

Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance

Dong An¹, Noah Linden², Jin-Peng Liu^{3,4,5}, Ashley Montanaro^{2,6}, Changpeng Shao², and Jiasu Wang¹

¹Department of Mathematics, University of California, Berkeley, CA 94720, USA

²School of Mathematics, Fry Building, University of Bristol, BS8 1UG, UK

³Joint Center for Quantum Information and Computer Science, University of Maryland, MD 20742, USA

⁴Institute for Advanced Computer Studies, University of Maryland, MD 20742, USA

⁵Department of Mathematics, University of Maryland, MD 20742, USA

⁶Phasecraft Ltd, Quantum Technologies Innovation Centre, Bristol BS1 5DD, UK

Inspired by recent progress in quantum algorithms for ordinary and partial differential equations, we study quantum algorithms for stochastic differential equations (SDEs). Firstly we provide a quantum algorithm that gives a quadratic speed-up for multilevel Monte Carlo methods in a general setting. As applications, we apply it to compute expectation values determined by classical solutions of SDEs, with improved dependence on precision. We demonstrate the use of this algorithm in a variety of applications arising in mathematical finance, such as the Black-Scholes and Local Volatility models, and Greeks. We also provide a quantum algorithm based on sublinear binomial sampling for the binomial option pricing model with the same improvement.

1 Introduction

Differential equations are ubiquitous throughout mathematics, science, and engineering. Specifically, ordinary differential equations (ODEs) and partial differential equations (PDEs) characterize continuous processes of systems arising extensively in many fields, from solid mechanics, fluid dynamics, and electromagnetism to biology [24]. Calculations of properties of such deterministic systems, typically require numerical schemes, that is one discretizes the differential equation in order to provide an approximate value of the quantity of interest [3].

For numerous systems arising in statistical physics, molecular dynamics, finance, and other real-world models, the dynamics is captured by a stochastic differential equation (SDE) [42, 50]. Given a typical SDE, a fundamental computational problem is to provide an expected value of a random variable Y , denoted $\mathbb{E}[Y]$, which is a functional determined by the solution of the SDE. Such a computational problem has been widely studied in mathematical finance, where the quantity Y represents the payoff in option and derivative pricing. It is often computationally expensive to estimate $\mathbb{E}[Y]$, since a scheme that approximates the SDE is necessarily run many times to average over the randomness. In this domain, Monte Carlo (MC) methods are basic tools with a provable complexity analysis.

Monte Carlo simulation, known for its flexibility and generality, refers to performing Monte Carlo methods in simulating SDEs [39, 42]. This method makes use of randomness to estimate $\mathbb{E}[Y]$ of a SDE as introduced above. In general, Monte Carlo simulation generates k independent approximate samples from Y by performing a chosen scheme, and then outputs an average of those k outputs as an approximate expectation of Y . Assuming the variance of Y is bounded by σ^2 , according to Chebyshev's inequality, it suffices to utilize $k = O(\sigma^2/\epsilon^2)$ samples to estimate $\mathbb{E}[Y]$, where ϵ is the additive error [39].

Simulating SDEs using Monte Carlo is very useful when the cost of each sample is cheap. A typical example is the Geometric Brownian Motion (GBM) in the Black-Scholes (BS) model [6, 7, 37], which characterizes stock prices in the financial market. Classical algorithms are designed to directly sample from the solution of GBM, whose explicit form is known, rather than simulating the paths of the GBM itself. Assuming the cost of sampling the solution is ignorable, i.e. $O(1)$, the computational complexity equals the number of samples, $O(1/\epsilon^2)$. However, a challenge faced by Monte Carlo occurs when creating each sample is costly, so even quadratic dependence on ϵ may result in an computational complexity significantly larger than $O(1/\epsilon^2)$ in practice. More concretely, we consider a general SDE without an explicit solution. To compute an approximate solution by numerical methods, we discretize the SDE on the time interval $[0, T]$ with the step size h , and perform a scheme of strong order r (which produces a random variable $\hat{Y}$ that is ϵ -close to Y , where $\epsilon = O(h^r)$) [42]. Taking the Milstein scheme with strong order 1 as an example, i.e. $r = 1$, then $T/h = \Omega(1/\epsilon)$ number of iterations is required to produce one sample. Under this circumstance, the computational complexity of classical Monte Carlo simulation is $O(1/\epsilon^3)$ in total. Generally, when performing a scheme of strong order r , the complexity can be improved to $O(1/\epsilon^{2+1/r})$. Usually, it is challenging to perform a scheme with large r , due to the higher smoothness requirement of the SDE, and, in practice, it is harder to implement explicit forms of higher strong order schemes [11, 42, 43]. So the acceleration is moderate in practice.

To reduce such an expensive computational cost, multilevel Monte Carlo (MLMC) methods have attracted very considerable attention recently, and have successfully been applied to simulate SDEs with applications in finance [28, 29]. Recalling the goal of estimating $\mathbb{E}[Y]$ of a general random variable Y , given a sequence of estimators $P_0, P_1, \dots, P_L$ that approximates Y with increasing accuracy and cost, multilevel Monte Carlo aims to estimate $\mathbb{E}[P_L]$ by simulating a sum of $\mathbb{E}[P_l - P_{l-1}]$, with different numbers of samples at each level l . The main improvement of multilevel Monte Carlo is from reducing the number of samples when the variance of $P_l - P_{l-1}$ is large. By balancing the sample numbers and variances for different $P_l - P_{l-1}$ and summing them together, multilevel Monte Carlo gives an optimal overall cost of estimating $\mathbb{E}[P_L]$ that approximates $\mathbb{E}[Y]$ within the mean-squared error ϵ^2 . As described by Giles et al, multilevel Monte Carlo with a scheme of strong order $r > 1$, is capable of estimating $\mathbb{E}[Y]$ of general SDEs with the overall cost $\tilde{O}(1/\epsilon^2)$ ([29, Theorem 1]), where $\tilde{O}$ neglects logarithmic factors. Compared to the standard Monte Carlo simulation with the same scheme, it removes a $1/\epsilon^{1/r}$ factor in the overall complexity. Moreover, this approach does not require the use of higher strong order schemes, and hence avoids the smoothness requirement and the implementation difficulty as well. We note that when the samples are allowed not to be random and independent, but based on certain lattice rules, it could lead to a $O(1/\epsilon^p)$ cost with $p < 2$ under certain conditions [21, 31]. In our case we assume the samples are chosen randomly and independently, then $O(1/\epsilon^2)$ is the best known complexity that classical algorithm can achieve. [29] However such a quadratic dependence on $1/\epsilon$ of the overall complexity that

multilevel Monte Carlo can achieve is still far from ideal, particularly when applications to mathematical finance are considered.

Quantum computers are expected to outperform classical computers for solving a system of linear equations [1, 2, 13, 32, 34, 45, 58, 59] and differential equations [4, 5, 12, 14–16, 18, 23, 46, 48, 52, 60, 61]. Quantum algorithms for certain stochastic differential equations, such as simulating GBM of the Black-Scholes model as discussed above, have attracted increasing attention in quantum computational finance [8, 26, 33, 53, 55]. Reference [33] claims an exponential speedup over classical algorithms to solve the Black-Scholes PDE, but does not include a detailed complexity analysis; it uses a very different approach to that used here, which does not seem to be easily extendible to general SDEs. For the payoff models, using quantum-accelerated Monte Carlo methods [51] based on sampling from the explicit solution of GBM, quantum algorithms can approximate the expected value of the price of portfolio within error ϵ in complexity $\tilde{O}(1/\epsilon)$, a quadratically improved dependence on ϵ compared to classical Monte Carlo simulation [41, 53, 55, 57]. However, previous quantum algorithms, that sample the explicit solution of GBM, cannot be extended to simulate general SDEs with no explicit formula for the solution. Reference [40] presented a practical quantum circuit for simulating the Local Volatility (LV) model, which generalizes the GBM and does not have an explicit solution [20, 22]. However, reference [40] did not provide a concrete complexity for simulating the LV model. In summary, quantum speedups for payoff models of general SDEs lacking explicit solutions are far from well established.

In this paper, we provide quantum algorithms for approximating classical outputs determined by general SDEs (i.e. ones with no explicit formulas for the solutions) with computational complexity $\tilde{O}(1/\epsilon)$. We will apply these to several payoff models of SDEs which arise in finance, and the outputs are the payoffs in pricing. To achieve such an improvement, we first propose a quantum-accelerated multilevel Monte Carlo (QA-MLMC) method in a general setting, and then apply QA-MLMC for general SDEs. Compared to the classical counterpart, QA-MLMC achieves a quadratic speedup in precision up to a logarithmic factor. The main ingredient to this acceleration is the quantum speedup of the Monte Carlo method [51]. Roughly, to approximate $\mathbb{E}[P_L - P_{L-1}]$ in the MLMC approach as we discussed above, we only need to use $\tilde{O}(1/\epsilon)$ samples. We shall prove that this speedup is preserved in the telescoping sum $\sum_{l=0}^L \mathbb{E}[P_l - P_{l-1}]$ to approximate $\mathbb{E}[P_L]$. We remark that a somewhat similar idea was used in [51] for the special case of computing a partition function as a telescoping *product* of terms, each of which is approximated using QA-MC.

Instead of the mean-squared error considered in the classical case, QA-MLMC only returns an approximation of $\mathbb{E}[P_L]$ in the sense of the additive error. However, this will not incur any unfair comparison between MLMC and QA-MLMC since the two types of errors are almost equivalent (see Appendix A). Because of the different types of errors, MLMC and QA-MLMC have slight differences in their assumptions. This will become clear below when we apply QA-MLMC to solve practical problems in finance.

As discussed above, to solve a general SDE with no explicit solutions, usually we need a discretization scheme of strong order r . Classically, to apply MLMC, a numerical scheme of strong order $r > 1$ is usually sufficient to obtain a complexity of $\tilde{O}(1/\epsilon^2)$. However, in the quantum case, to ensure a quadratic speedup, i.e. to obtain a complexity of $\tilde{O}(1/\epsilon)$, we have to apply a numerical scheme of strong order $r > 2$. Note that when an order r scheme is used, the Monte Carlo method can solve the SDE with a complexity of $\tilde{O}(1/\epsilon^{2+1/r})$. And a direct corollary of [51] shows that in the quantum case this method can be improved to has a complexity of $\tilde{O}(1/\epsilon^{1+1/r})$. However, for the reasons discussed above we cannot

choose r as large as we want. So QA-MLMC still has advantages over QA-MC both in theory and in practice.

