrick Rebenrost ✉

Centre for Quantum Technologies, National University of Singapore, Singapore

Miklos Santha ✉

Centre for Quantum Technologies, National University of Singapore, Singapore

---

## Abstract

The famous least squares Monte Carlo (LSM) algorithm combines linear least square regression with Monte Carlo simulation to approximately solve problems in stochastic optimal stopping theory. In this work, we propose a quantum LSM based on quantum access to a stochastic process, on quantum circuits for computing the optimal stopping times, and on quantum techniques for Monte Carlo. For this algorithm, we elucidate the intricate interplay of function approximation and quantum algorithms for Monte Carlo. Our algorithm achieves a nearly *quadratic* speedup in the runtime compared to the LSM algorithm under some mild assumptions. Specifically, our quantum algorithm can be applied to American option pricing and we analyze a case study for the common situation of Brownian motion and geometric Brownian motion processes.

**2012 ACM Subject Classification** Mathematics of computing → Stochastic processes; Mathematics of computing → Markov-chain Monte Carlo methods; Theory of computation → Quantum computation theory

**Keywords and phrases** Quantum computation complexity, optimal stopping time, stochastic processes, American options, quantum finance

**Digital Object Identifier** 10.4230/LIPIcs.TQC.2022.2

**Related Version** *Full Version:* <https://arxiv.org/abs/2111.15332> [28]

**Funding** Research at CQT is funded by the National Research Foundation, the Prime Minister’s Office, and the Ministry of Education, Singapore under the Research Centres of Excellence programme’s research grant R-710-000-012-135.

**Acknowledgements** We thank Rajagopal Raman for pointing out Ref. [53] and the importance of the Longstaff-Schwarz algorithm in the insurance industry. We also thank Koichi Miyamoto for Ref. [50] and Eric Ghysels for Refs. [15, 16].

## 1 Introduction

Within stochastic optimization, optimal stopping theory is a broad area of applied mathematics that started in the 1940s and 1950s mainly with A. Wald [90] and is concerned with the problem of deciding the best time to “stop” or take an action in order to maximize an expected reward [81]. The time at which the observations are terminated, called *stopping time*, is a random variable depending on the data observed so far in the process. A simple



© João F. Doriguello, Alessandro Luongo, Jinge Bao, Patrick Rebenrost, and Miklos Santha; licensed under Creative Commons License CC-BY 4.0

17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022).

Editors: François Le Gall and Tomoyuki Morimae; Article No. 2; pp. 2:1–2:24

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

example of an optimal stopping problem is the following: consider a game in which 100 numbers are written on 100 pieces of paper without restrictions on the numbers, except that no number appears more than once. The pieces of paper are shuffled faced down and you are asked to look at the numbers, without having seen them before and *once* at a time, and to stop when you think that you have found the biggest number. It turns out that there is a stopping rule that allows you to stop at the biggest number for  $1/e$  fraction of the inputs.

Since its conception, optimal stopping theory collected problems from many disparate areas under a unique umbrella [71], e.g. quickest detection [82], sequential parameter estimation [64] and sequential hypothesis testing [26]. Probably the most famous optimal stopping problem is the one of option pricing in finance, especially American options [57]. A central problem in the world of finance is to assign a monetary value to a hitherto unvalued asset. In the capital markets, there exists a large variety of financial assets which are derivative to underlying assets such as stocks, bonds, or commodities. One of the most well-known examples is the European call option which allows the buyer to “lock in” a price for buying a stock at some future time (or “exercise” time). A fair valuation of such an option was first discussed in the seminal works of Black and Scholes [11] and Merton [61]. In the times since, the methods proposed in these works have become standard practice in the financial sector and have been extended and generalized for many financial derivatives and market models.

American options allow the buyer to exercise the option at any point in time between the time of purchase and a fixed final time. In contrast to an European option, there are no known closed formulas for the price of an American option with finite maturity date even in simple models like the Black-Scholes-Merton one. Theoretically, an American option can be viewed as a stochastic optimal stopping problem for the buyer and a super-martingale hedging problem for the seller [31]. Practical algorithms have been developed for the pricing of American options [23, 77, 48, 46], an important class being least squares Monte Carlo (LSM) algorithms originally proposed independently by Tsitsiklis and Van Roy [87] and by Longstaff and Schwartz [57].

Among the aforementioned classical algorithms for option pricing – and other topics in finance – is the sub-field of quantum computing of designing quantum algorithms in the context of financial problems [67, 12, 29], e.g. risk management [92], financial greeks [84, 3], portfolio optimization [6, 25, 45, 75, 42, 2] and option pricing [59, 89]. A common tool in obtaining a quantum advantage is amplitude estimation [14] and its generalizations for Monte Carlo sampling [63, 41, 22, 21]. A few different works devised quantum algorithms for derivative pricing based on quantum subroutines for Monte Carlo [74, 83, 18], e.g. European [73, 32, 72] and American/Bermudan [62] option pricing, and option pricing in the local volatility model [49, 3] (of which the Black-Scholes model is a subcase). Given its versatility and previous cases of success, it is only natural to explore the applicability of quantum methods for Monte Carlo to problems in optimal stopping theory. In this work we focus on tailoring these methods to LSM algorithms.

### Applications of LSM algorithm

Among the whole domain of optimal stopping problems, there are many that can be approached directly with LSM, e.g. the secretary problem [24], modelling the optimal time to call an election based on data [86], estimating the solvency of governments with respect to their debt [79], and multi-armed bandit problems [40]. Another important application of LSM is in the insurance sector. In fact, LSM can be used to estimate the VaR (Value at Risk) [53] and life insurance contracts [5] (see also [69] for a comparison of LSM with other methods). The computational challenges of this domain were further highlighted by recent European

regulatory requirements [1, 27]. LSM is also often used for solving Backward Stochastic Differential Equations (BSDE). Some numerical algorithms for BSDE are two-steps stochastic procedures involving a discretisation step where the solutions obtained at time  $t$  of the BSDE are projected onto a space obtained from the filtrations at time  $t - 1$ . This step involves a conditional expectation that cannot be calculated analytically, but must be estimated using some approximation procedure. The idea of applying LSM to BSDE was first introduced in [35] and further developed in [36, 54]. Recently, this method has been generalized to solve two-dimensional forward-backward stochastic differential equations [55, 8].

### Other algorithms for option pricing

LSM is not the only type of algorithm that can be used to price American options [30]. Besides a few attempts to give an analytical formula under certain conditions [51], the vast majority of them has been directed towards giving numerical results, which we briefly discuss in this section. A simple and well-known way of pricing American options is through the use of binomial trees. While the origins of this technique are somewhat unclear [19], the first articles that proposed the idea of binomial trees for pricing options are considered to be [23, 77], with the first seminal ideas proposed in the first edition of [80]. McKean [60] realized that the price of an American option can be cast as a free boundary problem [70], which is a particular partial differential equation that can be solved numerically. There is a flurry of other methods to price American options based on partial differential equations. We name a few approaches such as variational inequalities [48, 9], linear complementary [46], and those related to free boundaries [88]. However, as noted in [30], these methods often suffer from the curse of dimensionality, as they require the computing time and the storage to grow exponentially with the dimension of the underlying state space. LSM is also not the only Monte Carlo approach for pricing American options. One of the first works using Monte Carlo for option pricing is [13]. Reviews of Monte Carlo and other methods for the problem of American option pricing can be found in [33, 52, 17]. In contrast to giving lower bounds for the true optimal stopping value – as the LSM algorithm – Rogers [78] proposed a method which leverages a dual problem, resulting in an upper bound for the optimal stopping value. Last but not least, semi-analytical approaches for American option pricing and optimal stopping time are also used [7].

## 1.1 Problem statement

Optimal stopping theory is concerned with the problem of finding the best moment to stop a process in order to maximize an expected reward. More generally, assume a discrete-time stochastic process  $\mathbf{X} = (X_t)_{t=0}^T$  (which corresponds to the market model in financial applications), assumed to be Markovian, defined on a filtered probability space  $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t=0}^T, \mathbb{P})$  and with state space  $(E, \mathcal{E})$ , where  $E \subseteq \mathbb{R}^d$ . We shall assume that  $\mathbf{X}$  is *adapted* with respect to  $(\mathcal{F}_t)_{t=0}^T$ , meaning that each  $X_t$  is  $\mathcal{F}_t$ -measurable, and that  $X_0 = x_0$  is deterministic, therefore, sometimes we write the Markov chain  $(X_t)_{t=1}^T$  as starting from  $t = 1$ . Each element  $X_t$  for  $t \in \{0, \dots, T\}$ , called the underlying process at time  $t$ , gives rise to an image probability measure (also called pushforward measure)  $\rho_t$  in  $E \subseteq \mathbb{R}^d$ , i.e.,  $\rho_t(Y) = \mathbb{P}[\omega \in \Omega : X_t(\omega) \in Y]$  for any  $Y \in \mathcal{E}$  (note that  $\rho_0$  is the probability measure that assigns measure 1 to the singleton set containing  $x_0$ ). We denote by  $\mathbf{X}_t = (X_j)_{j=t}^T$  the last  $T - t + 1$  random variables in the stochastic process. Let  $L^2(E, \rho_t) = L^2(\rho_t)$  be the set of squared integrable functions with norm  $\|f\|_{L^2(\rho_t)} := \sqrt{\mathbb{E}_{\rho_t}[|f(X_t)|^2]}$  and define the uniform norm  $\|f\|_u = \sup\{|f(s)| : s \in E\}$  for  $f : E \rightarrow \mathbb{R}$ . Consider further a *payoff process*: a non-negative adapted process  $(Z_t)_{t=0}^T$

on the filtered probability space  $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t=0}^T, \mathbb{P})$  obtained from  $(X_t)_{t=0}^T$  as  $Z_t = z_t(X_t)$  for square-integrable real functions  $z_t \in L^2(E, \rho_t)$ , for  $t \in \{0, \dots, T\}$ . Moreover, given an event  $\mathcal{E}$ , we denote by  $\mathbf{1}\{\mathcal{E}\}$  the indicator function  $\mathbf{1}\{\mathcal{E}\} = 1$  if  $\mathcal{E}$  is true and 0 if not. Finally, we shall use  $\mathbb{P}$  to denote the probability measure behind the stochastic process  $(X_t)_{t=0}^T$ , and the probability notation  $\Pr$  for other processes, e.g. the outcome probability of a quantum measurement.

Our main problem is when to stop the process and take the payoff so that the expected payoff is maximized. This is formalized by the idea of *stopping time*, which is a random variable that selects one of the possible times  $\{0, 1, \dots, T\} \cup \{+\infty\}$  and satisfies a measurability condition.

► **Definition 1** (Stopping time). *A stopping time is a function  $\tau : \Omega \rightarrow \{0, 1, \dots, T\} \cup \{+\infty\}$  such that  $\{\omega \in \Omega \mid \tau(\omega) = t\} \in \mathcal{F}_t$  for  $t \in \{0, \dots, T\}$ . The payoff obtained by using  $\tau$  is  $Z_\tau(\omega) := Z_{\tau(\omega)}(\omega)$ . Let  $\mathbb{T}_t := \{\tau \mid \tau \text{ is a stopping time with } t \leq \tau \leq T\}$  be the set of all stopping times taking values in  $[t, T]$ . A stopping time  $\tau^* \in \mathbb{T}_t$  is called optimal in the interval  $[t, T]$  if*

$$\mathbb{E}[Z_{\tau^*} | X_t] = \operatorname{ess\,sup}_{\tau \in \mathbb{T}_t} \mathbb{E}[Z_\tau | X_t].$$

The maximization is expressed via an essential supremum such that the null sets of the probability measure do not affect the result. For more details on the essential supremum see [31, Appendix A.5]. The optimal stopping problem then consists in finding a stopping time  $\tau$  that maximizes the *expected* value payoff.

► **Problem 2** (Optimal stopping problem). *Let  $(Z_t)_{t=0}^T$  be a payoff process. For  $\epsilon_{\text{final}} > 0$ , approximate the exact value  $\sup_{\tau \in \mathbb{T}_0} \mathbb{E}[Z_\tau]$  to additive accuracy  $\epsilon_{\text{final}}$  with high probability.*

A well-studied solution strategy for the above problem statement is based on dynamic programming for a set of stopping times. A crucial concept is the *Snell envelope* [65, 31]  $\mathcal{U}_t : \Omega \rightarrow \mathbb{R}$  of a payoff process  $Z_t = z_t(X_t)$  (for some  $z_t \in L^2(E, \rho_t)$ ,  $t \in \{0, \dots, T\}$ ) defined as

$$\begin{cases} \mathcal{U}_T = Z_T, \\ \mathcal{U}_t = \max \{Z_t, \mathbb{E}[\mathcal{U}_{t+1} | X_t]\}, \quad 0 \leq t \leq T-1. \end{cases} \quad (1)$$

Define the stopping times  $\tau_t := \min\{u \geq t \mid \mathcal{U}_u = Z_u\}$ . The Snell envelope is related to the maximal expected payoff according to the next theorem:  $\tau_t$  maximizes the expectation of  $Z_\tau$  among all  $\tau \in \mathbb{T}_t$ , i.e., that  $\tau_t$  are optimal stopping times (in their respective intervals).

► **Theorem 3** ([31, Theorem 6.18]). *The Snell envelope  $\mathcal{U}_t$  of  $Z_t$  satisfies*

$$\mathcal{U}_t = \mathbb{E}[Z_{\tau_t} | X_t] = \operatorname{ess\,sup}_{\tau \in \mathbb{T}_t} \mathbb{E}[Z_\tau | X_t].$$

In particular,  $\mathcal{U}_0 = \mathbb{E}[Z_{\tau_0}] = \sup_{\tau \in \mathbb{T}_0} \mathbb{E}[Z_\tau] = \max\{Z_0, \mathbb{E}[Z_{\tau_1}]\}$ .

Hence, finding an approximate  $\mathcal{U}_0$  solves our Problem 2. In order to solve the dynamic programming behind the Snell envelope, it is more convenient to recast the dynamic programming in terms of the optimal stopping times  $\tau_t$  (rather than in terms of value functions) as follows.