	Algorithm	Model	Result
Classical	MC with direct sampling [7, 37]	Black-Scholes model	ϵ^{-2}
	MC with scheme of strong order r (Proposition 1)	payoff models of general SDEs	$\epsilon^{-2-1/r}$
	MLMC with scheme of strong order $r > 1$ (Proposition 3)	payoff models of general SDEs	ϵ^{-2}
	MC with binomial sampling (Proposition 4)	binomial option pricing model	ϵ^{-2}
Quantum	QA-MC with direct sampling [53, 55, 57]	Black-Scholes model	ϵ^{-1}
	QA-MC with scheme of strong order r (Theorem 1)	payoff models of general SDEs	$\epsilon^{-1-1/r}$
	QA-MLMC with scheme of strong order $r > 2$ (Theorem 3)	payoff models of general SDEs	ϵ^{-1}
	QA-MC with binomial sampling (Theorem 4)	binomial option pricing model	ϵ^{-1}

Table 1: Summary of the time complexities of classical and quantum algorithms for financial models with the additive error ϵ , in which logarithmic factors are omitted.

As applications, we apply QA-MLMC to solving various payoff models, which satisfy our smoothness requirements and are of great interest in mathematical finance. Examples include the well-known Black-Scholes model that prices a variety of financial derivatives; the Local Volatility model that generalizes the Black-Scholes model by treating volatility as a function of the asset and the time; the Greeks that label the sensitivity of the price of the option for hedge portfolios [37]; and the binomial option pricing model as introduced above.

For the analytically solvable Black-Scholes model, in which QA-MC has been applied to reduce the complexity from $\tilde{O}(\epsilon^{-2})$ to $\tilde{O}(\epsilon^{-1})$ [55], we verify QA-MLMC is able to achieve the same quantum speedup. For the rest of the models, we establish the first quantum acceleration to achieve the complexity $\tilde{O}(\epsilon^{-1})$, by applying QA-MLMC to the Local Volatility model and Greeks. Table 1 compares the performance of our approaches to classical ones for various financial models with respect to the dependence on error tolerance ϵ .

Furthermore, we also study the Black-Scholes model with European and digital payoffs in numerical experiments. We provide concrete numerical implementations of the schemes of strong order up to 3 to test those parameters with different payoff functions, and numerical results are in good agreement with our theoretical estimates. This provides evidence that our complexities for MLMC and QA-MLMC are reasonable and sharp.

We also consider the binomial option pricing model (BOPM), also known as the binomial lattice model (BLM), which provides an alternative option pricing model different from the Black-Scholes model, by constructing a binomial tree with the same expectation and variance as the Geometric Brownian Motion [19, 37, 56]. Inspired by the binomial structure, it is natural to simulate BOPM by performing sublinear binomial sampling, a recently developed technique that samples a binomial tree in sublinear time [9, 25]. This technique has been applied to construct a fast random walk, which is a specific binomial tree, to develop fast classical and quantum algorithms for heat equation [46]. Observing that sublinear binomial sampling provides a numerical method to output an estimate of Y with ignorable cost per sample, we can estimate $\mathbb{E}[Y]$ preserving the computational complexity the same as the sampling complexity, neglecting logarithmic factors. Therefore, we propose classical and quantum algorithms for BOPM, with the dependence on precision $\tilde{O}(1/\epsilon^2)$ and $\tilde{O}(1/\epsilon)$, respectively. Complementing the multilevel Monte Carlo methods, this provides an alternative method for efficiently studying properties of struc-

tured stochastic models in mathematical finance.

Overall, our theoretical and numerical results provide promise for potential applications of quantum computing in computational finance.

The rest of this paper is structured as follows. [Section 2](#) introduces the payoff problem of SDEs that we study, and quantum-accelerated Monte Carlo for solving SDEs. [Section 3](#) states the main theorem for the quantum-accelerated multilevel Monte Carlo method. [Section 4](#) applies the quantum-accelerated multilevel Monte Carlo method to the SDE problem. [Section 5](#) presents the Black-Scholes model as an application for estimating the price of the option. [Section 6](#) generalizes the Black-Scholes model to the Local Volatility model. [Section 7](#) introduces the Greeks as an application for estimating the sensitivity of the price. [Section 8](#) covers the binomial lattice model as an alternative option pricing model. [Section 9](#) includes a discussion and raises some open problems. [Appendix A](#) discusses the relationship between mean square error and additive error. Finally, [Appendix B](#) presents numerical results of several schemes for the Black-Scholes model with European and digital payoffs.

2 Payoff models of general SDEs

2.1 Problem settings

Suppose that we have a stochastic differential equation (SDE) with general drift and volatility terms

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t, \quad (2.1)$$

for $t \in [0, T]$, where $X_t \in \mathbb{R}$ is an Itô process, W_t is a standard Brownian motion. The payoff problem we are concerned with is as follows.

Problem 1. *Assuming there exists an oracle O_I that samples from an initial distribution π_0 to produce X_0 as an initial condition, an oracle O_W that samples from the Brownian motion W_t , and an oracle O_P that produces the payoff $\mathcal{P}(X)$ as a functional of a specific X , and oracles O_μ and O_σ that produce $\mu(X, t)$ and $\sigma(X, t)$ as functions of X and t , respectively.*

Given an evolution time $T > 0$, we aim to compute

$$\mathbb{E}[\mathcal{P}(X_T) \mid X_0 \in \pi_0], \quad (2.2)$$

within an error $\epsilon > 0$, where X_T is generated by (2.1).

[Problem 1](#) is widely investigated in mathematical finance. Different kinds of financial models, such as option pricing and Greeks, can be formulated as [Problem 1](#) with various assumptions, which are introduced in detail in [Section 4](#). Note that if $\pi_0 = \delta_{X_0}$, then $X \in \pi_0 \rightarrow X = X_0$, and [Problem 1](#) is reduced to a deterministic initial value problem.

Before we proceed, it is helpful to clarify the meaning of the errors. By saying to compute $\mathbb{E}[\mathcal{P}]$ within an error $\epsilon > 0$ in [Problem 1](#), there are two different scenarios (here the random variable Y is an estimator of $\mathbb{E}[\mathcal{P}]$):

- the mean-squared error $\mathbb{E}(Y - \mathbb{E}[\mathcal{P}])^2$ is bounded by ϵ^2 ,
- the additive error $|Y - \mathbb{E}[\mathcal{P}]|$ is bounded by ϵ with probability at least 0.99.

Most research on classical Monte Carlo simulation uses mean-squared error, whereas research on the quantum-accelerated Monte Carlo method typically uses additive error. To

reduce technical difficulty and be consistent with existing literature, we will follow this convention, using mean-squared error in our classical algorithms and using additive error in our quantum algorithms. We note however that this still allows fair comparison between classical and quantum algorithms, because these two types of error bounds are indeed almost equivalent, which is elaborated in detail in [Appendix A](#).

2.2 Monte Carlo method

For simplicity we assume that the costs of querying O_I , O_W , O_P , O_μ and O_σ are $O(1)$. For our quantum algorithms, we assume that the oracles can produce coherent superpositions corresponding to these distributions. This assumption can usually be satisfied, for example given a classical algorithm that generates samples based on uniformly random bits; see [51] for a discussion. In particular, note that we will not actually need to put the initial distribution (which may be quite complicated) into superposition – the algorithm will compute $\mathbb{E}[\mathcal{P}(X_T)]$ for a given starting position X_0 , and sampling X_0 can be performed classically. If a sample of X_T generated by (2.1) can be obtained with cost $O(1)$, the computational complexity of solving [Problem 1](#) equals the total number of samples required to estimate $\mathbb{E}[\mathcal{P}(X_T)]$ within ϵ . While for general SDEs, the cost of simulating (2.1) to calculate X_T each time should be taken into account.

For instance, we consider the widely used Milstein discretization with the step size h , giving

$$\hat{X}_{k+1} = \hat{X}_k + \mu(\hat{X}_k, t_k)h + \sigma(\hat{X}_k, t_k)\Delta W_k + \frac{1}{2}\sigma(\hat{X}_k, t_k)\partial_X\sigma(\hat{X}_k, t_k)((\Delta W_k)^2 - h), \quad (2.3)$$

where $k \in [n]_0$, where $n = T/h$. This has computational cost $O(1/h)$ to produce a sample $\hat{X}_n$ that approximates X_T . Based on Monte Carlo methods [39, 42], a simple estimation for $\mathbb{E}[\mathcal{P}(X_T)]$ is to generate N samples, each independently outputs $\mathcal{P}(X_T)$, and then to produce an average,

$$Y = \frac{1}{N} \sum_{i=1}^N \mathcal{P}(\hat{X}_n^{(i)}). \quad (2.4)$$

Note that the mean-squared error can be decomposed as

$$\mathbb{E}|Y - \mathbb{E}[\mathcal{P}(X_T)]|^2 = \mathbb{V}[Y] + |\mathbb{E}[Y] - \mathbb{E}[\mathcal{P}(X_T)]|^2, \quad (2.5)$$

which can be bounded by $O(N^{-1} + h^2)$. Here we let $\mathbb{V}[Y]$ denote the variance of a random variable Y . If we seek to bound the mean-squared error by ϵ^2 , then we can take $N = O(1/\epsilon^2)$, $h = O(\epsilon)$, so the expected computational cost equals $O(N/h) = O(\epsilon^{-3})$. This result is first stated in [28]. In general, for high order schemes, the complexity of classical Monte Carlo method can be bounded as follows.

Proposition 1. *Let Y be an estimator of $\mathbb{E}[\mathcal{P}(X_T)]$ with bounded variance, based on a numerical discretization with time step size h using a numerical scheme with strong order r (defined in (4.2)). Then in order to achieve the accuracy of mean-squared error ϵ^2 , the computational complexity of the Monte Carlo method is $O(\epsilon^{-2-1/r})$.*

Proof. For simplicity we assume $\mathcal{P}$ is globally Lipschitz continuous (we refer to [Section 4](#) for a more general analysis). To bound the mean-squared error, we only need to bound the right hand side of the equation (2.5). Note that

$$\mathbb{V}[Y] = \frac{1}{N} \mathbb{V}[\mathcal{P}(\hat{X}_T)] = O(N^{-1}), \quad (2.6)$$

and

$$|\mathbb{E}[Y] - \mathbb{E}[\mathcal{P}(X_T)]|^2 = |\mathbb{E}[\mathcal{P}(\hat{X}_n)] - \mathbb{E}[\mathcal{P}(X_T)]|^2 \leq \mathbb{E}|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)|^2 = O(h^{2r}). \quad (2.7)$$

Therefore

$$\mathbb{E}|Y - \mathbb{E}[\mathcal{P}(X_T)]|^2 = O(N^{-1} + h^{2r}). \quad (2.8)$$

In order to bound the mean-squared error by ϵ^2 , it suffices to choose $N \sim \epsilon^{-2}$ and $h \sim \epsilon^{1/r}$, thus the complexity becomes $O(N/h) = O(\epsilon^{-2-1/r})$. $\square$

Proposition 1 tells us that in the classical Monte Carlo method, the total complexity can be indeed improved by using a higher order scheme, but it is bottlenecked by $O(\epsilon^{-2})$ due to the variance of the estimator. To minimize the overall computational cost without needing higher order schemes, we will introduce an advanced Monte Carlo approach, known as multilevel Monte Carlo [28], in [Section 3](#).

However before doing so, we show that quantum-accelerated Monte Carlo, as described in [51] can give a speed up of non-multilevel methods.

2.3 Quantum-accelerated Monte Carlo method

In [51], Montanaro showed that using a quantum computer, the number of samples used in Monte Carlo can be reduced quadratically.

Lemma 1 (Theorem 5 of [51]). *Let $\mathcal{A}$ be a (classical or quantum) algorithm. Let $v(\mathcal{A})$ be the random variable corresponding to $v(x)$ when the outcome of $\mathcal{A}$ is x . Assume that $\mathbb{V}[v(\mathcal{A})] \leq \sigma^2$, then there is a quantum algorithm that estimates $\mathbb{E}[v(\mathcal{A})]$ up to additive error ϵ with success probability at least $2/3$ by using*

$$O\left((\sigma/\epsilon)(\log \sigma/\epsilon)^{3/2}(\log \log \sigma/\epsilon)\right) \quad (2.9)$$

samples.

The powering lemma stated below can increase the success probability of [Lemma 1](#) to $1 - \delta$ for any arbitrarily small δ .

Lemma 2 (Lemma 1 of [51]). *Let $\mathcal{A}$ be a (classical or quantum) algorithm which aims to estimate some quantity μ , and whose output $\tilde{\mu}$ satisfies $|\mu - \tilde{\mu}| \leq \epsilon$ except with probability γ , for some fixed $\gamma < 1/2$. Then, for any $\delta > 0$, it suffices to repeat $\mathcal{A}$ $O(\log 1/\delta)$ times and take the median to obtain an estimate which is accurate to within ϵ with probability at least $1 - \delta$.*

In [Lemma 1](#) and [Lemma 2](#), the randomized or quantum algorithm $\mathcal{A}$ is used to produce a random variable X and then compute the payoff function $\mathcal{P}(X) \in \mathbb{R}$. For the detailed implementation of $\mathcal{A}$, we refer to the start of Section 2 of [51] for more details.

Based on [Lemma 2](#), the success probability of [Lemma 1](#) can be improved to $1 - \delta$ by using $O((\sigma/\epsilon)(\log \sigma/\epsilon)^{3/2}(\log \log \sigma/\epsilon)(\log 1/\delta))$ samples. Thus, we are able to develop a quantum-accelerated Monte Carlo method for [Problem 1](#).

Theorem 1. *Let $\mathcal{A}$ be an algorithm that generates a sample of numerical solution $\hat{X}_n$ of the SDE using a numerical discretization with time step size h using r -th order scheme in the sense that $\mathbb{E}|\hat{X}_n - X_T| = O(h^r)$. Assume that $\mathcal{P}(\hat{X}_n)$ has bounded variance independent of h . Then there exists a quantum algorithm that achieves the accuracy of additive error ϵ with probability at least 0.99, with computational complexity $\tilde{O}(\epsilon^{-1-1/r})$.*

Proof. Similarly to [Proposition 1](#), for technical simplicity we assume $\mathcal{P}$ is globally Lipschitz continuous (we refer to [Section 4](#) for analysis on more general payoff functions). By [Lemma 1](#) and [Lemma 2](#), there exists a quantum algorithm that generates an estimator Y such that

$$|Y - \mathbb{E}[\mathcal{P}(\hat{X}_n)]| < \epsilon/2$$

with probability at least 0.99 using $\tilde{O}(\epsilon^{-1})$ queries to $\mathcal{A}$. Furthermore,

$$|\mathbb{E}[\mathcal{P}(\hat{X}_n)] - \mathbb{E}[\mathcal{P}(X_T)]| \leq \mathbb{E}|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)| = O(h^r). \quad (2.10)$$

Hence we can choose $h = O(\epsilon^{-1/r})$ to bound $|\mathbb{E}[\mathcal{P}(\hat{X}_n)] - \mathbb{E}[\mathcal{P}(X_T)]|$ by $\epsilon/2$, and the complexity of each query to $\mathcal{A}$ becomes $O(\epsilon^{-1/r})$. Combining the above two estimates, in order to bound the additive error $|Y - \mathbb{E}[\mathcal{P}(X_T)]|$ by ϵ with probability at least 0.99, the total complexity is $\tilde{O}(\epsilon^{-1-1/r})$. $\square$

3 Quantum-accelerated MLMC

3.1 Multilevel Monte Carlo method

Heinrich [\[35\]](#) developed the first work on multilevel Monte Carlo (MLMC) methods for parametric integration, then Giles [\[28\]](#) introduced MLMC to simulate SDEs. In the following, we first briefly introduce this method. Then we show how to accelerate this method using a quantum-accelerated Monte Carlo method [\[51\]](#). For more about MLMC, we refer to the survey paper [\[29\]](#).

Instead of focusing on SDEs, we review the idea of MLMC in the general setting. Now let P be a random variable, our goal is to estimate $\mathbb{E}[P]$, given a sequence $P_0, P_1, \dots, P_L$ that approximates P with increasing accuracy, but also increasing cost. For instance, for the SDE [\(2.1\)](#), P is the payoff, $P_l = \mathcal{P}(\hat{X}_{n_l})$ with $n_l = T/h_l = 2^l T$. Now we have the following telescoping sum

$$\mathbb{E}[P_L] = \sum_{l=0}^L \mathbb{E}[P_l - P_{l-1}], \quad (3.1)$$

where $P_{-1} = 0$. We can estimate $\mathbb{E}[P_L]$ by using the Monte Carlo method to approximate each term $\mathbb{E}[P_l - P_{l-1}]$. So we obtain the following approximation of $\mathbb{E}[P_L]$

$$Y = \sum_{l=0}^L Y_l, \quad \text{where } Y_l := \frac{1}{N_l} \sum_{i=0}^{N_l} (P_l^{(l,i)} - P_{l-1}^{(l,i)}). \quad (3.2)$$

The superindex l means that the samples are generated independently.

Note that for SDE, $P_l - P_{l-1}$ comes from two discrete approximations with different timesteps but the same Brownian path. To generate random samples $P_l^{(l,i)} - P_{l-1}^{(l,i)}$, one method suggested in [\[28\]](#) is as follows. First constructing the Brownian increments for the simulation of the discrete path leading to the evaluation of $P_l^{(l,i)}$. Then summing them in groups of size 2 to give the discrete Brownian increments for the evaluation of $P_{l-1}^{(l,i)}$.

To determine the cost, the MLMC approach considers the mean-squared error,

$$\mathbb{E}(Y - \mathbb{E}[P])^2, \quad (3.3)$$

which can be decomposed as follows

$$\mathbb{E}(Y - \mathbb{E}[P])^2 = \mathbb{V}[Y] + (\mathbb{E}[P_L] - \mathbb{E}[P])^2 \leq \mathbb{V}[Y] + \mathbb{E}[P_L - P]^2. \quad (3.4)$$

In order to achieve $\mathbb{E}(Y - \mathbb{E}[P])^2 \leq \epsilon^2$, it is sufficient to ensure that $\mathbb{V}[Y] \leq \epsilon^2/2$ and $\mathbb{E}[P_L - P]^2 \leq \epsilon^2/2$. For $l \geq 0$, let C_l, V_l be the cost and variance of one sample of $P_l - P_{l-1}$. Then the overall cost and variance of Y given in (3.2) is $\sum_{l=0}^L N_l C_l$ and $\sum_{l=0}^L N_l^{-1} V_l$.

To minimize the cost with a fixed variance $\epsilon^2/2$, we introduce the Lagrange multiplier λ^2 and minimize

$$\sum_{l=0}^L N_l C_l + \lambda^2 \left(\frac{\epsilon^2}{2} - \sum_{l=0}^L N_l^{-1} V_l \right). \quad (3.5)$$

This leads to $N_l = \lambda \sqrt{V_l/C_l}$ and $\lambda = 2\epsilon^{-2} \sum_{l=0}^L \sqrt{V_l C_l}$. The total computational cost is

$$2\epsilon^{-2} \left(\sum_{l=0}^L \sqrt{V_l C_l} \right)^2. \quad (3.6)$$

We restate the rigorous argument in [29] as follows.

Lemma 3 (Theorem 1 of [29]). *Let P be a random variable and P_l be the corresponding level l numerical approximation. Let Y_l be an approximation of $\mathbb{E}[P_l - P_{l-1}]$ based on Monte Carlo method such that the expected cost and variance of one sample is C_l and V_l respectively. If there exist positive constants α, β, γ such that $\alpha \geq \frac{1}{2} \min(\beta, \gamma)$ and*

- $|\mathbb{E}[P_l - P]| = O(2^{-\alpha l})$,
- $\mathbb{E}[Y_l] = \mathbb{E}[P_l - P_{l-1}]$, $l \geq 0$, where $P_{-1} = 0$,
- $V_l = O(2^{-\beta l})$,
- $C_l = O(2^{\gamma l})$,

then for any $\epsilon < 1/e$ there exists an L such that $Y = \sum_{l=0}^L Y_l$ has a mean-squared error with bound $\mathbb{E}(Y - \mathbb{E}[P])^2 \leq \epsilon^2$. Moreover, the total computational cost is

$$\begin{cases} O(\epsilon^{-2}), & \beta > \gamma, \\ O(\epsilon^{-2}(\log \epsilon)^2), & \beta = \gamma, \\ O(\epsilon^{-2-(\gamma-\beta)/\alpha}), & \beta < \gamma. \end{cases} \quad (3.7)$$

3.2 Quantum-accelerated multilevel Monte Carlo method

Recall that in MLMC, in the telescoping sum (3.1), the Monte Carlo method is utilized to approximate each mean value $\mathbb{E}[P_l - P_{l-1}]$, in which we can obtain a quadratic speedup using a quantum computer. To approximate the telescoping sum (3.1) via (3.2), we have a classical algorithm to do the sampling, and thus the assumption of the existence of the classical algorithm to do the sampling is satisfied for MLMC in Lemma 1. The proof of the following theorem is similar to that of Lemma 3.