► **Theorem 4.** *The dynamic programming principle in Eq. (1) can be recast in terms of the stopping times  $\tau_t = \min\{u \geq t \mid \mathcal{U}_u = Z_u\}$  as*

$$\begin{cases} \tau_T = T, \\ \tau_t = t \mathbf{1}\{Z_t \geq \mathbb{E}[Z_{\tau_{t+1}} | X_t]\} + \tau_{t+1} \mathbf{1}\{Z_t < \mathbb{E}[Z_{\tau_{t+1}} | X_t]\}, \quad 0 \leq t \leq T-1. \end{cases}$$

**Proof.** The case  $t = T$  is trivial. Assume  $t < T$ . Note that  $\mathbb{E}[Z_{\tau_{t+1}}|X_t] = \mathbb{E}[\mathbb{E}[Z_{\tau_{t+1}}|X_{t+1}]|X_t]$  because of the tower property of the expectation value with the filtration generated by  $X_t$ . In addition,  $\mathbb{E}[Z_{\tau_{t+1}}|X_{t+1}] = \mathcal{U}_{t+1}$  from Theorem 3. Hence, if  $Z_t \geq \mathbb{E}[Z_{\tau_{t+1}}|X_t]$ , then  $Z_t \geq \mathbb{E}[\mathcal{U}_{t+1}|X_t]$ . This latter statement, by the definition of the Snell envelope, implies  $\mathcal{U}_t = Z_t$  and then  $\tau_t = t$ . On the other hand, if  $Z_t < \mathbb{E}[Z_{\tau_{t+1}}|X_t]$ , then,  $Z_t < \mathbb{E}[\mathcal{U}_{t+1}|X_t] \implies Z_t \neq \mathcal{U}_t$ , and so  $\tau_t = \min\{u \geq t \mid Z_u = \mathcal{U}_u\} = \min\{u \geq t + 1 \mid Z_u = \mathcal{U}_u\} = \tau_{t+1}$ .  $\blacktriangleleft$

The stopping time  $\tau_0$  thus maximizes  $\mathbb{E}[Z_\tau]$  in Problem 2. The quantities  $\mathbb{E}[Z_{\tau_{t+1}}|X_t]$  are called *continuation values*. In the past, many different approaches were developed to tackle the dynamic programming above [51, 19, 23, 77, 80, 60, 70, 48, 9, 46, 88, 13, 33, 52, 17]. A famous approach is the least squares Monte Carlo (LSM) by Longstaff and Schwartz [57].

## 2 The least squares Monte Carlo algorithm

The LSM algorithm consists in solving the dynamic programming in Theorem 4 by means of two approximations. The first one is to approximate the continuation values  $\mathbb{E}[Z_{\tau_{t+1}}|X_t]$  using a set of measurable real-valued functions in  $L^2(E, \rho_t)$ , e.g. by projection onto a finite-dimensional set of linearly independent polynomials. Let  $\mathcal{H}_0 \subseteq \mathbb{R}$  and let, for  $t \in [T - 1]$ ,  $\mathcal{H}_t \subseteq L^2(E, \rho_t)$  be a subset of real-valued functions on  $E$ , called *approximation architecture* or hypothesis class, that will be used to approximate the continuation values. By approximating  $\mathbb{E}[Z_{\tau_{t+1}}|X_t]$  by  $f_t \in \mathcal{H}_t$  for each  $t \in \{0, \dots, T - 1\}$ , we can write the approximate dynamic programming as

$$\begin{cases} \tilde{\tau}_T = T, \\ \tilde{\tau}_t = t\mathbf{1}\{Z_t \geq f_t\} + \tilde{\tau}_{t+1}\mathbf{1}\{Z_t < f_t\}, \quad 0 \leq t \leq T - 1. \end{cases} \quad (2)$$

Note that  $\tilde{\tau}_t = \tilde{\tau}_t(f_t, \dots, f_{T-1})$  depends on the approximation architecture.

The second approximation of the algorithm is to numerically evaluate the approximations  $f_t$  in  $L^2(\rho_t)$  by a Monte Carlo procedure. We sample  $N$  independent paths  $(X_t^{(1)})_{t=0}^T, \dots, (X_t^{(N)})_{t=0}^T$  of the Markov chain  $\mathbf{X} = (X_t)_{t=0}^T$  and denote by  $Z_t^{(n)} = z_t(X_t^{(n)})$  the associated payoffs conditioned on  $X_t^{(n)}$ , where  $z_t \in L^2(E, \rho_t)$ ,  $t \in \{0, \dots, T\}$ . Write the random variables of the last  $T - t + 1$  elements of all the sampled Markov chains by  $\mathbf{X}_t^{(N)} = (X_t^{(1)}, \dots, X_t^{(N)}, X_{t+1}^{(1)}, \dots, X_{t+1}^{(N)}, \dots, X_T^{(1)}, \dots, X_T^{(N)})$ . For each path, the dynamic programming in Eq. (2) is solved recursively by approximating the continuation values in  $\mathcal{H}_t$  via a least square estimator. The result is sampled stopping times  $\tilde{\tau}_t^{(n)}$  that  $\tilde{\tau}_t$  takes on each random path. We stress that, due to the recursive nature of Eq. (2), the stopping times  $\tilde{\tau}_t^{(n)}$  will depend on  $\mathbf{X}_t^{(N)}$ , and consequently also the payoffs  $Z_{\tilde{\tau}_t^{(n)}}^{(n)} = z_{\tilde{\tau}_t^{(n)}}(\mathbf{X}_t^{(N)})(X_{\tilde{\tau}_t^{(n)}}^{(n)}(\mathbf{X}_t^{(N)}))$ .

The dependence on  $\mathbf{X}_t^{(N)}$  should be clear from the context and therefore we shall simply write  $Z_{\tilde{\tau}_t^{(n)}}^{(n)}$ . In summary, combining both the approximation architecture and the Monte Carlo sampling, at each  $t \in [T - 1]$  we take  $f_t \in \mathcal{H}_t$ , depending on  $\mathbf{X}_t^{(N)}$ , satisfying

$$\frac{1}{N} \sum_{n=1}^N \left( Z_{\tilde{\tau}_{t+1}^{(n)}}^{(n)} - f_t(X_t^{(n)}) \right)^2 \leq \epsilon + \inf_{g \in \mathcal{H}_t} \frac{1}{N} \sum_{n=1}^N \left( Z_{\tilde{\tau}_{t+1}^{(n)}}^{(n)} - g(X_t^{(n)}) \right)^2 \quad (3)$$

for some given  $\epsilon \geq 0$ . It might be the case that an exact minimizer of the above optimization problem does not exist (the infimum does not belong to  $\mathcal{H}_t$ ) or is hard to compute, meaning that an  $\epsilon$ -approximation could be used. Given the choice of  $f_t \in \mathcal{H}_t$ , it is then used

in Eq. (2) to obtain  $\tilde{\tau}_t^{(n)}$ , and so on recursively. At the end of the recursion we can take  $f_0 = \frac{1}{N} \sum_{n=1}^N Z_{\tilde{\tau}_1^{(n)}}^{(n)}$  as an exact minimizer, since  $X_0$  is constant, and obtain the approximation  $\tilde{\mathcal{U}}_0$  to  $\mathcal{U}_0$  as

$$\tilde{\mathcal{U}}_0 = \max \left\{ Z_0, \frac{1}{N} \sum_{n=1}^N Z_{\tilde{\tau}_1^{(n)}}^{(n)} \right\}. \quad (4)$$

In this paper we shall be particularly interested in finite-dimensional linear approximation architectures, for which an exact minimizer exists in Eq. (3). Consider then a set  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  of  $m$  linearly independent measurable real functions and take the vector space generated by them as our approximation architecture  $\mathcal{H}_t$ ,  $t \in [T-1]$ . Therefore, the infimum in Eq. (3) is attained by projecting the continuation values onto  $\mathcal{H}_t$  as  $\alpha_t \cdot \vec{e}_t(X_t)$ , where  $\vec{e}_t(\cdot) := (e_{t,1}(\cdot), \dots, e_{t,m}(\cdot))^\top$  and the  $m$ -dimensional vector  $\alpha_t$ , the projection coefficients, is the least square estimator given by [20]

$$\alpha_t = \arg \min_{a \in \mathbb{R}^m} \mathbb{E}[(Z_{\tilde{\tau}_{t+1}} - a \cdot \vec{e}_t(X_t))^2].$$

Given the assumption that  $\{e_{t,k}\}_{k=1}^m$  are linearly independent for each  $t \in [T-1]$ , the vector  $\alpha_t \in \mathbb{R}^m$  has the explicit expression

$$\alpha_t = A_t^{-1} b_t \text{ where } b_t = \mathbb{E}[Z_{\tilde{\tau}_{t+1}} \vec{e}_t(X_t)] \quad (5)$$

and the  $m \times m$  matrix  $A_t$  has coefficients

$$(A_t)_{k,l} = \mathbb{E}[e_{t,k}(X_t)e_{t,l}(X_t)]. \quad (6)$$

Often it is hard to compute  $\alpha_t$  and  $A_t$  exactly. As previously mentioned, the LSM algorithm approximates these by Monte Carlo sampling,

$$\tilde{\alpha}_t = \tilde{A}_t^{-1} \frac{1}{N} \sum_{n=1}^N Z_{\tilde{\tau}_{t+1}^{(n)}} \vec{e}_t(X_t^{(n)}) \quad (7)$$

and

$$(\tilde{A}_t)_{k,l} = \frac{1}{N} \sum_{n=1}^N e_{t,k}(X_t^{(n)})e_{t,l}(X_t^{(n)}). \quad (8)$$

More generally, though, any good approximation  $\tilde{\alpha}_t$  and  $\tilde{A}_t$  to  $\alpha_t$  and  $A_t$ , respectively, is valid, and we shall not restrict the notation  $\tilde{\alpha}_t$  and  $\tilde{A}_t$  to only mean the above sampled quantities.

We have introduced the quantities that are important for the LSM algorithm. To present the algorithm, we first specify the input model.

► **Definition 5** (Sampling access to Markov chain). *Given a Markov chain  $(X_t)_{t=1}^T$  on a probability space  $(\Omega, \mathbb{P})$  and with state space  $E \subseteq \mathbb{R}^d$ , we define sampling access as the ability to draw a sample  $\omega \in \Omega$  according to  $\mathbb{P}$  and observe the value  $X_t(\omega)$  for some  $t \in [T]$ . One sample costs time  $\mathcal{T}_{\text{samp}}$ .*

► **Definition 6** (Query access to function). *Let  $E \subseteq \mathbb{R}^d$  and  $h : E \rightarrow \mathbb{R}$  be a function. We define query access as the ability to observe the value  $h(x)$  for any given  $x \in E$ . One query costs time  $\mathcal{T}_h$ .*

■ **Algorithm 1** Classical LSM algorithm for optimal stopping problem.

---

**Input:** Integer  $N \in \mathbb{N}$ . Sampling access to Markov chain  $(X_t)_{t=0}^T$  defined on a sample space  $\Omega$  and with state space  $E \subseteq \mathbb{R}^d$ . Query access to functions  $\{z_t : E \rightarrow \mathbb{R}\}_{t=0}^T$  and  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{t \in [T-1], k \in [m]}$ , where  $\{e_{t,k}\}_{k=1}^m$  are linearly independent for each  $t \in [T-1]$ . Let  $\vec{e}_t(\cdot) := (e_{t,1}(\cdot), \dots, e_{t,m}(\cdot))^\top$ .

- 1: Sample  $N$  independent paths  $(X_t^{(1)}, \dots, X_t^{(N)})_{t=0}^T$ .
- 2: Query payoffs  $(Z_t^{(1)}, \dots, Z_t^{(N)})_{t=0}^T$  and values  $(e_{t,k}(X_t^{(1)}), \dots, e_{t,k}(X_t^{(N)}))_{t \in [T-1], k \in [m]}$ .
- 3: Compute the matrices  $\{\tilde{A}_t\}_{t=1}^{T-1}$  with entries as in Eq. (8).
- 4: Compute the inverses  $\{\tilde{A}_t^{-1}\}_{t=1}^{T-1}$ .
- 5: Set  $\tilde{\tau}_T^{(n)} = T$  for  $n \in [N]$ .
- 6: **for**  $t = T - 1$  to  $1$  **do**
- 7:     Calculate the vector  $\tilde{\alpha}_t = \tilde{A}_t^{-1} \frac{1}{N} \sum_{n=1}^N Z_{\tilde{\tau}_{t+1}^{(n)}}^{(n)} \vec{e}_t(X_t^{(n)})$ .
- 8:     Calculate  $\tilde{\tau}_t^{(n)} = t \mathbf{1}\{Z_t^{(n)} \geq \tilde{\alpha}_t \cdot \vec{e}_t(X_t^{(n)})\} + \tilde{\tau}_{t+1}^{(n)} \mathbf{1}\{Z_t^{(n)} < \tilde{\alpha}_t \cdot \vec{e}_t(X_t^{(n)})\}$ ,  $n \in [N]$ .
- 9: **end for**
- 10: Output  $\tilde{U}_0 := \max \left\{ Z_0, \frac{1}{N} \sum_{n=1}^N Z_{\tilde{\tau}_1^{(n)}}^{(n)} \right\}$ .

---

Here, we assume that the functions of the approximation architecture and functions for the payoff take time  $\mathcal{T}_e$  and  $\mathcal{T}_z$ , respectively, to evaluate. Both sampling and function access have natural quantum extensions, as will be defined in Section 3.

We are now in the position to present the classical LSM algorithm in Algorithm 1. Since we focus on the case where the approximation architectures  $\mathcal{H}_t$  are finite-dimensional and linear, we write Algorithm 1 for this particular case.

### 3 Quantum least squares Monte Carlo algorithm

In this section we shall present our quantum algorithm, which is based on the classical LSM algorithm (Algorithm 1). Before we discuss it, we review our computational model, input assumptions, and the quantum algorithm for Monte Carlo used in this work. In what follows, for simplicity, we suppose that  $|0\rangle$  describes a register with sufficiently many qubits initialized in the all-0 state.

#### 3.1 Computational model

In this subsection, we address our quantum computational model. We work in the standard circuit model of quantum computation [66]. Aside from these standard assumptions, we take the following additional assumptions on the computational model.

##### Arithmetic model

In our work, we perform the arithmetic computations on the quantum computer by using a fixed point representation for real numbers. We assume that we can have enough qubits for storing these numbers, represented as bit strings using the following definition. We also assume to work with enough precision so that numerical errors in the computation are negligible, and will not impact the final output of our algorithm.