Theorem 2. *Let P denote a random variable, and let P_l ($l = 0, 1, \dots, L$) denote a sequence of random variables such that P_l approximates P at level l . Further define $P_{-1} = 0$. Let C_l be the cost of sampling from P_l , and let V_l be the variance of $P_l - P_{l-1}$. If there exist positive constants $\alpha, \beta = 2\hat{\beta}, \gamma$ such that $\alpha \geq \min(\hat{\beta}, \gamma)$ and*

- $|\mathbb{E}[P_l - P]| = O(2^{-\alpha l})$,
- $V_l = O(2^{-\beta l}) = O(2^{-2\hat{\beta} l})$,
- $C_l = O(2^{\gamma l})$,

then for any $\epsilon < 1/e$ there is a quantum algorithm that estimates $\mathbb{E}[P]$ up to additive error ϵ with probability at least 0.99, and with cost

$$\begin{cases} O\left(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2\right), & \hat{\beta} > \gamma, \\ O\left(\epsilon^{-1}(\log 1/\epsilon)^{7/2}(\log \log 1/\epsilon)^2\right), & \hat{\beta} = \gamma, \\ O\left(\epsilon^{-1-(\gamma-\hat{\beta})/\alpha}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2\right), & \hat{\beta} < \gamma. \end{cases} \quad (3.8)$$

Proof. In MLMC, we use the telescoping sum $\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^L \mathbb{E}[P_l - P_{l-1}]$ to estimate $\mathbb{E}[P_L]$. Let Y_l be the approximation of $\mathbb{E}[P_l - P_{l-1}]$ obtained by the quantum-accelerated Monte Carlo method (see Lemma 1). By Lemma 1 and Lemma 2, for any $\epsilon_l \geq 0$, to make sure $|\mathbb{E}[P_l - P_{l-1}] - Y_l| \leq \epsilon_l$ with probability at least $1 - \delta$, we need $N_l = O((2^{-\hat{\beta}l}/\epsilon_l)(\log 2^{-\hat{\beta}l}/\epsilon_l)^{3/2}(\log \log 2^{-\hat{\beta}l}/\epsilon_l)(\log 1/\delta))$ samples. The error in approximating $\mathbb{E}[P_L]$ with $Y := \sum_{l=0}^L Y_l$ satisfies $|Y - \mathbb{E}[P_L]| \leq \sum_{l=0}^L |\mathbb{E}[P_l - P_{l-1}] - Y_l| \leq \sum_{l=0}^L \epsilon_l$. As a result, the error in approximating $\mathbb{E}[P]$ satisfies the bound

$$|Y - \mathbb{E}[P]| \leq |\mathbb{E}[P_L] - \mathbb{E}[P]| + |\mathbb{E}[P_L] - Y| \leq |\mathbb{E}[P_L] - \mathbb{E}[P]| + \sum_{l=0}^L \epsilon_l \quad (3.9)$$

with probability at least $(1 - \delta)^{L+1}$. The total cost equals $C = \sum_{l=0}^L C_l N_l$.

Choose

$$L = \left\lceil \frac{\log(2\epsilon^{-1})}{\alpha} \right\rceil \quad (3.10)$$

so that $2^{-\alpha L} \leq \epsilon/2$. This ensures the first error term of (3.9) is bounded by $\epsilon/2$. As for the second term, we choose different N_l based on the values $\epsilon_l, \hat{\beta}, \gamma$. Choosing $\delta = 1/(100(L+1))$, then we can make sure that the success probability is at least $(1 - \delta)^{L+1} \geq e^{-1/100} > 0.99$.

We now split into cases to analyse and optimise the complexity of this approach. In the following analysis, for notational convenience we just write $N_l = \lceil 2^{-\hat{\beta}l}/\epsilon_l \rceil$. But in the end, the cost should be multiplied by $(\log 1/\delta)(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)$ for all cases.

(a). If $\hat{\beta} > \gamma$, then choose $\epsilon_l = \frac{\epsilon}{2}(1 - 2^{-(\hat{\beta}-\gamma)/2})2^{-(\hat{\beta}-\gamma)l/2}$. The second error term of (3.9) is bounded by

$$\frac{\epsilon}{2}(1 - 2^{-(\hat{\beta}-\gamma)/2}) \sum_{l=0}^L 2^{-(\hat{\beta}-\gamma)l/2} = \frac{\epsilon}{2}(1 - 2^{-(\hat{\beta}-\gamma)/2}) \frac{1 - 2^{-(\hat{\beta}-\gamma)(L+1)/2}}{1 - 2^{-(\hat{\beta}-\gamma)/2}} < \frac{\epsilon}{2}. \quad (3.11)$$

Also we have

$$N_l = 2\epsilon^{-1}(1 - 2^{-(\hat{\beta}-\gamma)/2})^{-1}2^{-(\hat{\beta}+\gamma)l/2} + 1, \quad (3.12)$$

where the “+1” term is caused by the ceiling function. The cost is bounded by

$$\begin{aligned} \sum_{l=0}^L N_l C_l &= O\left(\sum_{l=0}^L \left(\epsilon^{-1}(1 - 2^{-(\hat{\beta}-\gamma)/2})^{-1}2^{-(\hat{\beta}+\gamma)l/2} + 1\right) 2^{\gamma l}\right) \\ &= O\left(\epsilon^{-1}(1 - 2^{-(\hat{\beta}-\gamma)/2})^{-1} \sum_{l=0}^L 2^{-(\hat{\beta}-\gamma)l/2} + \sum_{l=0}^L 2^{\gamma l}\right) \\ &= O\left(\epsilon^{-1}(1 - 2^{-(\hat{\beta}-\gamma)/2})^{-1} \frac{1 - 2^{-(\hat{\beta}-\gamma)(L+1)/2}}{1 - 2^{-(\hat{\beta}-\gamma)/2}} + \epsilon^{-\gamma/\alpha}\right) \\ &= O(\epsilon^{-1}), \end{aligned}$$

In the above, we used $\sum_{l=0}^L 2^{\gamma l} = O(2^{\gamma L}) = O(\epsilon^{-\gamma/\alpha})$ by our choice (3.10) and the fact that $\alpha \geq \gamma$.

(b). If $\hat{\beta} = \gamma$, then set $\epsilon_l = \epsilon/(2(L+1))$. This ensures that the error of the second term of (3.9) is bounded by $\epsilon/2$. Thus $N_l = 2(L+1)\epsilon^{-1}2^{-\hat{\beta}l} + 1$. The cost is bounded by

$$\begin{aligned} \sum_{l=0}^L N_l C_l &= O\left(\sum_{l=0}^L ((L+1)\epsilon^{-1}2^{-\hat{\beta}l} + 1)2^{\gamma l}\right) \\ &= O\left((L+1)^2\epsilon^{-1} + \sum_{l=0}^L 2^{\gamma l}\right) \\ &= O(\epsilon^{-1}(\log 1/\epsilon)^2). \end{aligned}$$

(c). If $\hat{\beta} < \gamma$, then we set $\epsilon_l = \frac{\epsilon}{2}2^{-(\gamma-\hat{\beta})L/2}(1-2^{-(\gamma-\hat{\beta})/2})2^{(\gamma-\hat{\beta})l/2}$. So the second error term of (3.9) is bounded by

$$\begin{aligned} \frac{\epsilon}{2}2^{-(\gamma-\hat{\beta})L/2}(1-2^{-(\gamma-\hat{\beta})/2})\sum_{l=0}^L 2^{(\gamma-\hat{\beta})l/2} &= \frac{\epsilon}{2}2^{-(\gamma-\hat{\beta})L/2}(1-2^{-(\gamma-\hat{\beta})/2})\frac{2^{(\gamma-\hat{\beta})(L+1)/2}-1}{2^{(\gamma-\hat{\beta})/2}-1} \\ &= \frac{\epsilon}{2}(1-2^{-(\gamma-\hat{\beta})/2})\frac{2^{(\gamma-\hat{\beta})/2}-2^{-(\gamma-\hat{\beta})L/2}}{2^{(\gamma-\hat{\beta})/2}-1} \\ &< \frac{\epsilon}{2}. \end{aligned}$$

Moreover, we have

$$N_l = 2\epsilon^{-1}2^{(\gamma-\hat{\beta})L/2}(1-2^{-(\gamma-\hat{\beta})/2})^{-1}2^{-(\hat{\beta}+\gamma)l/2} + 1.$$

The cost is bounded by

$$\begin{aligned} \sum_{l=0}^L N_l C_l &= O\left(\sum_{l=0}^L (\epsilon^{-1}2^{(\gamma-\hat{\beta})L/2}(1-2^{-(\gamma-\hat{\beta})/2})^{-1}2^{-(\hat{\beta}+\gamma)l/2} + 1)2^{\gamma l}\right) \\ &= O\left(\epsilon^{-1}2^{(\gamma-\hat{\beta})L/2}(1-2^{-(\gamma-\hat{\beta})/2})^{-1}\sum_{l=0}^L 2^{(\gamma-\hat{\beta})l/2} + \sum_{l=0}^L 2^{\gamma l}\right) \\ &= O\left(\epsilon^{-1}2^{(\gamma-\hat{\beta})L/2}(1-2^{-(\gamma-\hat{\beta})/2})^{-1}\frac{2^{(\gamma-\hat{\beta})(L+1)/2}-1}{2^{(\gamma-\hat{\beta})/2}-1} + \epsilon^{-\gamma/\alpha}\right) \\ &= O(\epsilon^{-1-(\gamma-\hat{\beta})/\alpha}). \end{aligned}$$

In the above, we used equation (3.10) and

$$2^{(\gamma-\hat{\beta})L} < 2^{(\gamma-\hat{\beta})(\frac{\log(2\epsilon^{-1})}{\alpha}+1)} = 2^{(\gamma-\hat{\beta})}2^{(\gamma-\hat{\beta})/\alpha}\epsilon^{-(\gamma-\hat{\beta})/\alpha}.$$

Since $L \approx \alpha^{-1}\log(2/\epsilon)$ and $\delta = 1/(100(L+1))$, we have $\log(1/\delta) = \log \log(2/\epsilon) + \log(100/\alpha) = O(\log \log 1/\epsilon)$. Each estimation of the cost should be multiplied by $O((\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2)$. $\square$

4 Quantum-accelerated MLMC for solving SDEs

Let us discuss how to apply MLMC to solve Problem 1 with stochastic differential equation (2.1).