► **Definition 7** (Fixed-point encoding of real numbers [76]). *Let  $c_1, c_2$  be positive integers, and  $a \in \{0, 1\}^{c_1}$ ,  $b \in \{0, 1\}^{c_2}$ , and  $s \in \{0, 1\}$  be bit strings. Define the rational number as:*

$$\mathcal{Q}(a, b, s) := (-1)^s \left( 2^{c_1-1} a_{c_1} + \cdots + 2a_2 + a_1 + \frac{1}{2}b_1 + \cdots + \frac{1}{2^{c_2}}b_{c_2} \right) \in [-R, R], \quad (9)$$

where  $R := 2^{c_1} - 2^{-c_2}$ . If  $c_1, c_2$  are clear from the context, we can use the shorthand notation for a number  $z := (a, b, s)$  and write  $\mathcal{Q}(z)$  instead of  $\mathcal{Q}(a, b, s)$ . Given an  $n$ -dimensional vector  $v \in (\{0, 1\}^{c_1} \times \{0, 1\}^{c_2} \times \{0, 1\})^n$  the notation  $\mathcal{Q}(v)$  means an  $n$ -dimensional vector whose  $j$ -th component is  $\mathcal{Q}(v_j)$ , for  $j \in [n]$ .

The choice of values for  $c_1$  and  $c_2$  depends on the choice of input functions used when running the algorithm. For the purposes of optimizing the quantum circuit, these constants can be changed dynamically in various steps of the computation. While analyzing how error propagates and accumulates throughout the operations in the quantum circuit is essential to ensure a correct estimation of the final result, this analysis can only be done for a given choice of input functions. We avoid the analysis of such details by using the quantum arithmetic model as in Definition 8. A standard result is that any Boolean function can be reversibly computed. Any reversible computation can be realized with a circuit involving negation and three-bit Toffoli gates. Such a circuit can be turned into a quantum circuit with single-qubit NOT gates and three-qubit Toffoli gates. Since most circuits for arithmetic operations operate with a number of gates of  $O(\text{poly}(c_1, c_2))$  this implies a number of quantum gates of  $O(\text{poly}(c_1, c_2))$  for the corresponding quantum circuit.

► **Definition 8** (Quantum arithmetic model). *Given  $c_1, c_2 \in \mathbb{N}$  specifying fixed-point precision numbers as in Definition 7, we say we use a quantum arithmetic model of computation if the four arithmetic operations can be performed in constant time in a quantum computer.*

In our computational model we do not include the cost for performing operations described in our arithmetic model. For instance, a central computational step of the quantum algorithm is the circuit computing the stopping times  $\tilde{\tau}_t(x)$ , but as the circuit depth depends polynomially on  $c_1$  and  $c_2$ , we do not take into account this cost when stating our runtime. For an example of a resource estimation for a financial problem that takes into account the cost of arithmetic operations in fixed-point precision, we refer to [18].

### Quantum input access

We assume that we have quantum oracles for certain input functions. The classical algorithm assumes access to two different kinds of oracles. The first is an oracle that allows us to obtain samples from the Markov chain  $(X_t)_{t=0}^T$ . The second kind of oracle is evaluating the functions  $\{z_t\}_{t=0}^T$  and  $\{e_{t,k}\}_{t \in [T-1], k \in [m]}$ . We assume access to the quantum versions of these oracles (formalized below). The first kind of quantum oracle prepares a quantum state that is in a superposition over the different outcomes of the Markov chain, weighted by amplitudes which are square roots of their classical probabilities. A measurement in the computational basis of such a state obtains a single sample with the corresponding probability and hence directly recovers a single use of the classical sampling access. The second kind of quantum oracle evaluates a given function in superposition over its inputs. While the functions  $\{e_{t,k}\}_{t \in [T-1], k \in [m]}$  are usually chosen to be low-degree polynomials (and thus admit efficient classical and quantum circuits with gate complexity proportional to the degree of the polynomial), the functions  $\{z_t\}_{t=0}^T$  might be arbitrarily complex. Usually the complexity of these functions is not discussed in classical literature, and we use placeholders for their evaluation cost.



► **Definition 9** (Quantum sampling access to a Markov chain). *Let  $(X_t)_{t=1}^T$  be a Markov chain defined on a filtered probability space  $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t=1}^T, \mathbb{P})$ , for a finite  $\Omega$ , assuming values in a finite state space  $E \subseteq \mathbb{R}^d$ . Given  $x \in E^T$ , let  $p(x) := \mathbb{P}[X_1 = x_1] \prod_{t=1}^{T-1} \mathbb{P}[X_{t+1} = x_{t+1} | X_t = x_t]$  be the probability that  $X_1 = x_1, \dots, X_T = x_T$ . Let  $\mathcal{H}$  be a finite-dimensional Hilbert space with basis  $\{|x\rangle\}_{x \in E^T}$ . We say that we have quantum sampling access to  $(X_t)_{t=1}^T$  if we are given an oracle  $U$  on  $\mathcal{H}$  such that  $U|0\rangle = \sum_{x \in E^T} \sqrt{p(x)}|x\rangle$ . One application of  $U$  costs  $T \cdot \mathcal{T}_{\text{samp}}$  time. If  $T = 1$ , we say that we have quantum sampling access to a random variable  $X$  if we are given an oracle  $U$  on  $\mathcal{H}$  such that  $U|0\rangle = \sum_{x \in E} \sqrt{p(x)}|x\rangle$ , where  $p(x) := \mathbb{P}[X = x]$  is the probability that  $X = x$ .*

We note the alternative definition of the unitary  $U$  such that  $U|0\rangle = \sum_{x \in E^T} \sqrt{p(x)}|x\rangle|\psi_x\rangle$  for unknown garbage unit states  $|\psi_x\rangle$ . Such garbage states do not change our analysis, so we shall ignore them and work with the unitary  $U$  from Definition 9.

Even though we assume the existence of the oracle  $U$ , constructing such unitary is an important question on its own. A few methods have been proposed in order to tackle such problem, one of the most famous is due to Grover and Rudolph [39] (see [58] for recent improvements on the Grover-Rudolph method), which loads into a quantum computer a discretization of a distribution with density function  $p(x)$ . More specifically, it creates the quantum state  $\sum_{i=0}^{2^n-1} \sqrt{p_i^{(n)}}|i\rangle$  with  $p_i^{(n)} = \int_{x_i^{(n)}}^{x_{i+1}^{(n)}} p(x)dx$  by recursively (on  $n$ ) computing quantities like  $f_n(i) = \int_{x_i^{(n)}}^{(x_i^{(n)}+x_{i+1}^{(n)})/2} p(x)dx / \int_{x_i^{(n)}}^{x_{i+1}^{(n)}} p(x)dx$  for  $i = 0, \dots, 2^n - 1$ . It is also possible to perform simple Taylor approximations on  $f_n(i)$  when  $n$  is sufficiently large (see [49, Equation (35)]). We briefly note that the issues about the Grover-Rudolph method recently pointed out by [44] only arise when one needs to *sample* from the distribution  $p(x)$  in order to compute  $f_n(i)$ , which is not the case in many situations, e.g. in finance.

► **Definition 10** (Quantum access to a function). *Let  $E \subseteq \mathbb{R}^d$  be a finite set and let  $\mathcal{H}$  be a Hilbert space with basis  $\{|x\rangle\}_{x \in E}$ . Given  $h : E \rightarrow \{0, 1\}^n$ , we say that we have  $(V_h, \mathcal{T}_h)$ -quantum access to  $h$  if we have access to a quantum circuit  $V_h$  on  $\mathcal{H} \otimes \mathbb{C}^{2^n}$  such that  $V_h|x\rangle|b\rangle = |x\rangle|b \oplus h(x)\rangle$  for any bit string  $b \in \{0, 1\}^n$ . One application of  $V_h$  costs time  $\mathcal{T}_h$ .*

### Access to quantum controlled rotations

Controlled rotations are a central step in the quantum algorithm for Monte Carlo (Theorem 13). The cost of a controlled rotation depends directly on the number of bits used to specify the angle of rotation [91]. In our computational model we assume that controlled rotations come with a unit cost.

► **Definition 11** (Access to quantum controlled rotations). *We say that we have access to quantum controlled rotations if we have a quantum circuit  $R$  whose application takes constant time and, for all rational numbers  $x \in [0, 1]$  defined by a  $(1 + c_2)$ -bit string in our fixed-point arithmetic model 8, operates as:*

$$R|x\rangle|0\rangle = |x\rangle(\sqrt{1-x}|0\rangle + \sqrt{x}|1\rangle). \tag{10}$$

We note that again this definition allows us to neglect terms  $O(c_1 + c_2)$  in the runtime and to neglect complications arising from the arithmetic computation of the arcsin. The access to these rotation unitaries leads to the following fact.

► **Fact 12** (Controlled rotations of a function with an interval). *Consider a rational number representation from Definition 7 for some  $c_1, c_2 \in \mathbb{N}$ . Assume access to controlled rotations according to Definition 11. Assume  $(V_h, \mathcal{T}_h)$ -quantum access to a function  $h$  according to Definition 10. For any two bit strings  $a, b \in \{0, 1\}^n$ , with  $0 \leq \mathcal{Q}(a) < \mathcal{Q}(b)$ , we can construct a unitary operator  $R_{a,b}^h$  on  $\mathcal{H} \otimes \mathbb{C}^2$ , such that, for all  $x \in E$ ,*

$$R_{a,b}^h|x, 0\rangle = \begin{cases} |x\rangle \left( \sqrt{1 - \frac{\mathcal{Q}(h(x))}{\mathcal{Q}(b)}}|0\rangle + \sqrt{\frac{\mathcal{Q}(h(x))}{\mathcal{Q}(b)}}|1\rangle \right) & \text{if } \mathcal{Q}(a) \leq \mathcal{Q}(h(x)) \leq \mathcal{Q}(b), \\ |x, 0\rangle & \text{otherwise,} \end{cases}$$

where an application of  $R_{a,b}^h$  costs  $O(\mathcal{T}_h)$  time.

**Proof.** The quantum circuits for the division and for checking the interval run in constant time in the quantum arithmetic model. The quantum circuits allow us to prepare  $|x\rangle |\mathcal{Q}(h(x))/\mathcal{Q}(b)\rangle$  on the interval using  $V_h$  two times, where the second register is of size polynomial in  $c_1$  and  $c_2$ . Performing a controlled rotation of an ancilla costs constant time by Definition 11. ◀

### Quantum algorithm for Monte Carlo

Our quantum algorithm requires the computation of several expectation values. In this work we use the quantum algorithm for Monte Carlo from Montanaro [63], already adapted to our computational model.

► **Theorem 13** (Quantum algorithm for Monte Carlo `QMonteCarlo`, [63, Theorem 2.5]). *Let  $X$  be a random variable given via quantum sampling access as in Definition 9. Consider a rational number representation from Definition 7 for some  $c_1, c_2 \in \mathbb{N}$ . Let  $E \subseteq \mathbb{R}^d$  be a finite set and let  $\mathcal{H}$  be a Hilbert space with basis  $\{|x\rangle\}_{x \in E}$ . Consider a function  $h : E \rightarrow \{0, 1\}^n$  via quantum access to the controlled rotations as in Fact 12, where  $n \in \mathbb{N}$  such that  $c_1 + c_2 + 1 = n$ , and each access costs time  $\mathcal{T}_h$ . Assume that the random variable  $\mathcal{Q}(h(X))$  has finite mean  $\mu$ , and variance upper-bounded by  $\sigma^2$  for some known  $\sigma > 0$ . Given  $\delta, \epsilon \in (0, 1)$ , there is a quantum algorithm, called `QMonteCarlo` $(h(X), \epsilon, \delta, \sigma)$ , that runs in  $O((\sigma/\epsilon) \log(1/\delta) \log^{3/2}(\sigma/\epsilon) \log \log(\sigma/\epsilon)) \times (T\mathcal{T}_{\text{samp}} + \mathcal{T}_h)$  time and outputs an estimate  $\tilde{\mu}$  such that  $\Pr[|\tilde{\mu} - \mu| \geq \epsilon] \leq \delta$ .*

The above result will be used to approximate expected values, e.g.  $\mathbb{E}[e_{t,k}(X_t)e_{t,l}(X_t)]$  and  $\mathbb{E}[Z_{\tilde{\tau}_{t+1}} \tilde{e}_t(X_t)]$ , and was chosen for its simplicity. It is possible, though, to use other, more complicated quantum subroutines for Monte Carlo, e.g. [41, 22, 21]. Refs. [22, 21] propose quantum algorithms for multivariate Monte Carlo estimation, which could be particularly suitable in our case, since most of our quantities of interest are vectors and matrices. However, since these more complex and alternative quantum subroutines for Monte Carlo lead to the same time complexities up to polylogarithm factors as Theorem 13, we decided to use the above result.

### Quantum circuits for the stopping times

Recall that Theorem 4 allows us to formulate the stochastic optimal stopping problem with dynamic programming for the optimal stopping times. Having introduced the quantum computational model, we are now in the position to construct quantum circuits for various computations related to the dynamic programming. In particular, we construct a unitary

that propagates backwards the optimal stopping time by one time step according to Eq. (2). In what follows, given a path  $x \in E^T$ , by  $z_{\tilde{\tau}_t(x)}$  we mean  $z_{\tilde{\tau}_t(x)}(x_{\tilde{\tau}_t(x)})$ , i.e., the associated payoff of the  $\tilde{\tau}_t(x)$ -th time step of  $x$ .

► **Lemma 14** (Quantum circuits for computing the stopping times). *Let  $\tilde{\tau}_{t'}$ ,  $t' \in \{0, \dots, T\}$ , be stopping times defined in Eq. (2). For all  $t \in [T]$ , given quantum query access to functions  $\{z_{t'} : E \rightarrow \mathbb{R}\}_{t'=1}^T$  and  $\{e_{t',k} : E \rightarrow \mathbb{R}\}_{t' \in [T-1], k \in [m]}$  in time  $\mathcal{T}_z$  and  $\mathcal{T}_e$ , respectively, and to the components of real vectors  $\{\tilde{\alpha}_{t'} \in \mathbb{R}^m\}_{t'=t}^T$  in time  $O(1)$ , the following statements are true (let  $e_{0,k}(x_0) := 1$ ):*

1. *There is a unitary  $W_t$  such that, in time  $O(\mathcal{T}_z + m\mathcal{T}_e)$ ,*

$$\begin{cases} W_t |x\rangle |\tilde{\tau}_{t+1}(x)\rangle |\mathbf{0}\rangle^{\otimes 3} = |x\rangle |\tilde{\tau}_{t+1}(x)\rangle |z_t(x_t)\rangle |\tilde{\alpha}_t \cdot \vec{e}_t(x_t)\rangle |\tilde{\tau}_t(x)\rangle & \text{if } t \neq T, \\ W_t |x\rangle |\mathbf{0}\rangle = |x\rangle |T\rangle & \text{if } t = T. \end{cases}$$

2. *There is a unitary  $V_t^{(k)}$  such that  $V_t^{(k)} |x\rangle |\tilde{\tau}_t(x)\rangle |\mathbf{0}\rangle = |x\rangle |\tilde{\tau}_t(x)\rangle |z_{\tilde{\tau}_t(x)} e_{t-1,k}(x_{t-1})\rangle$ , for  $k \in [m]$ , in time  $O(T \log(T) \mathcal{T}_z + \mathcal{T}_e)$ .*

3. *The unitary  $C_t^{(k)} := W_T^\dagger \dots W_{t+1}^\dagger W_t^\dagger V_t^{(k)} W_t W_{t+1} \dots W_T$  is such that*

$$C_t^{(k)} |x\rangle |\mathbf{0}\rangle^{\otimes (3(T-t)+2)} = |x\rangle |z_{\tilde{\tau}_t(x)} e_{t-1,k}(x_{t-1})\rangle |\mathbf{0}\rangle^{\otimes (3(T-t)+1)}, \text{ for } k \in [m], \text{ in time } O(T(\log(T) \mathcal{T}_z + m\mathcal{T}_e)).$$

**Proof.** We start with the first statement. The existence of  $W_T$  is trivial. Assume  $t \in [T-1]$ , then, with one query access to function oracle  $z_t$  and  $m$  query accesses to function oracle  $e_{t,k}$  for  $k \in [m]$ , we can perform

$$|x\rangle |\mathbf{0}\rangle |\mathbf{0}\rangle \mapsto |x\rangle |z_t(x_t)\rangle |\mathbf{0}\rangle \mapsto |x\rangle |z_t(x_t)\rangle |\vec{e}_t(x_t)\rangle.$$

By using access to the  $m$  elements of  $\tilde{\alpha}_t$ , and  $O(m)$  multiplications and additions, we can compute the inner product of  $\tilde{\alpha}_t \cdot \vec{e}_t(x_t)$  in superposition over  $x_t$ , as

$$|x\rangle |\vec{e}_t(x_t)\rangle |\mathbf{0}\rangle \mapsto |x\rangle |\vec{e}_t(x_t)\rangle |\tilde{\alpha}_t \cdot \vec{e}_t(x_t)\rangle.$$

Comparing between  $z_t(x_t)$  and  $\tilde{\alpha}_t \cdot \vec{e}_t(x_t)$  in constant time, we can compute  $\tilde{\tau}_t(x)$  according to Eq. (2), and hence obtain

$$|x\rangle |\tilde{\tau}_{t+1}(x)\rangle |z_t(x_t)\rangle |\tilde{\alpha}_t \cdot \vec{e}_t(x_t)\rangle |\mathbf{0}\rangle \mapsto |x\rangle |\tilde{\tau}_{t+1}(x)\rangle |z_t(x_t)\rangle |\tilde{\alpha}_t \cdot \vec{e}_t(x_t)\rangle |\tilde{\tau}_t(x)\rangle.$$

Uncomputing the intermediate steps leads to the desired operation. The total runtime is  $O(\mathcal{T}_z + m\mathcal{T}_e + m + m + 1) = O(\mathcal{T}_z + m\mathcal{T}_e)$ .

Regarding the second statement, we require a few circuits which can be constructed once as a pre-processing step. First, we prepare the input for the payoff functions in an ancillary register, where the input depends on the content of the register  $|\tilde{\tau}_t(x)\rangle$ . For this step, we prepare a conditional copy quantum circuit  $V_{\text{copy}}$  which operates as  $|x\rangle |\tilde{\tau}_t(x)\rangle |\mathbf{0}\rangle \rightarrow |x\rangle |\tilde{\tau}_t(x)\rangle |x_{\tilde{\tau}_t(x)}\rangle$ , where the register  $|x_{\tilde{\tau}_t(x)}\rangle$  stores the  $\tilde{\tau}_t(x)$ -th step of the path  $x$ . This circuit operates in time given by the size of the registers of at most  $O(T \log(T))$ . Second, from the access to the different payoff functions we construct access to the functions in superposition of the time parameter. By assumption, we are given quantum circuits  $V_{z_{t'}}$  for  $t' \in [T]$ . From these quantum circuits we construct the controlled circuit  $V_{\text{select}} := \sum_{t'=1}^T |t'\rangle \langle t'| \otimes V_{z_{t'}}$ , which consists of the controlled versions of the circuits  $V_{z_{t'}}$  and has a runtime of  $O(T \log(T) \mathcal{T}_z)$  [10]. Now, given  $|x\rangle |\tilde{\tau}_t(x)\rangle$ , with one application of  $V_{\text{copy}}$  and one application of  $V_{\text{select}}$ , we obtain  $z_{\tilde{\tau}_t(x)}$ , i.e, the payoff evaluated at  $x_{\tilde{\tau}_t(x)}$ , as

$$|x\rangle |\tilde{\tau}_t(x)\rangle |\mathbf{0}\rangle |\mathbf{0}\rangle \mapsto |x\rangle |\tilde{\tau}_t(x)\rangle |x_{\tilde{\tau}_t(x)}\rangle |\mathbf{0}\rangle \mapsto |x\rangle |\tilde{\tau}_t(x)\rangle |x_{\tilde{\tau}_t(x)}\rangle |z_{\tilde{\tau}_t(x)}\rangle.$$

Using  $V_{\text{copy}}$  again we uncompute the third register. One query to the function oracle  $e_{t-1,k}$  obtains  $e_{t-1,k}(x_{t-1})$  and by multiplication we obtain,

$$|x\rangle |\tilde{\tau}_t(x)\rangle |z_{\tilde{\tau}_t(x)}\rangle |e_{t-1,k}(x_{t-1})\rangle |\mathbf{0}\rangle \mapsto |x\rangle |\tilde{\tau}_t(x)\rangle |z_{\tilde{\tau}_t(x)}\rangle |e_{t-1,k}(x_{t-1})\rangle |z_{\tilde{\tau}_t(x)} e_{t-1,k}(x_{t-1})\rangle.$$

We uncompute the third and fourth registers using the given circuits to obtain the desired operation. The total runtime is  $O(T \log(T) \mathcal{T}_z + \mathcal{T}_e)$ .

Finally, for the third statement, it is not hard to see that (let  $\tilde{\tau}_T := T$ )

$$\begin{aligned} C_t^{(k)} |x\rangle |\mathbf{0}\rangle^{\otimes(3(T-t)+2)} &= W_T^\dagger \dots W_t^\dagger V_t^{(k)} |x\rangle |T\rangle |\mathbf{0}\rangle \bigotimes_{j=t}^{T-1} |z_j(x_j)\rangle |\tilde{\alpha}_j \cdot \tilde{e}_j(x_j)\rangle |\tilde{\tau}_j(x)\rangle \\ &= W_T^\dagger \dots W_t^\dagger |x\rangle |T\rangle |z_{\tilde{\tau}_t(x)} e_{t-1,k}(x_{t-1})\rangle \bigotimes_{j=t}^{T-1} |z_j(x_j)\rangle |\tilde{\alpha}_j \cdot \tilde{e}_j(x_j)\rangle |\tilde{\tau}_j(x)\rangle \\ &= |x\rangle |z_{\tilde{\tau}_t(x)} e_{t-1,k}(x_{t-1})\rangle |\mathbf{0}\rangle^{\otimes(3(T-t)+1)}. \end{aligned}$$

From the two previous statements, the runtime of  $C_t^{(k)}$  is  $2(T-t+1)O(\mathcal{T}_z + m\mathcal{T}_e) + O(T \log(T) \mathcal{T}_z + \mathcal{T}_e) = O(T(\log(T) \mathcal{T}_z + m\mathcal{T}_e))$ .  $\blacktriangleleft$

### 3.2 The algorithm

We present our quantum LSM algorithm in Algorithm 2. It computes the expectations  $\mathbb{E}[e_{t,k}(X_t) e_{t,l}(X_t)]$  (for the matrices  $\{A_t\}_{t=1}^{T-1}$ ),  $\mathbb{E}[Z_{\tilde{\tau}_{t+1}} \tilde{e}_t(X_t)]$  and  $\mathbb{E}[Z_{\tilde{\tau}_1}]$  using Theorem 13 instead of drawing random samples. Recall that by definition  $Z_{\tilde{\tau}_{t+1}} = z_{\tilde{\tau}_{t+1}(\mathbf{x}_{t+1})}(X_{\tilde{\tau}_{t+1}(\mathbf{x}_{t+1})})$ , i.e., both the optimal stopping time and the payoff depend on the path of the Markov chain.

As previously mentioned, it follows the classical version in Algorithm 1. However, the dynamic programming is not solved separately along different sampled paths, but in superposition along all possible stochastic processes. More specifically, at any given time  $t$ , the dynamic programming is solved in a backward fashion from time  $T$  to  $t+1$  by constructing a unitary that prepares the approximate stopping times  $\tilde{\tau}_{t+1}$  in superposition via the mapping  $|x\rangle |\mathbf{0}\rangle \mapsto |x\rangle |\tilde{\tau}_{t+1}(x)\rangle$  for all  $x \in E^T$ . Such unitary is constructed (Lemma 14) using the values of all stopping times  $\tilde{\tau}_{t+1}$  calculated so far in the dynamic program and allows access to the quantity  $Z_{\tilde{\tau}_{t+1}}$ , which in turn is used in the quantum subroutines for Monte Carlo to extract expectation values  $\mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]$  that make up the vector  $b_t$ . The matrices  $\{A_t\}_{t=1}^{T-1}$ , in turn, are computed in an entrywise fashion at the start of the algorithm by using quantum access to the functions  $e_{t,k}(x_t) e_{t,l}(x_t)$ . In hold of the approximations  $\tilde{A}_t$  and  $\tilde{b}_t$  to  $A_t$  and  $b_t$ , respectively, the vector  $\tilde{\alpha}_t = \tilde{A}_t^{-1} \tilde{b}_t$  is then computed classically and used to continue the dynamic programming at the next time step  $t$  when solving  $\tilde{\tau}_t(x) = t \mathbf{1}\{z_t(x_t) \geq \tilde{\alpha}_t \cdot \tilde{e}_t(x_t)\} + \tilde{\tau}_{t+1}(x) \mathbf{1}\{z_t(x_t) < \tilde{\alpha}_t \cdot \tilde{e}_t(x_t)\}$  from time  $T$  to time  $t$  in superposition. Such procedure is repeated until  $t=1$ , when the optimal stopping time  $\tilde{\tau}_1$  can be computed in superposition and thus the quantity  $\sup_\tau \mathbb{E}[Z_\tau]$  can finally be approximated by  $\max\{Z_0, \tilde{Z}_{\tilde{\tau}_1}\}$ . We note that the procedure of approximating a matrix  $A_t$  and a vector  $b_t$  entrywise via quantum algorithms for Monte Carlo followed by the classical computation of  $\tilde{\alpha}_t = \tilde{A}_t^{-1} \tilde{b}_t$  was already used in [50]. We also note that, unlike the classical LSM, our quantum algorithm requires redoing all previous dynamic programming steps before a given time  $t$  in order to progress into the next time step  $t-1$ . The final procedure involves  $O(T^2)$  time steps instead of  $O(T)$ .

■ **Algorithm 2** Quantum LSM algorithm for optimal stopping problem.

---

**Input:** Parameters  $\delta \in (0, 1)$ ,  $\epsilon > 0$ . Quantum sampling access to Markov chain  $(X_t)_{t=1}^T$  defined on a finite sample space  $\Omega$  and with finite state space  $E \subseteq \mathbb{R}^d$ . Quantum query access to  $\{z_t : E \rightarrow \mathbb{R}\}_{t=1}^T$  and linearly independent functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  for  $t \in [T-1]$ . Let  $L := \max_{t \in [T-1], k \in [m]} \|e_{t,k}\|_{L^2(\rho_t)}$  and  $R := \max_{t \in [T]} \|z_t\|_u$ .