4.1 Preliminary

Throughout the paper we make the following assumptions on the coefficients of the SDE and the payoff function.

Assumption 1. *We assume μ and σ are globally Lipschitz continuous, i.e., there exists a constant L such that*

$$|\mu(t, x) - \mu(s, y)| \leq L(|t - s| + |x - y|), \quad |\sigma(t, x) - \sigma(s, y)| \leq L(|t - s| + |x - y|) \quad (4.1)$$

hold for all $s, t \in [0, T], x, y \in \mathbb{R}$. We further assume the initial value X_0 satisfies $\mathbb{E}[X_0^m] \leq C_m$ for constants $C_m \geq 0$.

We remark that [Assumption 1](#) implies at most linear growth of μ and σ , and there exists a unique strong solution of SDE [\(2.1\)](#) [\[42\]](#).

We say a numerical approximation $\hat{X}_k$ with time step size $h = T/n$ is of strong order r , if for any $m \geq 1$, there exists a constant C_m such that

$$\mathbb{E} \left(\sup_{0 \leq kh \leq T} |\hat{X}_k - X_{kh}|^m \right) \leq C_m h^{rm}. \quad (4.2)$$

One class of general high order schemes is the Taylor-Itô scheme of the general form [\[42\]](#)

$$\hat{X}_{k+1} = \sum_{\alpha \in \mathcal{A}_m} f_\alpha(kh, \hat{X}_k) I_\alpha \quad (4.3)$$

where f_α 's are the coefficient functions (depending on μ and σ) and I_α are multiple Itô integrals over the time interval $[kh, (k+1)h]$. For instance, we may consider the Euler-Maruyama scheme (of strong order $1/2$)

$$\hat{X}_{k+1} = \hat{X}_k + \mu(\hat{X}_k, t)h + \sigma(\hat{X}_k, t)\Delta W_k, \quad (4.4)$$

for $k \in [n]_0$, or the Milstein scheme (of strong order 1)

$$\hat{X}_{k+1} = \hat{X}_k + \mu(\hat{X}_k, t)h + \sigma(\hat{X}_k, t)\Delta W_k + \frac{1}{2}\sigma(\hat{X}_k, t)\partial_X \sigma(\hat{X}_k, t)((\Delta W_k)^2 - h), \quad (4.5)$$

for $k \in [n]_0$, where ΔW_k are i.i.d. normal random variables with expected value zero and variance h . We remark that there exists another kind of general high order schemes called Taylor-Stratonovich schemes [\[42\]](#), which is easier to implement. We will discuss it in [Appendix B](#).

Assumption 2. *The coefficient functions f_α are globally Lipschitz continuous with respect to x .*

There exist a Taylor-Itô scheme and a Taylor-Stratonovich scheme satisfying [Assumption 2](#), which can achieve strong order $r = k/2$ for all $k \geq 1$. We refer to [\[42, Section 10\]](#) for more details.

We are also given an assumption for the final payoffs:

Assumption 3. *We assume the payoff function*

$$\mathcal{P} = \mathcal{P}(X_T) \quad (4.6)$$

is piecewise Lipschitz continuous, i.e., there exist constants $-\infty = l_0 < l_1 < \dots < l_q < l_{q+1} = +\infty$ and $L > 0$, such that

$$|\mathcal{P}(x) - \mathcal{P}(y)| \leq L|x - y|, \quad \forall x, y \in (l_j, l_{j+1}). \quad (4.7)$$

In the reminder of this paper, we consider the SDEs, stochastic schemes, and payoffs satisfying [Assumption 1](#), [Assumption 2](#), [Assumption 3](#), respectively. We remark that our results also hold true for high-dimensional systems of SDEs, given that the payoff function is “piecewise Lipschitz continuous” in some sense (*e.g.* all the discontinuous points are jump discontinuous points and form several separable hyperplanes). For technical simplicity, we will only focus on the analysis of SDE in one dimension in this section.

4.2 Method and theory

To solve a SDE problem, we apply the standard multilevel Monte Carlo method and regard the Taylor-Itô scheme as the discretization subroutine [\[28, 29\]](#).

At the high level, we estimate the discretized path $\hat{X}_k$ ($k \in [n]_0$) for different numbers of iterations n , and perform the quantum oracle

$$U_P(|x\rangle|0\rangle) = |x\rangle|\mathcal{P}(x)\rangle \quad (4.8)$$

to evaluate $\mathcal{P}(x)$ for any x . Setting $n_l = 2^l$ for $l = 1, \dots, L$, we apply the quantum-accelerated multilevel Monte Carlo

$$\mathbb{E}[\mathcal{P}(\hat{X}_{n_L})] = \mathbb{E}[\mathcal{P}(\hat{X}_{n_1})] + \sum_{l=1}^L [\mathbb{E}[\mathcal{P}(\hat{X}_{n_l}) - \mathcal{P}(\hat{X}_{n_{l-1}})]] \quad (4.9)$$

to estimate $\mathbb{E}[\mathcal{P}(X_T)]$. At the lower level, we divide $[0, T]$ by a uniform partition $0 = t_0 < t_1 < \dots < t_{n_l} = T$ with $h = T/n_l = T/2^l$ on the l -level discretization of [\(2.1\)](#), and perform stochastic numerical schemes to approximate X_T by $\hat{X}_{n_l}$.

To estimate the complexity of QA-MLMC, we need to figure out the parameters α, β, γ in [Theorem 2](#).

Proposition 2. *Under [Assumption 1](#), [Assumption 2](#) and [Assumption 3](#), for QA-MLMC with a numerical scheme of strong order r , we have $\alpha = r - o(1)$, $\beta = r - o(1)$, and $\gamma = 1$. Here $o(1)$ refers to an arbitrarily small real positive number. Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous, then the estimates on the parameters can be improved to $\alpha = r, \beta = 2r, \gamma = 1$.*

Proof. The estimate on γ comes from the construction of the algorithm. At the level l , we use a time step size $T/2^l$ to discretize the path, and compute the expectation at the final time. The dominant computational cost comes from simulating the path, which requires 2^l time steps. At the level $l+1$, we halve the time step size, and the number of the time steps is doubled. Since we are using the same numerical scheme at each level, the computational cost of propagating a single step remains the same, thus the total computational cost at the level $l+1$ is doubled. This indicates that $\gamma = 1$. We now focus on the estimate of α and β . We first consider the general payoff function satisfying [Assumption 3](#).

The proof is inspired by [\[30\]](#). For a sample of X_T and a sample of numerical approximation $\hat{X}_n$ with time step size h , we define a linear path $\Lambda(\lambda) = \lambda\hat{X}_n + (1-\lambda)X_T$ for $0 \leq \lambda \leq 1$. Let $M(\hat{X}_n, X_T) \in \{0, 1, \dots, q\}$ be the number of the discontinuity points along the path. Note that $\mathcal{P}$ is at least piecewise Lipschitz continuous, all the discontinuity points are of jump discontinuity. We hereby define the maximum size of the jump to be J , *i.e.*

$$J = \max_{1 \leq j \leq q} \left| \lim_{x \rightarrow l_j+} \mathcal{P}(x) - \lim_{x \rightarrow l_j-} \mathcal{P}(x) \right|.$$

Then

$$\mathbb{E}|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)|$$

$$\begin{aligned}
&= \mathbb{E} \left[\mathbb{E}[|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)| \mid M(\hat{X}_n, X_T)] \right] \\
&= \mathbb{E}[|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)| \mid M(\hat{X}_n, X_T) = 0] \times \mathbb{P}(M(\hat{X}_n, X_T) = 0) \\
&\quad + \sum_{j=1}^q \mathbb{E}[|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)| \mid M(\hat{X}_n, X_T) = j] \times \mathbb{P}(M(\hat{X}_n, X_T) = j) \\
&\leq L \mathbb{E}[|\hat{X}_n - X_T| \mid M(\hat{X}_n, X_T) = 0] \times \mathbb{P}(M(\hat{X}_n, X_T) = 0) \\
&\quad + \sum_{j=1}^q \left[qJ + L \mathbb{E}[|\hat{X}_n - X_T| \mid M(\hat{X}_n, X_T) = j] \right] \times \mathbb{P}(M(\hat{X}_n, X_T) = j) \\
&= L \mathbb{E}|\hat{X}_n - X_T| + qJ \mathbb{P}(M(\hat{X}_n, X_T) \geq 1).
\end{aligned}$$

The first part is bounded by $O(h^r)$. The second part can be bounded as, for a large integer m ,

$$\begin{aligned}
&\mathbb{P}(M(\hat{X}_n, X_T) \geq 1) \\
&\leq \mathbb{P}(\min_{1 \leq j \leq q} |X_T - l_j| \leq h^\alpha) + \mathbb{P}(|\hat{X}_n - X_T| \geq h^\alpha) \\
&\leq O(h^\alpha) + \frac{\mathbb{E}|\hat{X}_n - X_T|^m}{h^{\alpha m}} \\
&\leq O(h^\alpha) + O(h^{m(r-\alpha)}).
\end{aligned}$$

In the second inequality, the first term $O(h^\alpha)$ follows from that X_T is a continuously-distributed random variable with a bounded density due to the Picard iteration used to establish existence and uniqueness [50] under the global Lipschitz continuous assumption. The second term follows from the Markov inequality. Therefore we have

$$\mathbb{E}|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)| \leq O(h^r) + O(h^\alpha) + O(h^{m(r-\alpha)}) \quad (4.10)$$

holds for all m , which implies $\alpha = rm/(m+1) = r - o(1)$.