- 1:  $\delta_A := \delta/(4Tm^2)$ ,  $\delta_b := \delta/(4Tm)$ ,  $\epsilon_A := \epsilon/m$  and  $\epsilon_b := \epsilon/\sqrt{m}$ .
- 2: Construct quantum access and controlled rotation access to  $e_{t,k}e_{t,l}$ ,  $\forall k, l \in [m]$ ,  $t \in [T-1]$ , with quantum query access to  $e_{t,j}$ , quantum circuits for multiplication and Fact 12.
- 3: Compute  $\{\tilde{A}_t\}_{t=1}^{T-1}$  by calling **QMonteCarlo** $(e_{t,k}(X_t)e_{t,l}(X_t), \epsilon_A, \delta_A, L^2)$  for  $k, l \in [m]$ .
- 4: Compute the inverses  $\{\tilde{A}_t^{-1}\}_{t=1}^{T-1}$ .
- 5: Prepare unitary  $W_T$  s.t.  $W_T|x\rangle|\mathbf{0}\rangle = |x\rangle|\tilde{\tau}_T(x)\rangle$ , where  $\tilde{\tau}_T(x) = T$  for all  $x \in E^T$ .
- 6: **for**  $t = T$  to 2 **do**
- 7:     **if**  $t \neq T$  **then**
- 8:         Prepare unitary  $W_t$  s.t.  $W_t|x\rangle|\tilde{\tau}_{t+1}(x)\rangle|\mathbf{0}\rangle^{\otimes 3} = |x\rangle|\tilde{\tau}_{t+1}(x)\rangle|z_t(x_t)\rangle|\tilde{\alpha}_t \cdot \vec{e}_t(x_t)\rangle|\tilde{\tau}_t(x)\rangle$  for any  $\tilde{\tau}_{t+1}(x) \in [T]$  (Lemma 14).
- 9:     **end if**
- 10:     Prepare unitaries  $\{V_t^{(k)}\}_{k=1}^m$  s.t.  $V_t^{(k)}|x\rangle|\tilde{\tau}_t(x)\rangle|\mathbf{0}\rangle = |x\rangle|\tilde{\tau}_t(x)\rangle|z_{\tilde{\tau}_t(x)}e_{t-1,k}(x_{t-1})\rangle$  (Lemma 14).
- 11:     Prepare unitary  $W_T^\dagger \dots W_{t+1}^\dagger W_t^\dagger V_t^{(k)} W_t W_{t+1} \dots W_T$  for  $k \in [m]$  (Lemma 14).
- 12:     Construct quantum access to the controlled rotations of the functions  $z_{\tilde{\tau}_t(x)}e_{t-1,k}(x_{t-1})$  (Fact 12).
- 13:     Execute **QMonteCarlo** $(Z_{\tilde{\tau}_t} e_{t-1,k}(X_{t-1}), \epsilon_b, \delta_b, RL)$ , for all  $k \in [m]$ , to compute  $\tilde{b}_{t-1}$ .
- 14:     Compute the vector  $\tilde{\alpha}_{t-1} = \tilde{A}_{t-1}^{-1} \tilde{b}_{t-1}$  classically.
- 15:     **end for**
- 16:     Prepare unitary  $W_1$  s.t.  $W_1|x\rangle|\tilde{\tau}_2(x)\rangle|\mathbf{0}\rangle^{\otimes 3} = |x\rangle|\tilde{\tau}_2(x)\rangle|z_1(x_1)\rangle|\tilde{\alpha}_1 \cdot \vec{e}_1(x_1)\rangle|\tilde{\tau}_1(x)\rangle$  for any  $\tilde{\tau}_2(x) \in [T]$  (Lemma 14).
- 17:     Prepare unitary  $V_1$  s.t.  $V_1|x\rangle|\tilde{\tau}_1(x)\rangle|\mathbf{0}\rangle = |x\rangle|\tilde{\tau}_1(x)\rangle|z_{\tilde{\tau}_1(x)}\rangle$  (Lemma 14).
- 18:     Prepare unitary  $W_T^\dagger \dots W_2^\dagger W_1^\dagger V_1 W_1 W_2 \dots W_T$  (Lemma 14).
- 19:     Construct quantum access to the controlled rotations of the function  $z_{\tilde{\tau}_1}$  (Fact 12).
- 20:     Execute **QMonteCarlo** $(Z_{\tilde{\tau}_1}, \epsilon, \frac{\delta}{2}, R)$  to compute  $\tilde{Z}_{\tilde{\tau}_1}$ .
- 21:     Output  $\tilde{\mathcal{U}}_0 := \max\{Z_0, \tilde{Z}_{\tilde{\tau}_1}\}$ .

---

### 3.3 Error analysis and complexity

In Appendix A we prove that the classical LSM algorithm and our proposed quantum LSM algorithm approximate the sought-after quantity  $\mathcal{U}_0$  up to additive accuracy with high probability. Among several results, the following encapsulates the overall complexity of the classical and quantum LSM algorithms. For simplicity we assume that  $\mathcal{T}_{\text{samp}}, \mathcal{T}_z, \mathcal{T}_e = O(1)$ .

► **Theorem 15** (Informal version of Corollary 22). *Consider a set of linearly independent functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  for each  $t \in [T-1]$  and payoff functions  $\{z_t : E \rightarrow \mathbb{R}\}_{t=0}^T$ . Then, for  $\delta \in (0, 1)$  and  $\epsilon > 0$ , the classical and quantum LSM algorithms output  $\tilde{\mathcal{U}}_0$  such that*

$$\Pr \left[ |\tilde{\mathcal{U}}_0 - \mathcal{U}_0| \geq 5^T \left( \epsilon + \max_{0 < t < T} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} \right) \right] \leq \delta$$

using time, respectively,  $\tilde{O}\left(\frac{Tm^6}{\epsilon^2}\right)$  and  $\tilde{O}\left(\frac{T^2m^4}{\epsilon}\right)$ , up to polylog terms.

The error  $\epsilon$  arises from the Monte Carlo subroutines and can be made smaller by increasing the calls to the quantum inputs (or to the number of sampled paths in the classical counterpart). Compared to the classical algorithm, the number of oracle calls is quadratically less in the quantum algorithm. The quantity  $\min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)}$  appearing in the theorem above is known as *approximation error*. This term arises from approximating the continuation values by the  $m$  expansion functions and is a deterministic quantity implicitly dependent on  $m$  and on smoothness properties of the continuation values.

In order to obtain a final additive accuracy  $\epsilon_{\text{final}}$  for  $\tilde{\mathcal{U}}_0$ , we must resolve the implicit dependence of the approximation error on  $m$ . This is done by considering specific sets of expansion functions and assuming sufficiently good smoothness properties for the continuation values. More specifically, for each  $t \in [T - 1]$  we consider functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  that generate the space  $\mathcal{R}_q$  of all polynomials of degree at most  $q$ , so that  $m = \binom{q+d}{d}$ . We also assume that  $\mathbb{E}[Z_{\tau_{t+1}} | X_t] \in C^n$ , i.e., the continuation values are  $n$ -differentiable functions. Then it is possible to bound the approximation error  $\min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)}$  by using a Jackson-like inequality [47] and obtain the following result (see the arXiv version [28] for the full statement and proof).

► **Theorem 16.** *For each  $t \in [T - 1]$  consider a set of linearly independent functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  that spans the space  $\mathcal{R}_q$  with  $m = \binom{q+d}{d}$  and consider payoff functions  $\{z_t : E \rightarrow \mathbb{R}\}_{t=0}^T$ . Assume that  $\mathbb{E}[Z_{\tau_{t+1}} | X_t] \in C^n$  for all  $t \in \{0, \dots, T - 1\}$ , where  $n \leq q$ . Then, for  $\delta \in (0, 1)$  and  $\epsilon \geq 0$ , if  $q = \lceil (5^T/\epsilon)^{1/n} \rceil$ , the classical and quantum LSM algorithms output  $\tilde{\mathcal{U}}_0$  such that  $\Pr [|\tilde{\mathcal{U}}_0 - \mathcal{U}_0| \geq \epsilon] \leq \delta$  using time, respectively,  $\tilde{O}((5^T/\epsilon)^{2+6d/n})$  and  $\tilde{O}((5^T/\epsilon)^{1+4d/n})$ , up to polylog terms.*

If the continuation values are  $n$ -times differentiable, for  $n = \Theta(\log(5^T/\epsilon)/\log \log(5^T/\epsilon))$ , then we get the sought-after quadratic improvement from  $\tilde{O}((5^T/\epsilon)^2)$  classical runtime to  $\tilde{O}(5^T/\epsilon)$  quantum runtime, up to polylog terms. We briefly note that such smoothness conditions on the continuation values are not unusual in areas like finance. Indeed, the continuation values can even be in  $C^\infty$  in some models, e.g. Black-Scholes [34, 85].

Very recently, Miyamoto [62] proposed a quantum LSM algorithm based on Chebyshev interpolation through Chebyshev nodes and obtained  $O(\epsilon^{-1} \log^d(1/\epsilon) \text{poly log log}(1/\epsilon))$  as a final complexity. Our approach, in contrast, is to project  $\mathbb{E}[Z_{\tau_{t+1}} | X_t]$  onto a set of polynomials and is, for this reason, much more general. Moreover, our final result is a *time* complexity, while the result from [62] is a *query* complexity on the number of unitaries called by all quantum routines for Monte Carlo. Finally, Miyamoto [62] assumes that the continuation values are analytical functions, i.e., are in  $C^\infty$ , while we only need to assume  $\mathbb{E}[Z_{\tau_{t+1}} | X_t] \in C^n$  for  $n = \Theta(\log(5^T/\epsilon)/\log \log(5^T/\epsilon))$  in order to recover  $\tilde{O}(\epsilon^{-1})$  up to polylog factors. One downside of our approach, though, is the presence of quantities that implicitly depend on the underlying Markov chain.

As just mentioned, the full results behind the informal theorems above involve parameters that depend on the underlying Markov chain such as the minimum singular value of the matrices  $A_t$ . In order to explicitly work these parameters out, we also study the case when the underlying Markov process follows Brownian motion or geometric Brownian motion and obtain a simplified version of our algorithm (see arXiv version [28]). In the case of Brownian motion, we choose Hermite polynomials as the functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  for each  $t \in [T - 1]$ , since they are orthogonal under the probability measure underlying a Brownian motion. This means that the matrices  $A_t$  are just the identity. The final result is a mild reduction on the classical and quantum time complexities to  $\tilde{O}((5^T/\epsilon)^{2+4d/n})$  and  $\tilde{O}((5^T/\epsilon)^{1+7d/2n})$ , respectively. For the geometric Brownian motion,



we pick suitable monomials that reduce the matrices  $A_t$  to Vandermonde matrices, whose minimum singular value can be bounded. We obtain the final classical and quantum complexities  $e^{O((5^T/\epsilon)^{2/n})}(5^T/\epsilon)^{2+12d/n}$  and  $e^{O((5^T/\epsilon)^{2/n})}(5^T/\epsilon)^{1+15d/2n}$ , respectively. If the continuation values are again  $n$ -times differentiable for  $n = \Theta(\log(5^T/\epsilon)/\log\log(5^T/\epsilon))$ , then the classical and quantum complexities for the Brownian motion setting reduce to the usual  $\tilde{O}(\epsilon^{-2})$  and  $\tilde{O}(\epsilon^{-1})$ , respectively, while, for the geometric Brownian motion, they reduce to  $e^{O(\log^c(5^T/\epsilon))}(5^T/\epsilon)^2$  and  $e^{O(\log^c(5^T/\epsilon))}(5^T/\epsilon)$  for any constant  $0 < c < 1$ . These results for the geometric Brownian motion are slightly weaker than the usual  $\tilde{O}((5^T/\epsilon)^2)$  and  $\tilde{O}(5^T/\epsilon)$ , since the bound on the minimum singular value of the matrix  $A_t$  is very sensitive to the degree  $q$  of the chosen monomials.

## 4 Conclusions

In this work, we developed a new quantum algorithm for a stochastic optimal stopping problem (as in Theorem 4) with a quantum advantage in the runtime. This problem cannot be solved accurately by a single application of quantum algorithms for Monte Carlo [63, 41, 22, 21, 3]. Instead one must compute in superposition (and recursively) the stopping times as in Lemma 14, which is key to obtaining a quantum speedup. As the classical LSM algorithm can be used to solve a large variety of problems, our quantum LSM can also be used for problems in finance including insurance [53] and risk management [38], and for many optimization problems outside finance, such as quickest detection [82] and sequential Bayesian hypothesis testing [26]. Additionally, we believe that there are many other problems in, for example, dynamic programming, stochastic optimal stopping and optimal control where the interplay of function approximation and quantum subroutines for Monte Carlo could be used to design new quantum algorithms.

A few design choices of the quantum algorithm were guided by real problems where the classical algorithm is already used. Even though we took number of expansion functions  $m = \text{poly}(5^T/\epsilon)$  in order to bound the approximation error, in practice one typically assumes  $m$  to be constant [57]. For big values of  $m$ , our algorithm could be modified in order to use quantum subroutines for inner product estimation, and reduce the complexity polynomially in  $m$ , but introducing a further  $\epsilon$  dependence. Thus, further analysis is needed to understand the impact of the precision parameters on the runtime of these subroutines. Along these lines, we have chosen to invert the linear systems for finding  $\alpha_t$  on a classical computer. A possible modification of our algorithm could output quantum states  $|\alpha_t\rangle$  via HHL-like algorithms [43]. We also discussed how, under the hypothesis that the Markov chain is a Brownian motion or a Geometric Brownian motion, the matrices  $A_t$  can be expressed with a closed formula and their minimum singular value be bounded. This idea exhibits some similarity with the idea proposed in [56]. There, they leveraged quasi-regression algorithms and a particular choice of expansion functions [68], so to pre-compute the matrices  $A_t$ , and thus skip costly Monte Carlo estimators. Moreover, when considering a Brownian or a Geometric Brownian motion, the chosen functions  $\{e_{t,k}\}_{k=1}^m$  had an explicit time dependence on  $t$ , but it is possible to transform the optimal stopping time problem behind American option pricing in a way that such dependence is suppressed and a single set of approximating functions is employed [15, 16]. We believe that such approach could improve our complexities. Finally, in our algorithm, we use quantum subroutines from [63], but could equivalently use the subroutines from [41, 22, 21]. Our template could be extended to quantum algorithms that are similar in spirit but are solving different problems [87].



Our final complexities have an exponential dependence on  $T$ , the number of time steps. We believe that such dependence, present in several past works [30, 93, 94, 95, 96, 62], is only a consequence of a loose error bound and could possibly be improved. Such hope is backed up by the ubiquitous employment of LSM algorithms for pricing American options in every day financial markets. We also note that a more careful error analysis would improve classical results as well, but, regardless, the quantum quadratic improvement would still be present. Finally, notice that it is always possible to compensate a reduction on the number of time steps with more accurate approximations for continuation values and similar quantities.

We stress the importance of fast quantum algorithms for optimal stopping problems. For American option pricing, the value of the payoff function could easily reach a few million dollars, and the additive precision  $\epsilon$  could be of the order of  $10^{-11}$  [4]. Given the level of specialization in classical algorithms and architectures for this specific problem, how and when our algorithm can find application in practice is left for future work.

---

## References

- 1 Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvency II), 2009.
- 2 Javier Alcazar, Vicente Leyton-Ortega, and Alejandro Perdomo-Ortiz. Classical versus quantum models in machine learning: insights from a finance application. *Machine Learning: Science and Technology*, 1(3):035003, 2020.
- 3 Dong An, Noah Linden, Jin-Peng Liu, Ashley Montanaro, Changpeng Shao, and Jiasu Wang. Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance. *Quantum*, 5:481, 2021.
- 4 Leif B. G. Andersen, Mark Lake, and Dimitri Offengenden. High-performance American option pricing. *Journal of Computational Finance*, 20(1):39–87, 2016.
- 5 Anna Rita Bacinello, Enrico Biffis, and Pietro Millosovich. Pricing life insurance contracts with early exercise features. *Journal of computational and applied mathematics*, 233(1):27–35, 2009.
- 6 Panagiotis Kl. Barkoutsos, Giacomo Nannicini, Anton Robert, Ivano Tavernelli, and Stefan Woerner. Improving variational quantum optimization using CVaR. *Quantum*, 4:256, 2020.
- 7 Giovanni Barone-Adesi and Robert E. Whaley. Efficient analytic approximation of American option values. *the Journal of Finance*, 42(2):301–320, 1987.
- 8 Christian Bender and Jessica Steiner. Least-squares Monte Carlo for backward SDEs. In *Numerical methods in finance*, pages 257–289. Springer, 2012.
- 9 Alain Bensoussan and J.-L. Lions. *Applications of variational inequalities in stochastic control*. Elsevier, 2011.
- 10 Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys. Rev. Lett.*, 114:090502, 2015. doi:10.1103/PhysRevLett.114.090502.
- 11 Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973.
- 12 Adam Bouland, Wim van Dam, Hamed Joorati, Iordanis Kerenidis, and Anupam Prakash. Prospects and challenges of quantum finance. *arXiv preprint*, 2020. arXiv:2011.06492.
- 13 Phelim P. Boyle. Options: a Monte Carlo approach. *Journal of Financial Economics*, 4(3):323–338, 1977.
- 14 Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.
- 15 Mark Broadie, Jérôme Detemple, Eric Ghysels, and Olivier Torrès. American options with stochastic dividends and volatility: A nonparametric investigation. *Journal of Econometrics*, 94(1-2):53–92, 2000.

- 16 Mark Broadie, Jérôme Detemple, Eric Ghysels, and Olivier Torrès. Nonparametric estimation of American options' exercise boundaries and call prices. *Journal of Economic Dynamics and Control*, 24(11-12):1829–1857, 2000.
- 17 Mark Broadie and Paul Glasserman. Monte Carlo methods for pricing high-dimensional American options: an overview. *Net Exposure*, 3:15–37, 1997.
- 18 Shouvanik Chakrabarti, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J. Zeng. A threshold for quantum advantage in derivative pricing. *Quantum*, 5:463, 2021.
- 19 Don M. Chance. A synthesis of binomial option pricing models for lognormally distributed assets. *Journal of Applied Finance (Formerly Financial Practice and Education)*, 18(1), 2008.
- 20 Emmanuelle Clément, Damien Lamberton, and Philip Protter. An analysis of a least squares regression method for American option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.
- 21 Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi. Near-optimal quantum algorithms for multivariate mean estimation. *arXiv preprint*, 2021. [arXiv:2111.09787](https://arxiv.org/abs/2111.09787).
- 22 Arjan Cornelissen and Sofiene Jerbi. Quantum algorithms for multivariate Monte Carlo estimation. *arXiv preprint*, 2021. [arXiv:2107.03410](https://arxiv.org/abs/2107.03410).
- 23 John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option pricing: a simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979.
- 24 Gary Wayne Crosby. *Optimal multiple stopping: theory and applications*. PhD thesis, The University of North Carolina at Charlotte, 2017.
- 25 Samudra Dasgupta and Arnab Banerjee. Quantum annealing algorithm for expected shortfall based dynamic asset allocation. *arXiv preprint*, 2019. [arXiv:1909.12904](https://arxiv.org/abs/1909.12904).
- 26 Constantinos Daskalakis and Yasushi Kawase. Optimal stopping rules for sequential hypothesis testing. In *25th Annual European Symposium on Algorithms (ESA)*, 2017.
- 27 Georgios Dimitrakopoulos. Least-squares Monte Carlo simulation and high performance computing for Solvency II regulatory capital estimation. Master's thesis, The University of Manchester (United Kingdom), 2013.
- 28 João F. Doriguello, Alessandro Luongo, Jinge Bao, Patrick Rebentrost, and Miklos Santha. Quantum algorithm for stochastic optimal stopping problems with applications in finance. *arXiv preprint*, 2021. [arXiv:2111.15332](https://arxiv.org/abs/2111.15332).
- 29 Daniel J. Egger, Claudio Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain. Quantum computing for finance: state of the art and future prospects. *IEEE Transactions on Quantum Engineering*, 2020.
- 30 Daniel Egloff. Monte Carlo algorithms for optimal stopping and statistical learning. *The Annals of Applied Probability*, 15(2):1396–1432, 2005.
- 31 Hans Föllmer and Alexander Schied. *Stochastic finance*. de Gruyter, 2016.
- 32 Filipe Fontanela, Antoine Jacquier, and Mugad Oumgari. A quantum algorithm for linear PDEs arising in finance. *SIAM Journal on Financial Mathematics*, 12(4):SC98–SC114, 2021.
- 33 Michael C. Fu, Scott B. Laprise, Dilip B. Madan, Yi Su, and Rongwen Wu. Pricing American options: a comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3):39–88, 2001.
- 34 Stefan Gerhold. The Longstaff-Schwartz algorithm for Lévy models: results on fast and slow convergence. *The Annals of Applied Probability*, 21(2):589–608, 2011.
- 35 Emmanuel Gobet, Jean-Philippe Lemor, and Xavier Warin. A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3):2172–2202, 2005.
- 36 Emmanuel Gobet and Plamen Turkedjiev. Approximation of discrete BSDE using least-squares regression, November 2011. Technical report. URL: <https://hal.archives-ouvertes.fr/hal-00642685>.
- 37 Gene H. Golub and Charles F. Van Loan. *Matrix computations*, 2013.

- 38 Andrew Green. *XVA: Credit, Funding and Capital Valuation Adjustments*. John Wiley & Sons, 2015.
- 39 Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint*, 2002. [arXiv:quant-ph/0208112](https://arxiv.org/abs/quant-ph/0208112).
- 40 Eli Gutin. *Practical applications of large-scale stochastic control for learning and optimization*. PhD thesis, Massachusetts Institute of Technology, 2018.
- 41 Yassine Hamoudi. Quantum sub-Gaussian mean estimator. In *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 42 Jeong Yu Han and Patrick Rebentrost. Quantum advantage for multi-option portfolio pricing and valuation adjustments. *arXiv preprint*, 2022. [arXiv:2203.04924](https://arxiv.org/abs/2203.04924).
- 43 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.
- 44 Steven Herbert. No quantum speedup with Grover-Rudolph state preparation for quantum Monte Carlo integration. *Physical Review E*, 103(6):063302, 2021.
- 45 Mark Hodson, Brendan Ruck, Hugh Ong, David Garvin, and Stefan Dulman. Portfolio rebalancing experiments using the quantum alternating operator ansatz. *arXiv preprint*, 2019. [arXiv:1911.05296](https://arxiv.org/abs/1911.05296).
- 46 Jacqueline Huang and Jong-Shi Pang. Option pricing and linear complementarity. Technical report, Cornell University, 2003.
- 47 Dunham Jackson. A general class of problems in approximation. *American Journal of Mathematics*, 46(4):215–234, 1924.
- 48 Patrick Jaillet, Damien Lambertson, and Bernard Lapeyre. Variational inequalities and the pricing of American options. *Acta Applicandae Mathematica*, 21(3):263–289, 1990.
- 49 Kazuya Kaneko, Koichi Miyamoto, Naoyuki Takeda, and Kazuyoshi Yoshino. Quantum pricing with a smile: implementation of local volatility model on quantum computer. *arXiv preprint*, 2020. [arXiv:2007.01467](https://arxiv.org/abs/2007.01467).
- 50 Kazuya Kaneko, Koichi Miyamoto, Naoyuki Takeda, and Kazuyoshi Yoshino. Linear regression by quantum amplitude estimation and its extension to convex optimization. *Phys. Rev. A*, 104:022430, 2021. [doi:10.1103/PhysRevA.104.022430](https://doi.org/10.1103/PhysRevA.104.022430).
- 51 In Joon Kim. The analytic valuation of American options. *The Review of Financial Studies*, 3(4):547–572, 1990.
- 52 Michael Kohler. A review on regression-based Monte Carlo methods for pricing American options. In *Recent developments in applied probability and statistics*, pages 37–58. Springer, 2010.
- 53 Anne-Sophie Krah, Zoran Nikolić, and Ralf Korn. A least-squares Monte Carlo framework in proxy modeling of life insurance companies. *Risks*, 6(2):62, 2018.
- 54 Jean-Philippe Lemor, Emmanuel Gobet, and Xavier Warin. Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations. *Bernoulli*, 12(5):889–916, 2006.
- 55 Xiaofei Li, Yi Wu, Quanxin Zhu, Songbo Hu, and Chuan Qin. A regression-based Monte Carlo method to solve two-dimensional forward backward stochastic differential equations. *Advances in Difference Equations*, 2021(1):1–13, 2021.
- 56 Chen Liu, Henry Schellhorn, and Qidi Peng. American option pricing with regression: convergence analysis. *International Journal of Theoretical and Applied Finance*, 22(08):1950044, 2019.
- 57 Francis A. Longstaff and Eduardo S. Schwartz. Valuing American options by simulation: a simple least-squares approach. *The review of financial studies*, 14(1):113–147, 2001.
- 58 Gabriel Marin-Sancheza, Javier Gonzalez-Conde, and Mikel Sanz. Quantum algorithms for approximate function loading. *arXiv preprint*, 2021. [arXiv:2111.07933](https://arxiv.org/abs/2111.07933).
- 59 Ana Martin, Bruno Candelas, Ángel Rodríguez-Rozas, José D. Martín-Guerrero, Xi Chen, Lucas Lamata, Román Orús, Enrique Solano, and Mikel Sanz. Toward pricing financial derivatives with an IBM quantum computer. *Physical Review Research*, 3(1):013167, 2021.

- 60 Henry P. McKean Jr. A free boundary problem for the heat equation arising from a problem of mathematical economics. *Industrial Management Review*, 6:32–39, 1965.
- 61 Robert C. Merton. Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.
- 62 Koichi Miyamoto. Bermudan option pricing by quantum amplitude estimation and Chebyshev interpolation. *arXiv preprint*, 2021. [arXiv:2108.09014](https://arxiv.org/abs/2108.09014).
- 63 Ashley Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015.
- 64 Alexey Muravlev and Mikhail Zhitlukhin. A Bayesian sequential test for the drift of a fractional Brownian motion. *Advances in Applied Probability*, 52(4):1308–1324, 2020.
- 65 Jacques Neveu. *Discrete-parameter martingales*. Elsevier, 1975.
- 66 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. doi:10.1017/CB09780511976667.
- 67 Roman Orus, Samuel Mugel, and Enrique Lizaso. Quantum computing for finance: overview and prospects. *Reviews in Physics*, 4:100028, 2019.
- 68 Art B. Owen. Assessing linearity in high dimensions. *The Annals of Statistics*, 28(1):1–19, 2000.
- 69 Antoon Pelsser and Janina Schweizer. The difference between lsmc and replicating portfolio in insurance liability modeling. *European actuarial journal*, 6(2):441–494, 2016.
- 70 Goran Peskir and Albert Shiryaev. *Optimal stopping and free-boundary problems*. Springer, 2006.
- 71 Warren B. Powell. A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3):795–821, 2019.
- 72 Santosh Kumar Radha. Quantum option pricing using wick rotated imaginary time evolution. *arXiv preprint*, 2021. [arXiv:2101.04280](https://arxiv.org/abs/2101.04280).
- 73 Sergi Ramos-Calderer, Adrián Pérez-Salinas, Diego García-Martín, Carlos Bravo-Prieto, Jorge Cortada, Jordi Planaguma, and José I. Latorre. Quantum unary approach to option pricing. *Physical Review A*, 103(3):032414, 2021.
- 74 Patrick Rebentrost, Brajesh Gupta, and Thomas R. Bromley. Quantum computational finance: Monte Carlo pricing of financial derivatives. *Phys. Rev. A*, 98:022321, 2018.
- 75 Patrick Rebentrost and Seth Lloyd. Quantum computational finance: quantum algorithm for portfolio optimization. *arXiv preprint*, 2018. [arXiv:1811.03975](https://arxiv.org/abs/1811.03975).
- 76 Patrick Rebentrost, Miklos Santha, and Siyi Yang. Quantum alphasatron. *arXiv preprint*, 2021. [arXiv:2108.11670](https://arxiv.org/abs/2108.11670).
- 77 Richard J. Rendleman. Two-state option pricing. *The Journal of Finance*, 34(5):1093–1110, 1979.
- 78 Leonard C. G. Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286, 2002.
- 79 Joose Mikko Juhani Sauli. On the suitability of the Longstaff-Schwartz term structure model for modelling the cost of government debt. Master’s thesis, Helsingfors Universitet, 2013.
- 80 William F. Sharpe, Gordon J. Alexander, and Jeffrey W. Bailey. *Investments*. Prentice-Hall, 1999.
- 81 Albert N. Shiryaev. *Optimal stopping rules*, volume 8. Springer Science & Business Media, 2007.
- 82 Albert N. Shiryaev. Quickest detection problems: fifty years later. *Sequential Analysis*, 29(4):345–385, 2010.
- 83 Nikitas Stamatopoulos, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner. Option pricing using quantum computers. *arXiv preprint*, 2019. [arXiv:1905.02666](https://arxiv.org/abs/1905.02666).
- 84 Nikitas Stamatopoulos, Guglielmo Mazzola, Stefan Woerner, and William J. Zeng. Towards quantum advantage in financial market risk using quantum gradient algorithms. *arXiv preprint*, 2021. [arXiv:2111.12509](https://arxiv.org/abs/2111.12509).