The estimate of β is similar to that of α . Note that $V_l = \mathbb{V}[\mathcal{P}_l - \mathcal{P}_{l-1}]$, so $V_l \leq (\sqrt{\mathbb{V}[\mathcal{P} - \mathcal{P}_l]} + \sqrt{\mathbb{V}[\mathcal{P} - \mathcal{P}_{l-1}]})^2$. It suffices to bound $\mathbb{E}[|\mathcal{P} - \mathcal{P}_l|^2]$, which is larger than $\mathbb{V}[\mathcal{P} - \mathcal{P}_l]$. Using the same technique of estimating α ,

$$\begin{aligned}
&\mathbb{E}|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)|^2 \\
&= \mathbb{E}[|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)|^2 \mid M(\hat{X}_n, X_T) = 0] \times \mathbb{P}(M(\hat{X}_n, X_T) = 0) \\
&\quad + \sum_{j=1}^q \mathbb{E}[|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)|^2 \mid M(\hat{X}_n, X_T) = j] \times \mathbb{P}(M(\hat{X}_n, X_T) = j) \\
&\leq L^2 \mathbb{E}[|\hat{X}_n - X_T|^2 \mid M(\hat{X}_n, X_T) = 0] \times \mathbb{P}(M(\hat{X}_n, X_T) = 0) \\
&\quad + \sum_{j=1}^q \left[2q^2 J^2 + 2L^2 \mathbb{E}[|\hat{X}_n - X_T|^2 \mid M(\hat{X}_n, X_T) = j] \right] \times \mathbb{P}(M(\hat{X}_n, X_T) = j) \\
&\leq 2L^2 \mathbb{E}|\hat{X}_n - X_T|^2 + 2q^2 J^2 \mathbb{P}(M(\hat{X}_n, X_T) \geq 1).
\end{aligned}$$

The first part is bounded by $O(h^{2r})$. The second part is bounded by $O(h^\beta) + O(h^{m(r-\beta)})$ for an arbitrarily large integer m , for the same reason in estimating α . It follows that

$$\mathbb{E}|\mathcal{P}(\hat{X}_n) - \mathcal{P}(X_T)|^2 \leq O(h^{2r}) + O(h^\beta) + O(h^{m(r-\beta)}), \quad (4.11)$$

which implies $\beta = rm/(m+1) = r - o(1)$.

Finally, if further the payoff function $\mathcal{P}$ is globally Lipschitz everywhere, then we have $\mathbb{P}(M(\hat{X}_n, X_T) \geq 1) = 0$. It is straightforward to conclude from the previous analysis that $\alpha = r, \beta = 2r$. $\square$

We remark that the estimates of α and β are possibly not sharp for some of the Taylor-Itô schemes. For example, if the payoff function $\mathcal{P}$ is a linear function, then α will be exactly the weak convergence order, which is, for many numerical schemes, larger than the strong convergence order. Nevertheless, our [Proposition 2](#) holds true for more general payoff functions and general high order schemes, and it suffices for QA-MLMC to achieve speedup over classical algorithms. In [Appendix B](#), we perform careful numerical tests of the values of α and β under different smoothness assumptions. We observe that our estimate for β is sharp for both Lipschitz continuous payoff function and discontinuous payoff function, and our estimate for α is sharp for discontinuous payoff functions, while larger α is observed for smoother payoff functions.

[Proposition 2](#) is sufficient to determine the complexity of both classical and quantum-accelerated MLMC for solving SDEs. We start with the classical case. A discussion about certain numerical schemes for typical payoffs has been proposed in Section 5 of [\[29\]](#). For high-order schemes and general payoffs, we state the result as follows.

Proposition 3. *Consider [Problem 1](#) for the stochastic differential equation [\(2.1\)](#) under [Assumption 1](#), [Assumption 2](#) and [Assumption 3](#). Then MLMC with a numerical scheme for SDE of strong order r estimates $\mathbb{E}[\mathcal{P}]$ up to additive error ϵ with probability at least 0.99 in cost*

$$\begin{cases} O(\epsilon^{-2}), & r > 1, \\ O(\epsilon^{-1-1/r-o(1)}), & r \leq 1. \end{cases} \quad (4.12)$$

Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous everywhere, then the cost can be improved to

$$\begin{cases} O(\epsilon^{-2}), & r > 1/2, \\ O(\epsilon^{-2}(\log \epsilon)^2), & r = 1/2, \\ O(\epsilon^{-1/r}), & r < 1/2. \end{cases} \quad (4.13)$$

Proof. This is a straightforward result from [Lemma 3](#) and [Proposition 2](#). $\square$

[Proposition 3](#) tells that, for general SDE [\(2.1\)](#) and payoff function satisfying [Assumption 1](#), [Assumption 2](#), and [Assumption 3](#), it suffices for MLMC to use a numerical scheme of strong order $r > 1$ to obtain the complexity $O(\epsilon^{-2})$, and using a numerical scheme of strong order 1, *e.g.*, Milstein scheme, will lead to the complexity $O(\epsilon^{-2-o(1)})$, *i.e.* almost quadratic dependence on $1/\epsilon$. We note that if the payoff function is globally Lipschitz continuous everywhere, then it suffices to use a numerical scheme of strong order $1/2$, *e.g.*, Euler-Maruyama scheme, to achieve the complexity $\tilde{O}(\epsilon^{-2})$.

As a counterpart, we are ready to state our main theorem regarding the complexity of QA-MLMC for solving SDE.

Theorem 3. *Consider [Problem 1](#) for the stochastic differential equation [\(2.1\)](#) under [Assumption 1](#), [Assumption 2](#) and [Assumption 3](#). Then QA-MLMC with a numerical scheme for SDE of strong order r estimates $\mathbb{E}[\mathcal{P}]$ up to additive error ϵ with probability at least 0.99 in cost*

$$\begin{cases} O(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2), & r > 2, \\ O(\epsilon^{-1/2-1/r-o(1)}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2), & r \leq 2. \end{cases} \quad (4.14)$$

Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous everywhere, then the cost can be improved to

$$\begin{cases} O\left(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2\right), & r > 1, \\ O\left(\epsilon^{-1}(\log 1/\epsilon)^{7/2}(\log \log 1/\epsilon)^2\right), & r = 1, \\ O\left(\epsilon^{-1/r}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2\right), & r < 1. \end{cases} \quad (4.15)$$

Proof. This is a straightforward result from [Theorem 2](#) and [Proposition 2](#). $\square$

[Theorem 3](#) tells that, for general SDE (2.1) and payoff function satisfying [Assumption 1](#), [Assumption 2](#), and [Assumption 3](#), it suffices for QA-MLMC to use a numerical scheme of strong order $r > 2$ to obtain the complexity $\tilde{O}(\epsilon^{-1})$, and using a numerical scheme of strong order 2 will lead to the complexity $\tilde{O}(\epsilon^{-1-o(1)})$, *i.e.* almost linear dependence on $1/\epsilon$. We note that if the payoff function is globally Lipschitz continuous everywhere, then it suffices to use a numerical scheme of strong order 1, *e.g.*, Milstein scheme, to achieve the complexity $\tilde{O}(\epsilon^{-1})$.

Compared with the classical MLMC, the convergence order required to achieve possibly optimal complexity is higher. This is due to the tighter requirement on the parameters α, β and γ . Nevertheless, once the optimal complexity is reached, QA-MLMC achieves a quadratic speedup over the classical MLMC in terms of ϵ .

4.3 Generalization to the entire path dependence case

So far we have confined ourselves to the case that the payoff function $\mathcal{P}$ only depends on the final value X_T of the stochastic process. However, the payoff functions of some options of widely practical interest, including Asian call and put options, are in general dependent on the stochastic integral along the entire trajectory of X_t for any $0 \leq t \leq T$. Furthermore, as being discussed later in [Section 7](#), one of the most efficient ways to estimate the sensitivity of the portfolio involves calculating the expectation of a stochastic integral. Fortunately, QA-MLMC can be straightforwardly generalized to this situation, and we will elaborate it in this subsection.

We first allow the payoff function $\mathcal{P}$ to be possibly dependent on the stochastic integral of the entire trajectory through the following assumption:

Assumption 4. *We assume the payoff function has the form*

$$\mathcal{P} = \mathcal{P}\left(X_T, \int_0^T f(X_t)dt, \int_0^T g(X_t)dW_t\right), \quad (4.16)$$

where $\mathcal{P}(X, y, z)$ is piecewise Lipschitz continuous with respect to X, y, z , and f and g are two globally Lipschitz continuous functions.

Such a payoff function can be transformed to only depend on the final value of another stochastic process by extending the system to higher dimension. Precisely, let us define

$$Y_t = \int_0^t f(X_s)ds, \quad Z_t = \int_0^t g(X_s)dW_s,$$

then Y_t and Z_t , together with X_t , satisfy the system of stochastic differential equations

$$\begin{aligned} dX_t &= \mu(X_t, t)dt + \sigma(X_t, t)dW_t, \\ dY_t &= f(X_t)dt, \\ dZ_t &= g(X_t)dW_t. \end{aligned} \quad (4.17)$$

This can be formally written as a system of stochastic differential equations in higher dimension, *i.e.*

$$d\tilde{X}_t = \tilde{\mu}(\tilde{X}_t, t)dt + \tilde{\sigma}(\tilde{X}_t, t)d\tilde{W}_t,$$

where $\tilde{X}_t^\top = (X_t^\top, Y_t^\top, Z_t^\top)$, $\tilde{W}_t$ a Brownian motion in the dimension of $\tilde{X}_t$,

$$\tilde{\mu} = \begin{pmatrix} \mu(X_t, t) \\ f(X_t) \\ 0 \end{pmatrix}, \quad \tilde{\sigma} = \begin{pmatrix} \sigma(X_t, t) & 0 & 0 \\ 0 & 0 & 0 \\ g(X_t) & 0 & 0 \end{pmatrix}. \quad (4.18)$$

Define

$$\tilde{\mathcal{P}}(\tilde{X}_t) = \mathcal{P}(X_t, Y_t, Z_t),$$

then

$$\tilde{\mathcal{P}}(\tilde{X}_T) = \mathcal{P}(X_T, Y_T, Z_T) = \mathcal{P}\left(X_T, \int_0^T f(X_t)dt, \int_0^T g(X_t)dW_t\right),$$

and thus estimating the expectation of $\mathcal{P}$ is equivalent to solving [Problem 1](#) in terms of $\tilde{X}_t$ and $\tilde{\mathcal{P}}$. By [Assumption 4](#), we can check that $\tilde{X}_t$ and $\tilde{\mathcal{P}}$ satisfy the [Assumption 1](#), [Assumption 2](#) and [Assumption 3](#). Therefore, according to [Proposition 3](#) and [Theorem 3](#), we have the following result.

Corollary 1. *Consider the payoff function satisfying [Assumption 4](#). Then, further under [Assumption 1](#) and [Assumption 2](#), MLMC with a numerical scheme for SDE of strong order r estimates $\mathbb{E}[\mathcal{P}]$ up to additive error ϵ with probability at least 0.99 in cost*

$$\begin{cases} O(\epsilon^{-2}), & r > 1, \\ O(\epsilon^{-1-1/r-o(1)}), & r \leq 1. \end{cases} \quad (4.19)$$

Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous everywhere, then the cost can be improved to

$$\begin{cases} O(\epsilon^{-2}), & r > 1/2, \\ O(\epsilon^{-2}(\log \epsilon)^2), & r = 1/2, \\ O(\epsilon^{-1/r}), & r < 1/2. \end{cases} \quad (4.20)$$

Corollary 2. *Consider the payoff function satisfying [Assumption 4](#). Then, further under [Assumption 1](#) and [Assumption 2](#), QA-MLMC with a numerical scheme for SDE of strong order r estimates $\mathbb{E}[\mathcal{P}]$ up to additive error ϵ with probability at least 0.99 in cost*

$$\begin{cases} O(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2), & r > 2, \\ O(\epsilon^{-1/2-1/r-o(1)}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2), & r \leq 2. \end{cases} \quad (4.21)$$

Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous everywhere, then the cost can be improved to

$$\begin{cases} O(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2), & r > 1, \\ O(\epsilon^{-1}(\log 1/\epsilon)^{7/2}(\log \log 1/\epsilon)^2), & r = 1, \\ O(\epsilon^{-1/r}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2), & r < 1. \end{cases} \quad (4.22)$$

We next discuss several applications arising in mathematical finance.

5 Black-Scholes option pricing model

We consider the Black-Scholes model for option pricing [6, 7, 37], which is at the core of quantitative finance, as the first application of [Problem 1](#).

5.1 Black-Scholes equation

The Black-Scholes model is used to price a variety of financial derivatives via a simple and analytical solvable model using a small number of input parameters. In Black-Scholes model, we are mainly interested in the following Geometric Brownian Motion

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad (5.1)$$

where S_t is the asset price, σ the velocity of the asset and μ is the risk-free interest rate. The rate of return on the asset μ is assumed to be a constant. There are some other assumptions on the model, which Fischer Black and Myron Scholes have thoroughly discussed in [6]. For path-independent option, the price of option $V(s, t)$ follows the Black-Scholes equation

$$\frac{\partial V}{\partial t} + \frac{\sigma^2 s^2}{2} \frac{\partial^2 V}{\partial s^2} + \mu s \frac{\partial V}{\partial s} = \mu V \quad (5.2)$$

for $s > 0$, $t \leq T$, with a terminal condition $V(s, T) = \psi(s)$, where ψ is the final payoff. The link between (5.2) and (5.1) is established by the Feynman-Kac formula. The solution of (5.2) can be represented as the expectation of the solution of (5.1).

Lemma 4. *Consider a PDE defined in (5.2) subject to a terminal condition $V(s, T) = \psi(s)$. The solution $V(s, t)$ can be written as a conditional expectation*

$$V(s, t) = \mathbb{E}[e^{-\mu(T-t)} \psi(S_T) \mid S_t = s], \quad (5.3)$$

where S_t is an Itô process driven by (5.1).

Using Itô lemma, (5.1) under the condition $S_t = s$ can be solved as

$$S_T = s e^{\sigma W_{T-t} + (\mu - \sigma^2/2)(T-t)}. \quad (5.4)$$

For European options, one can analytically solve for $V(s, t)$, which has a deterministic expression. However, in the case of more complex payoff functions, one needs to resort to the Monte Carlo method, which gives an approximation of $V(s, t)$ by averaging the payoff over samples W_t . The quantum version of the Monte Carlo method can be applied to reduce the complexity from $\tilde{O}(\epsilon^{-2})$ to $\tilde{O}(\epsilon^{-1})$, and the examples of Black-Scholes model for European options and Asian options are thoroughly discussed in [55].

Nevertheless, the analytical solution (5.4) can be used for benchmarking numerical simulations to understand the scaling behavior of discretization schemes. In [Appendix B](#), we present some numerical tests on the Black-Scholes equation to demonstrate our theory results. In our method, we apply discretization schemes to (5.1) to simulate random paths of S_t , such as Euler-Maruyama scheme

$$\hat{S}_{k+1} = \hat{S}_k + \mu \hat{S}_k h + \sigma \hat{S}_k \Delta W_k, \quad (5.5)$$

and Milstein scheme

$$\hat{S}_{k+1} = \hat{S}_k + \mu \hat{S}_k h + \sigma \hat{S}_k \Delta W_k + \frac{\sigma^2}{2} \hat{S}_k ((\Delta W_k)^2 - h). \quad (5.6)$$

Here, (5.5) and (5.6) are reduced from (4.4) and (4.5) respectively. Then we estimate $\mathbb{E}[P(S_T)]$ with $P(S_t) = e^{-\mu(T-t)}\psi(S_T)$, where the quantum oracle (4.8) can be reduced to

$$U_\psi(|x\rangle|0\rangle) = |x\rangle|\psi(x)\rangle. \quad (5.7)$$

It is worthwhile to mention that our method is suitable for general SDEs and payoff functions.

5.2 Option pricing

Black-Scholes model provides a useful tool for option pricing, which was a longstanding problem in finance. Briefly speaking, an option is a contract that allows the holder to buy or sell a financial asset at a fixed price in the future. A call option is an option to buy an asset and a put option is an option to sell it. In this subsection, we would like to present some well-known call options as examples. Put options can be developed in a similar way.

We first consider Lipschitz continuous options. One of the famous options is *European option*, in which the final payoff is given by

$$\psi(S_T) = (S_T - K)^+ := \max\{S_T - K, 0\}, \quad (5.8)$$

where $K > 0$ is the strike price of the option. That means that if $S_T \leq K$, the option is worthless, and if $S_T > K$, the holder can buy the asset for K dollars and sell it at market price, making a profit of $S_T - K$. Note that the European option is path independent and only relies on the terminal price S_T , without the consideration of the whole path.

There also exist path-dependent options relying on the path $\{S_t\}$. One example is *Asian option*, in which the final payoff is considered to be

$$\psi(S_T) = \left(\frac{1}{T} \int_0^T S_t dt - K\right)^+ \quad (5.9)$$

with the strike $K > 0$. The payoff of Asian option is determined by the average of the asset price over $[0, T]$.

It follows from a theorem that the classical multilevel Monte Carlo method with Euler-Maruyama scheme achieves the complexity $\tilde{O}(\epsilon^{-2})$ for globally Lipschitz continuous options, which improves the complexity of standard Monte Carlo method $\tilde{O}(\epsilon^{-3})$. [28, 29].

We then consider piecewise Lipschitz continuous options. A typical option is *Digital option*, in which the digital option (Cash-or-nothing option) gives the final payoff as the form

$$\psi(S_T) = \mathcal{H}(S_T - K), \quad (5.10)$$

with the strike $K > 0$, where $\mathcal{H}$ is the Heaviside function. Clearly it is a non-Lipschitz continuous option.

For piecewise Lipschitz continuous options, classical multilevel Monte Carlo method with Euler-Maruyama scheme achieves the complexity $\tilde{O}(\epsilon^{-2.5})$, and it can be further improved to $\tilde{O}(\epsilon^{-2})$ by Milstein scheme and extreme paths [29].

By Theorem 3 and Corollary 2, the quantum-accelerated multilevel Monte Carlo method can be used to obtain the following result:

Corollary 3. *We consider Problem 1 given by a Geometric Brownian Motion (5.1) with an initial condition $S_t = s_0$. We are given the zeroth-order quantum oracle (5.7) for a final piecewise Lipschitz continuous payoff $\mathcal{P}$ as defined in (4.6) or (4.16). Then QA-MLMC*

with a numerical scheme for SDE of strong order r estimates $\mathbb{E}[P]$ up to additive error ϵ with probability at least 0.99 in cost

$$\begin{cases} \tilde{O}(\epsilon^{-1}), & r > 2, \\ \tilde{O}(\epsilon^{-1/2-1/r-o(1)}), & r \leq 2. \end{cases} \quad (5.11)$$

Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous everywhere, then the cost can be improved to

$$\begin{cases} \tilde{O}(\epsilon^{-1}), & r \geq 1, \\ \tilde{O}(\epsilon^{-1/r}), & r < 1. \end{cases} \quad (5.12)$$

For globally Lipschitz continuous P , it suffices to perform the Milstein scheme (5.6) with strong order $r = 1$ to achieve the complexity $\tilde{O}(\epsilon^{-1})$. While for piecewise Lipschitz continuous P , it requires to use a numerical scheme of strong order $5/2$ to obtain the complexity $\tilde{O}(\epsilon^{-1})$.

6 Local Volatility model

The Local Volatility model generalizes the Black-Scholes model (5.1) by treating volatility σ as a function of the asset S_t and the time t [20, 22]. The Local Volatility model is a kind of simplification of the stochastic volatility model, which assumes σ has a randomness of its own. Differing from the analytically solvable Black-Scholes model, we are required to simulate the SDE, as there is a lack of explicit solutions to estimate the price of the Local Volatility model.

The Local Volatility model characterizes S_t by a generalized Geometric Brownian Motion as the form

$$dS_t = \mu S_t dt + \sigma(S_t, t) S_t dW_t \quad (6.1)$$

for $t \leq T$, with an initial condition $S_t = s_0$. Here μ is the instantaneous risk-free interest rate, $\sigma(S_t, t)$ is the instantaneous volatility of the risky asset, which is sufficiently smooth with respect to S_t and t , and W_t is a standard Brownian motion. We further require (6.1) satisfies the assumptions in Section 4.

By Feynman-Kac Formula, the SDE (6.1) corresponds to the Black-Scholes equation that describes the price of the option $V = V(s, t)$ by

$$\frac{\partial V}{\partial t} + \frac{\sigma^2(s, t) s^2}{2} \frac{\partial^2 V}{\partial s^2} + \mu s \frac{\partial V}{\partial s} = \mu V \quad (6.2)$$

for $s > 0$, $t \leq T$, with a terminal condition $V(s, T) = \psi(s)$, where ψ is the final payoff.

We establish the link between (6.1) and (6.2) by Feynman-Kac Formula.

Lemma 5. Consider a PDE defined in (6.2) subject to a terminal condition $V(s, T) = \psi(s)$. The solution $V(s, t)$ can be written as a conditional expectation

$$V(s, t) = \mathbb{E}[e^{-\mu(T-t)} \psi(S_T) \mid S_t = s], \quad (6.3)$$

where S_t is an Itô process driven by (6.1).

Thus, we can estimate $\mathbb{E}[P(S_T)]$ with $P(S_t) = e^{-\mu(T-t)} \psi(S_T)$ to obtain $V(s, t)$ for general options.