- 85 Peter Tankov, Ekaterina Voltchkova, and Rama Cont. Option pricing models with jumps: integro-differential equations and inverse problems. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, 2004.
- 86 Elliot Tonkes and Dharma Lesmono. A Longstaff and Schwartz approach to the early election problem. *Advances in Decision Sciences*, 2012, 2012.
- 87 John N. Tsitsiklis and Benjamin Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.
- 88 Pierre Van Moerbeke. On optimal stopping and free boundary problems. *Archive for Rational Mechanics and Analysis*, 60(2):101–148, 1976.
- 89 Almudena Carrera Vazquez and Stefan Woerner. Efficient state preparation for quantum amplitude estimation. *Physical Review Applied*, 15(3):034027, 2021.
- 90 Abraham Wald. *Sequential Analysis*. John Wiley and Sons, 1st edition edition, 1947.
- 91 Pawel Wocjan, Chen-Fu Chiang, Daniel Nagaj, and Anura Abeyesinghe. Quantum algorithm for approximating partition functions. *Phys. Rev. A*, 80:022340, 2009.
- 92 Stefan Woerner and Daniel J. Egger. Quantum risk analysis. *arXiv preprint*, 2018. arXiv:1806.06893.
- 93 Daniel Z. Zanger. Convergence of a least-squares Monte Carlo algorithm for bounded approximating sets. *Applied Mathematical Finance*, 16(2):123–150, 2009.
- 94 Daniel Z. Zanger. Quantitative error estimates for a least-squares Monte Carlo algorithm for American option pricing. *Finance and Stochastics*, 17(3):503–534, 2013.
- 95 Daniel Z. Zanger. Convergence of a least-squares Monte Carlo algorithm for American option pricing with dependent sample data. *Mathematical Finance*, 28(1):447–479, 2018.
- 96 Daniel Z. Zanger. General error estimates for the Longstaff-Schwartz least-squares Monte Carlo algorithm. *Mathematics of Operations Research*, 45(3):923–946, 2020.

## A Error Analysis and Complexity

In what follows, given  $b \in \mathbb{R}^m$  and  $A \in \mathbb{R}^{m \times m}$  for some  $m \in \mathbb{N}$ , let  $\|b\|_2 := \sqrt{\sum_{i=1}^m b_i^2}$  and  $\|A\|_2 := \sigma_{\max}(A)$  be the vector and matrix norms, respectively, where  $\sigma_{\max}(A)$  is the maximum singular value of  $A$ . We shall denote by  $\sigma_{\min}(A)$  the minimum singular value of  $A$ . Let  $\omega_*$  denote the matrix multiplication exponent. Moreover, recall the uniform norm  $\|f\|_u = \sup\{|f(s)| : s \in E\}$  for  $f : E \rightarrow \mathbb{R}$ .

We shall analyze the approximation error and complexity from Algorithm 2. In order to do so, we will need the following result from [93, 96] (already modified to our notation) that bounds the error between the exact continuation values  $\mathbb{E}[Z_{\tau_{t+1}} | X_t]$  and their approximation  $\tilde{\alpha}_t \cdot \tilde{e}_t(X_t)$  in terms of the error between the continuation values  $\mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]$  evaluated on the approximated stopping times  $\tilde{\tau}_{k+1}$  and  $\tilde{\alpha}_k \cdot \tilde{e}_k(X_k)$  for  $k \in \{t, \dots, T-1\}$ . Recall the image probability measures  $\rho_t$  in  $E \subseteq \mathbb{R}^d$  induced by each element  $X_t$ ,  $t \in \{0, \dots, T\}$ .

► **Lemma 17** ([93, Lemma 2.2]). *For each  $t \in \{0, \dots, T-1\}$ , we have*

$$\begin{aligned} \|\tilde{\alpha}_t \cdot \tilde{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} &\leq 2 \sum_{k=t}^{T-1} \|\tilde{\alpha}_k \cdot \tilde{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)}, \\ \|\mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t] - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} &\leq 2 \sum_{k=t+1}^{T-1} \|\tilde{\alpha}_k \cdot \tilde{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)}, \end{aligned}$$

where  $\tilde{\alpha}_0 \cdot \tilde{e}_0(X_0) := \tilde{Z}_{\tilde{\tau}_1}$  approximates  $\mathbb{E}[Z_{\tilde{\tau}_1}]$ .

We will also need the following technical result on the sensitivity of square systems.

► **Theorem 18** ([37, Theorem 2.6.2]). *Let  $Ax = b$  and  $\tilde{A}\tilde{x} = \tilde{b}$ , where  $A, \tilde{A} \in \mathbb{R}^{d \times d}$  and  $b, \tilde{b} \in \mathbb{R}^d$ , with  $b \neq 0$ . Suppose that  $\|A - \tilde{A}\|_2 \leq \epsilon_A$  and  $\|b - \tilde{b}\|_2 \leq \epsilon_b$ . If  $\epsilon_A \leq \sigma_{\min}(A)/2$ , where  $\sigma_{\min}(A)$  is the minimum singular value of  $A$ , then*

$$\|x - \tilde{x}\|_2 \leq \frac{2}{\sigma_{\min}(A)} \left( \frac{\epsilon_A \|b\|_2}{\sigma_{\min}(A)} + \epsilon_b \right).$$

We are now able to state a central theorem for our quantum LSM algorithm.

► **Theorem 19.** *Within the setting of Algorithm 2 with input parameters  $\delta$  and  $\epsilon$ , let  $T\mathcal{T}_{\text{samp}}$  be the sampling cost of the Markov chain and consider a set of linearly independent functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  for each  $t \in [T-1]$  with  $L := \max_{t \in [T-1], k \in [m]} \|e_{t,k}\|_{L^2(\rho_t)}$  and query cost  $\mathcal{T}_e$ . Also consider  $\{z_t : E \rightarrow \mathbb{R}\}_{t=0}^T$  with  $R := \max_{t \in [T]} \|z_t\|_u < \infty$  and query cost  $\mathcal{T}_z$ . Moreover, let  $\sigma_{\min} := \min_{t \in [T-1]} \sigma_{\min}(A_t) > 0$ . Assume that  $\sqrt{m}RL/\sigma_{\min} \geq 1$  and define  $\mathcal{T}_{\text{total}} := \mathcal{T}_{\text{samp}} + \mathcal{T}_z + \mathcal{T}_e$ . Then, for any  $\delta \in (0, 1)$  and  $\epsilon \in (0, \sigma_{\min}/2]$ , Algorithm 2 outputs  $\tilde{\mathcal{U}}_0$  such that*

$$\Pr \left[ |\tilde{\mathcal{U}}_0 - \mathbb{E}[Z_{\tau_0}]| \geq \frac{8T\epsilon mRL^2}{\sigma_{\min}^2} + 2 \sum_{t=1}^{T-1} \min_{a \in \mathbb{R}^m} \|a \cdot \tilde{e}_t(X_t) - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)} \right] \leq \delta \quad (11)$$

in time

$$O \left( \frac{T^2 m^3}{\epsilon} \mathcal{T}_{\text{total}} L(L+R) \log(T) \log(Tm^2/\delta) \log^{3/2}(mL(L+R)/\epsilon) \log \log(mL(L+R)/\epsilon) \right).$$

**Proof.** Set  $b_t := \mathbb{E}[Z_{\tilde{\tau}_{t+1}} \tilde{e}_t(X_t)]$ . Recall that  $\epsilon_A := \epsilon/m$ ,  $\epsilon_b := \epsilon/\sqrt{m}$ ,  $\delta_A := \delta/(4Tm^2)$  and  $\delta_b := \delta/(4Tm)$ . We start by computing the complexity of the algorithm. We first note the bounds  $\|b_t\|_2 \leq R \|\mathbb{E}[\tilde{e}_t(X_t)]\|_2 \leq \sqrt{m}RL$  and  $\|A_t\|_2 \leq m \max_{k,l} \|\mathbb{E}[e_{t,k}(X_t)e_{t,l}(X_t)]\| \leq mL^2$  for all  $t \in [T-1]$ . The computation of all the entries of the matrices  $\{A_t\}_{t=1}^{T-1}$  requires time

$$O \left( \frac{Tm^2}{\epsilon_A} L^2 (T\mathcal{T}_{\text{samp}} + \mathcal{T}_e) \log(1/\delta_A) \log^{3/2} \left( \frac{L^2}{\epsilon_A} \right) \log \log \left( \frac{L^2}{\epsilon_A} \right) \right),$$

by calling  $\text{QMonteCarlo}(e_{t,k}(X_t)e_{t,l}(X_t), \epsilon_A, \delta_A, L^2)$  from Theorem 13 for  $t \in [T-1]$  and  $k, l \in [m]$ . The computation of all  $b_t$  uses time

$$O \left( \frac{Tm}{\epsilon_b} RL (T\mathcal{T}_{\text{samp}} + T(\log(T)\mathcal{T}_z + m\mathcal{T}_e)) \log(1/\delta_b) \log^{3/2} \left( \frac{RL}{\epsilon_b} \right) \log \log \left( \frac{RL}{\epsilon_b} \right) \right),$$

by calling  $\text{QMonteCarlo}(Z_{\tilde{\tau}_t} e_{t-1,k}(X_{t-1}), \epsilon_b, \delta_b, RL)$  from Theorem 13 for  $t \in \{2, \dots, T\}$  and  $k \in [m]$ . Note that the term  $T(\log(T)\mathcal{T}_z + m\mathcal{T}_e)$  comes from using the unitaries  $C_t^{(k)}$  in  $\text{QMonteCarlo}$ , each with cost  $O(T(\log(T)\mathcal{T}_z + m\mathcal{T}_e))$  according to Lemma 14. Computing  $\mathbb{E}[Z_{\tilde{\tau}_1}]$  requires time

$$O \left( \frac{R}{\epsilon} (T\mathcal{T}_{\text{samp}} + T(\log(T)\mathcal{T}_z + m\mathcal{T}_e)) \log(1/\delta) \log^{3/2} \left( \frac{R}{\epsilon} \right) \log \log \left( \frac{R}{\epsilon} \right) \right),$$

by calling  $\text{QMonteCarlo}(Z_{\tilde{\tau}_1}, \epsilon, \frac{\delta}{2}, R)$  from Theorem 13 and where the term  $T(\log(T)\mathcal{T}_z + m\mathcal{T}_e)$  again comes from the unitaries  $C_t^{(k)}$  in  $\text{QMonteCarlo}$ . The classical computation of  $\{\tilde{A}_t^{-1}\}_{t=1}^{T-1}$  and  $\{\tilde{\alpha}_t = \tilde{A}_t^{-1}\tilde{b}_t\}_{t=1}^{T-1}$  requires time  $O(Tm^{\omega_*})$ , where  $2 \leq \omega_* < 3$ . Hence, by keeping the largest terms of each complexity, the final complexity is upper-bounded by

$$O \left( \frac{T^2 m^3}{\epsilon} \mathcal{T}_{\text{total}} L(L+R) \log(T) \log(Tm^2/\delta) \log^{3/2}(mL(L+R)/\epsilon) \log \log(mL(L+R)/\epsilon) \right).$$



We now move to the error analysis. Fix  $t \in [T - 1]$ . We start by bounding the error  $\|\tilde{\alpha}_t - \alpha_t\|_2$  between  $\alpha_t = A_t^{-1}b_t$  and  $\tilde{\alpha}_t = \tilde{A}_t^{-1}\tilde{b}_t$ . By using **QMonteCarlo** from Theorem 13 we approximate each entry of  $A_t$  and  $b_t$  as  $|(A_t)_{jl} - (\tilde{A}_t)_{jl}| \leq \epsilon/m$  and  $|(b_t)_j - (\tilde{b}_t)_j| \leq \epsilon/\sqrt{m}$  for all  $j, l \in [m]$ . All approximations hold with probability at least  $1 - \delta/2T$  by the union bound. This means that, with probability at least  $1 - \delta/2T$ ,

$$\|A_t - \tilde{A}_t\|_2 \leq \sqrt{\sum_{j,l=1}^m |(A_t)_{jl} - (\tilde{A}_t)_{jl}|^2} \leq \epsilon, \quad \|b_t - \tilde{b}_t\|_2 = \sqrt{\sum_{j=1}^m |(b_t)_j - (\tilde{b}_t)_j|^2} \leq \epsilon.$$

According to Theorem 18, we obtain

$$\|\tilde{\alpha}_t - \alpha_t\|_2 \leq \frac{2\epsilon}{\sigma_{\min}(A_t)} \left(1 + \frac{\|b_t\|_2}{\sigma_{\min}(A_t)}\right) \leq \frac{4\epsilon\sqrt{m}RL}{\sigma_{\min}^2}$$

with probability at least  $1 - \delta/2T$ , using that  $\sqrt{m}RL/\sigma_{\min} \geq 1$ . This, in turn, implies that

$$\|\tilde{\alpha}_t \cdot \vec{e}_t(X_t) - \alpha_t \cdot \vec{e}_t(X_t)\|_{L^2(\rho_t)} \leq \|\tilde{\alpha}_t - \alpha_t\|_2 \|\sqrt{\vec{e}_t(X_t) \cdot \vec{e}_t(X_t)}\|_{L^2(\rho_t)} \leq \frac{4\epsilon m RL^2}{\sigma_{\min}^2},$$

using that  $\|\sqrt{\vec{e}_t(X_t) \cdot \vec{e}_t(X_t)}\|_{L^2(\rho_t)} = \sqrt{\sum_{x \in E} \rho_t(x) \sum_{k=1}^m e_{t,k}^2(x)} \leq \sqrt{m}L$ . Next, we bound

$$\begin{aligned} & \|\tilde{\alpha}_t \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)} \\ & \leq \|\tilde{\alpha}_t \cdot \vec{e}_t(X_t) - \alpha_t \cdot \vec{e}_t(X_t)\|_{L^2(\rho_t)} + \|\alpha_t \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)} \\ & \leq \frac{4\epsilon m RL^2}{\sigma_{\min}^2} + \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)}, \end{aligned} \quad (12)$$

using that  $\alpha_t = \arg \min_{a \in \mathbb{R}^m} \mathbb{E}[(Z_{\tilde{\tau}_{t+1}} - a \cdot \vec{e}_t(X_t))^2]$  minimizes the least square estimator.

Finally, by the union bound, with probability at least  $1 - \delta$ , Eq. (12) holds for all  $t \in [T - 1]$ , together with  $|\tilde{Z}_{\tilde{\tau}_1} - \mathbb{E}[Z_{\tilde{\tau}_1}]| \leq \epsilon$ . Lemma 17 then leads to

$$|\tilde{Z}_{\tilde{\tau}_1} - \mathbb{E}[Z_{\tilde{\tau}_1}]| \leq \frac{8T\epsilon m RL^2}{\sigma_{\min}^2} + 2 \sum_{t=1}^{T-1} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)},$$

which implies Eq. (11) by using  $|\max\{a_0, a_1\} - \max\{a_0, a_2\}| \leq |a_1 - a_2|$  with  $a_0, a_1, a_2 \in \mathbb{R}$  on the definition of  $\tilde{\mathcal{U}}_0$  in Eq. (4).  $\blacktriangleleft$

Note that the approximation errors  $\min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)}$  appearing in Theorem 19 depend on the approximated stopping times  $\tilde{\tau}_{t+1}$ , which in turn depend on  $\mathcal{H}_{t'}$  for  $t' > t$ . It is possible to restate Theorem 19 in terms of  $\min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)}$ , which we do in the next theorem by following a similar approach to [93, Theorem 6.1]. The downside is that the time dependence now becomes exponential.

► **Theorem 20.** *Within the setting of Algorithm 2 with input parameters  $\delta$  and  $\epsilon$ , let  $T\mathcal{T}_{\text{samp}}$  be the sampling cost of the Markov chain and consider a set of linearly independent functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  for each  $t \in [T - 1]$  with  $L := \max_{t \in [T-1], k \in [m]} \|e_{t,k}\|_{L^2(\rho_t)}$  and query cost  $\mathcal{T}_e$ . Also consider  $\{z_t : E \rightarrow \mathbb{R}\}_{t=0}^T$  with  $R := \max_{t \in [T]} \|z_t\|_u < \infty$  and query cost  $\mathcal{T}_z$ . Moreover, let  $\sigma_{\min} := \min_{t \in [T-1]} \sigma_{\min}(A_t) > 0$ . Assume that  $\sqrt{m}RL/\sigma_{\min} \geq 1$  and define  $\mathcal{T}_{\text{total}} := \mathcal{T}_{\text{samp}} + \mathcal{T}_z + \mathcal{T}_e$ . Then, for any  $\delta \in (0, 1)$  and  $\epsilon \in (0, \sigma_{\min}/2]$ , Algorithm 2 outputs  $\tilde{\mathcal{U}}_0$  such that*

$$\Pr \left[ |\tilde{\mathcal{U}}_0 - \mathbb{E}[Z_{\tau_0}]| \geq 5^T \left( \frac{4\epsilon m RL^2}{\sigma_{\min}^2} + \max_{0 < t < T} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} \right) \right] \leq \delta$$

in time

$$O \left( \frac{T^2 m^3}{\epsilon} \mathcal{T}_{\text{total}} L(L + R) \log(T) \log(Tm^2/\delta) \log^{3/2}(mL(L + R)/\epsilon) \log \log(mL(L + R)/\epsilon) \right).$$



**Proof.** The proof follows the same steps of the proof of Theorem 19, with the further observation in Eq. (12) that if

$$\|\tilde{\alpha}_\ell \cdot \vec{e}_\ell(X_\ell) - \mathbb{E}[Z_{\tilde{\tau}_{\ell+1}} | X_\ell]\|_{L^2(\rho_\ell)} \leq \epsilon_0 + \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_\ell(X_\ell) - \mathbb{E}[Z_{\tilde{\tau}_{\ell+1}} | X_\ell]\|_{L^2(\rho_\ell)}, \quad (13)$$

for all  $\ell \in \{t, \dots, T-1\}$ , where we defined  $\epsilon_0 := \frac{4\epsilon m R L^2}{\sigma_{\min}^2}$ , then

$$2(T-\ell)\epsilon_0 + 2 \sum_{k=\ell}^{T-1} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)} \leq 5^{T-\ell}(\epsilon_0 + M_\ell^*), \quad (14)$$

for all  $\ell \in \{t, \dots, T-1\}$ , where  $M_\ell^* := \max_{k=\ell, \dots, T-1} (\min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)})$ . We prove this bound using backward induction as follows. Eq. (14) clearly holds for  $\ell = T-1$ . Assume it holds for  $\ell = t+1$ , we shall prove it is also true for  $\ell = t$ . First notice that, by the triangle inequality followed by Lemma 17 and then Eq. (13),

$$\begin{aligned} & \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)} \\ & \leq \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} + \|\mathbb{E}[Z_{\tau_{t+1}} | X_t] - \mathbb{E}[Z_{\tilde{\tau}_{t+1}} | X_t]\|_{L^2(\rho_t)} \\ & \leq \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} + 2 \sum_{k=t+1}^{T-1} \|\tilde{\alpha}_k \cdot \vec{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)} \\ & \leq \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} + 2 \sum_{k=t+1}^{T-1} (\epsilon_0 + \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)}) \\ & \leq M_t^* + 2(T-t-1)\epsilon_0 + 2 \sum_{k=t+1}^{T-1} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)}. \end{aligned}$$

Using the above inequality followed by the induction hypothesis,

$$\begin{aligned} & 2(T-t)\epsilon_0 + 2 \sum_{k=t}^{T-1} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)} \\ & \leq 2\epsilon_0 + 2M_t^* + 6(T-t-1)\epsilon_0 + 6 \sum_{k=t+1}^{T-1} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_k(X_k) - \mathbb{E}[Z_{\tilde{\tau}_{k+1}} | X_k]\|_{L^2(\rho_k)} \\ & \leq 2\epsilon_0 + 2M_t^* + 3 \cdot 5^{T-t-1}(\epsilon_0 + M_{t+1}^*) \\ & \leq 5^{T-t}(\epsilon_0 + M_t^*), \end{aligned}$$

proving the induction statement. The theorem follows by taking  $\ell = 0$  in Eq. (14) and using  $|\max\{a_0, a_1\} - \max\{a_0, a_2\}| \leq |a_1 - a_2|$ ,  $a_0, a_1, a_2 \in \mathbb{R}$ , on the definition of  $\tilde{\mathcal{U}}_0$  in Eq. (4). ◀

Given the above theorems, we prove a classical analogue.

► **Theorem 21.** *Within the setting of Algorithm 1, consider  $N$  independent sample paths with sample cost  $\mathcal{T}_{\text{samp}}$  and let the linearly independent functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  for each  $t \in [T-1]$  be such that  $L := \max_{t \in [T-1], k \in [m]} \|e_{t,k}\|_{L^2(\rho_t)}$  and have query cost  $\mathcal{T}_e$ . Also consider  $\{z_t : E \rightarrow \mathbb{R}\}_{t=0}^T$  with  $R := \max_{t \in [T]} \|z_t\|_u < \infty$  and query cost  $\mathcal{T}_z$ . Moreover, let  $\sigma_{\min} := \min_{t \in [T-1]} \sigma_{\min}(A_t) > 0$  and assume that  $\sqrt{m}RL/\sigma_{\min} \geq 1$ . Then, for any  $\epsilon \in (0, \sigma_{\min}/2]$ , Algorithm 1 runs for  $O(Tm^2N + Tm^{\omega_*} + TN(\mathcal{T}_{\text{samp}} + \mathcal{T}_z + m\mathcal{T}_e))$  time and returns an estimate  $\tilde{\mathcal{U}}_0$  such that*

$$\Pr \left[ \|\tilde{\mathcal{U}}_0 - \mathbb{E}[Z_{\tau_0}]\| \geq 5^T \left( \frac{4\epsilon m R L^2}{\sigma_{\min}^2} + \max_{0 < t < T} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} \right) \right] \leq 6m^2 e^{-2N\epsilon^2/m^2}.$$

**Proof.** The error analysis is very similar to that of Theorems 19 and 20, therefore we shall just point out the required changes. Each entry of  $A_t$  and  $b_t$  is approximated using a Chernoff bound, i.e.,  $\Pr[|(A_t)_{jl} - (\tilde{A}_t)_{jl}| \geq \epsilon/m] \leq 2e^{-2N\epsilon^2/m^2}$  and  $\Pr[|(b_t)_j - (\tilde{b}_t)_j| \geq \epsilon/\sqrt{m}] \leq 2e^{-2N\epsilon^2/m}$  for all  $j, l \in [m]$ . Moreover,  $\Pr[|\tilde{Z}_{\tau_1} - \mathbb{E}[Z_{\tau_1}]| \geq \epsilon] \leq 2e^{-2N\epsilon^2}$ . Therefore, by the union bound, all approximations hold with probability at least  $1 - 2m^2e^{-2N\epsilon^2/m^2} - 2me^{-2N\epsilon^2/m} - 2e^{-2N\epsilon^2} \geq 1 - 6m^2e^{-2N\epsilon^2/m^2}$ .

Regarding the time complexity, the most expensive computational steps are calculating the matrices  $\tilde{A}_t$ , which requires  $O(Tm^2N)$  time, and inverting them and computing the vectors  $\tilde{\alpha}_t$ , which requires  $O(Tm^{\omega_*})$  time. Sampling  $(X_t^{(1)}, \dots, X_t^{(N)})_{t=0}^T$  and querying  $(Z_t^{(1)}, \dots, Z_t^{(N)})_{t=0}^T$  and  $(e_{t,k}(X_t^{(1)}), \dots, e_{t,k}(X_t^{(N)}))_{t \in [T-1], k \in [m]}$  require  $O(NT(\mathcal{T}_{\text{samp}} + \mathcal{T}_z + m\mathcal{T}_e))$  time. All the other steps, computing  $\frac{1}{N} \sum_{n=1}^N Z_{\tau_{t+1}}^{(n)} \vec{e}(X_t^{(n)})$ ,  $\tilde{\mathcal{U}}_0$  and  $\tilde{\alpha}_t \cdot \vec{e}_t(X_t^{(n)})$  require  $O(TmN)$  or  $O(Tm)$  time.  $\blacktriangleleft$

If  $\mathcal{T}_{\text{samp}}, \mathcal{T}_z, \mathcal{T}_e = O(1)$ , then the complexity is  $O(Tm^2N + Tm^{\omega_*})$ . The factor  $Tm^2$  comes from computing  $\{A_t\}_{t=1}^{T-1}$  and accounts for runtime instead of only number of samples.

We summarize and compare the results from Theorems 21 and 20 into a single corollary.

**► Corollary 22.** *Within the setting of Algorithm 1 with input parameters  $\delta, N$  and Algorithm 2 with input parameters  $\delta, \epsilon_0$ , let  $T\mathcal{T}_{\text{samp}}$  be the sampling cost of the Markov chain and consider a set of linearly independent functions  $\{e_{t,k} : E \rightarrow \mathbb{R}\}_{k=1}^m$  for  $t \in [T-1]$  with query cost  $\mathcal{T}_e$  and  $L := \max_{t \in [T-1], k \in [m]} \|e_{t,k}\|_{L^2(\rho_t)}$ . Also consider  $\{z_t : E \rightarrow \mathbb{R}\}_{t=0}^T$  with query cost  $\mathcal{T}_z$  and  $R := \max_{t \in [T]} \|z_t\|_u < \infty$ . Let  $\sigma_{\min} \leq \min_{t \in [T-1]} \sigma_{\min}(A_t)$  be known. Assume that  $\sqrt{mRL}/\sigma_{\min} \geq 1$  and define  $\mathcal{T}_{\text{total}} := \mathcal{T}_{\text{samp}} + \mathcal{T}_z + \mathcal{T}_e$ . For any  $\delta \in (0, 1)$  and  $\epsilon \in (0, \frac{2mRL^2}{\sigma_{\min}}]$ , if  $\epsilon_0 = \frac{\epsilon\sigma_{\min}^2}{4mRL^2}$  and  $N = \lceil \frac{m^2}{2\epsilon_0^2} \log(6m^2/\delta) \rceil$ , then Algorithms 1 and 2 output  $\tilde{\mathcal{U}}_0$  such that*

$$\Pr \left[ |\tilde{\mathcal{U}}_0 - \mathbb{E}[Z_{\tau_0}]| \geq 5^T \left( \epsilon + \max_{0 < t < T} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} \right) \right] \leq \delta$$

using time, respectively,

$$O \left( \frac{Tm^6 R^2 L^4}{\epsilon^2 \sigma_{\min}^4} \mathcal{T}_{\text{total}} \log(m^2/\delta) \right)$$

and (up to log log factors)

$$O \left( \frac{T^2 m^4 RL^3(L+R)}{\epsilon \sigma_{\min}^2} \mathcal{T}_{\text{total}} \log(T) \log \left( \frac{Tm^2}{\delta} \right) \log^{3/2} \left( \frac{m^2 RL^3(L+R)}{\epsilon \sigma_{\min}^2} \right) \right).$$

**Proof.** The results concerning the quantum Algorithm 2 were already proven in Theorem 20, we just use that  $\epsilon_0 = \frac{\epsilon\sigma_{\min}^2}{4mRL^2}$ . It is left to prove the results about the classical Algorithm 1. Indeed, by setting  $N = \lceil \frac{m^2}{2\epsilon_0^2} \log(6m^2/\delta) \rceil$  into Theorem 21 with  $\epsilon_0 = \frac{\epsilon\sigma_{\min}^2}{4mRL^2}$ , we obtain

$$\Pr \left[ |\tilde{\mathcal{U}}_0 - \mathbb{E}[Z_{\tau_0}]| \geq 5^T \left( \epsilon + \max_{0 < t < T} \min_{a \in \mathbb{R}^m} \|a \cdot \vec{e}_t(X_t) - \mathbb{E}[Z_{\tau_{t+1}} | X_t]\|_{L^2(\rho_t)} \right) \right] \leq \delta.$$

The final time complexity becomes then  $O(Tm^2N + Tm^{\omega_*} + TN(\mathcal{T}_{\text{samp}} + \mathcal{T}_z + m\mathcal{T}_e)) = O \left( \frac{Tm^4}{\epsilon^2} (m^2 + \mathcal{T}_{\text{samp}} + \mathcal{T}_z + m\mathcal{T}_e) \frac{R^2 L^4}{\sigma_{\min}^4} \log \left( \frac{m^2}{\delta} \right) \right) = O \left( \frac{Tm^6 R^2 L^4}{\epsilon^2 \sigma_{\min}^4} \mathcal{T}_{\text{total}} \log \left( \frac{m^2}{\delta} \right) \right)$ .  $\blacktriangleleft$

**► Remark 23.** We note that in the previous theorem it is necessary to know a lower bound on  $\min_{t \in [T-1]} \sigma_{\min}(A_t)$  since it must be inputted into quantum subroutines for Monte Carlo to bound the variance. It is possible, though, to get around this detail by using the more complicated algorithm from [41] which does not require a bound on the variance.