We consider the same setting for payoffs in Section 5. By Theorem 3 and Corollary 2, the quantum-accelerated multilevel Monte Carlo method can be used to obtain the following results:

Corollary 4. We consider [Problem 1](#) given by a SDE (6.1) with an initial condition $S_t = s_0$. We are given the zeroth-order quantum oracle (5.7) for a final piecewise Lipschitz continuous payoff $\mathcal{P}$ as defined in (4.6) or (4.16). Then QA-MLMC with a numerical scheme for SDE of strong order r estimates $\mathbb{E}[P]$ up to additive error ϵ with probability at least 0.99 in cost

$$\begin{cases} \tilde{O}(\epsilon^{-1}), & r > 2, \\ \tilde{O}(\epsilon^{-1/2-1/r-o(1)}), & r \leq 2. \end{cases} \quad (6.4)$$

Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous everywhere, then the cost can be improved to

$$\begin{cases} \tilde{O}(\epsilon^{-1}), & r \geq 1, \\ \tilde{O}(\epsilon^{-1/r}), & r < 1. \end{cases} \quad (6.5)$$

7 Sensitivity of price

Besides the price, another important factor of financial products is the risk. If two financial products have similar expected return, the one with lower risk is usually preferable because people would like to ensure a positive rate of return earnings under most situations and are not willing to take a risk. One common way to reduce the risk is hedging, that is, forming a portfolio by buying several financial products with different reaction and sensitivity to the change of the market. For example, if there are two stocks, and one is positively correlated with the global economy and the other one is negatively correlated, then one may purchase both of them with a specific ratio to guarantee the return no matter whether the global economy flourishes or declines. Correctly identifying and measuring the sensitivity is crucial to hedging portfolios.

“The Greeks” label the sensitivity of the price of the option [37]. They are partial derivatives with respect to the parameters such as the initial price, the starting time, the risk-free rate and the volatility. Computing these is essential to hedge portfolios, and therefore it is even more important than pricing the option itself. Different from the Black-Scholes model, there’s no closed form for Greeks with general payoff functions and coefficients in SDEs. Let $u(s, t)$ denote $\mathbb{E}(\mathcal{P}(X_T))$ where X_T is the solution of SDE

$$dX_\tau = \mu(X_\tau)d\tau + \sigma(X_\tau)dW_\tau, \quad \tau \in [t, T] \quad (7.1)$$

with starting time t and initial condition $X_t = s$. Different types of Greeks are as follow:

- Delta:

$$\frac{\partial u(s, t)}{\partial s}, \quad (7.2)$$

- Gamma:

$$\frac{\partial^2 u(s, t)}{\partial s^2}, \quad (7.3)$$

- Vega:

$$\frac{\partial u(s, t)}{\partial \sigma}, \quad (7.4)$$

- Theta:

$$\frac{\partial u(s, t)}{\partial t}, \quad (7.5)$$

- Rho:

$$\frac{\partial u(s, t)}{\partial r}. \quad (7.6)$$

Numerical treatments of Greeks have been widely studied in *e.g.* [10, 27, 47, 49, 54]. A natural way to compute the Greeks is finite difference, that is, to compute the expectation of the payoff function with different parameters and use finite difference to approximate the derivatives. Although finite difference approximation of derivatives suffers from numerical instability, using common stochastic paths for both estimators in finite difference can reduce the classical complexity. However, if the payoff function is not smooth enough, finite difference might not be effective. Here we consider an alternative approach using Malliavin calculus [27] to compute the Greeks. Although rigorous derivation of using Malliavin calculus to compute Greeks is technically involved, the basic idea is simply using an analog of calculus of variations and integration by parts formula in the stochastic calculus setting, and the outcome formula to compute Greeks is amazingly concise thus easy to implement. We will show how to combine it with QA-MLMC to achieve quadratic speedup.

For expository purpose let us consider the example of computing Delta in one dimension and omit a few technical details. We refer interested readers to [27] for elaborations and other types of the Greeks. Assume the process X_t is given by (2.1), and for simplicity we assume that the system is autonomous, *i.e.* there is no explicit time dependence in μ and σ . Define another stochastic process Y_t to be

$$dY_t = \mu'(X_t)Y_t dt + \sigma'(X_t)Y_t dW_t, \quad Y_0 = 1. \quad (7.7)$$

Under [Assumption 1](#), [Assumption 2](#), [Assumption 3](#), and several further technical assumptions that μ, σ are C^1 functions, σ is uniformly bounded away from 0 and the payoff function $\mathcal{P}$ has uniformly bounded second moment, [27, Proposition 3.2] tells that Delta can be represented via the following formula:

$$\frac{\partial u(s, 0)}{\partial s} = \mathbb{E} \left[\frac{1}{T} \mathcal{P}(X_T) \int_0^T Y_t \sigma^{-1}(X_t) dW_t \mid X_0 = s, Y_0 = 1 \right]. \quad (7.8)$$

Notice that here we only assume the payoff function $\mathcal{P}$ to be piecewise Lipschitz continuous. Therefore, we can compute Delta by sampling (X_t, Y_t) according to (2.1) and (7.7), and then use QA-MLMC to compute the expectation (7.8), which is in the form of [Assumption 4](#). By [Corollary 2](#), we have the following complexity estimate:

Corollary 5. *Assume we are given the zeroth-order quantum oracle (5.7) for a final piecewise Lipschitz continuous payoff $\mathcal{P}$ as defined in (4.16). Then by sampling (X_t, Y_t) according to (2.1) and (7.7) with a numerical scheme of strong order r and estimating Delta via (7.8) using QA-MLMC, the Delta can be approximated up to an additive error ϵ with probability at least 0.99 in cost*

$$\begin{cases} \tilde{O}(\epsilon^{-1}), & r > 2, \\ \tilde{O}(\epsilon^{-1/2-1/r-o(1)}), & r \leq 2. \end{cases} \quad (7.9)$$

Furthermore, if the payoff function $\mathcal{P}$ is globally Lipschitz continuous, then the cost can be improved to

$$\begin{cases} \tilde{O}(\epsilon^{-1}), & r \geq 1, \\ \tilde{O}(\epsilon^{-1/r}), & r < 1. \end{cases} \quad (7.10)$$

8 Binomial option pricing model

The binomial option pricing model, a.k.a. the binomial lattice model, provides a discrete time (lattice-based) approximation to the Black–Scholes model [19, 37, 56]. The bino-

mial model assumes that movements in the price follow a binomial distribution, which approaches the log-norm distribution of the Geometric Brownian Motion.

Given a risk-neutral measure and an initial condition S_0 , we divide $[0, T]$ by a uniform partition $0 = t_0 < t_1 < \dots < t_n = T$ with $h = T/n$, and model the price of a stock in discrete time by a Markov chain

$$S_{k+1} = S_k Y_{k+1}, \quad (8.1)$$

where $\{Y_k\} \ k \in [n]_0$ are i.i.d. with a two points “up” U and “down” D distribution

$$\begin{aligned} \Pr(Y_k = U) &= p; \\ \Pr(Y_k = D) &= 1 - p. \end{aligned} \quad (8.2)$$

Under this setting, the probability of the value S_N is given by

$$\Pr(S_n = U^k D^{n-k} S_0) = \binom{n}{k} p^k (1-p)^{n-k}. \quad (8.3)$$

Thus, we create a binomial tree with distribution $B(n, p)$ that describes the prices over time.

We require the conditional expectation of (8.1) matches the Geometric Brownian Motion (5.1), giving

$$pS_k U + (1-p)S_k D = \mathbb{E}[S_{k+1} | S_k] = S_k e^{rh}, \quad (8.4)$$

which is

$$p = \frac{e^{rh} - D}{U - D}. \quad (8.5)$$

Practially, we could use $\mathbb{E}[S_{k+1} | S_k] \approx S_k(1 + rh)$ instead, which in fact corresponds with the expectation of the Euler-Maruyama scheme (5.5). Similarly, we require the conditional variance of (8.1) matches the geometry Brownian motion (5.1),

$$pS_k^2 U^2 + (1-p)S_k^2 D^2 - [pS_k^2 U + (1-p)S_k^2 D]^2 = \text{Var}(S_{k+1} | S_k) = S_k^2 e^{2rh} (e^{\sigma^2 h} - 1) \approx S_k^2 \sigma^2 h. \quad (8.6)$$

There are various binomial lattice models satisfying (8.4) and (8.6). The first and the most famous model is the CRR model [19]. If we choose step size $h \leq \sigma^2/r^2$, and assume $D = 1/U$, we obtain

$$U = e^{\sigma\sqrt{h}}, \quad D = e^{-\sigma\sqrt{h}}. \quad (8.7)$$

Alternatively, we can also consider an equal probabilities model, the JR model [38]. If we choose $p = 1/2$ to determine U and D , we would obtain

$$U = e^{(r-\sigma^2/2)h+\sigma\sqrt{h}}, \quad D = e^{(r-\sigma^2/2)h-\sigma\sqrt{h}}. \quad (8.8)$$

Similar as before, we can use first-order approximations of exponential factors as values of U and D in practice.

Now we perform BOPM to estimate expectation of option pricing $\mathbb{E}[P(S_T)]$. The error of approximating general Black-Scholes option prices by CCR model (8.7) is $O(1/\sqrt{n})$ [36], while it can be further improved to $O(1/n)$ for European option [44].

In general, BOPM works for various options with complexity $O(n)$, since we need to calculate $S_n = U^k D^{n-k} S_0$. Even if $U = 1/D$, we should still multiply U in total $2k - n$ times. In general, we can apply multilevel MC to reduce the cost of BOPM. Since we are required to choose $n = O(1/\epsilon^2)$, its complexity is generally no better than Euler-Maruyama scheme.

But for the piecewise constant payoff, such as the digital option (5.10), it allows us to develop classical and quantum algorithms that only require $O(\log N)$ time complexity. Given $\bar{S}_0 < \dots < \bar{S}_m$, we define a piecewise constant payoff by

$$\psi(S_n) = \begin{cases} \psi_L, & S_n < \bar{S}_0, \\ \psi_j, & \bar{S}_{j-1} \leq S_n < \bar{S}_j, \\ \psi_R, & S_n \geq \bar{S}_m, \end{cases} \quad (8.9)$$

where ψ_j, ψ_L, ψ_R are constant. Noting that (8.9) is a special case of Assumption 3. We are also given a zeroth-order classical oracle

$$O_\psi(S_0, k) = \psi(U^k D^{n-k} S_0), \quad (8.10)$$

which outputs the final payoff (8.9). The procedure is described as follows: given S_0, U, D, N , we can determine the criteria k_j^* corresponding with $\bar{S}_j$, by

$$k_j^* \log U + (n - k_j^*) \log D + \log S_0 = \log \bar{S}_j. \quad (8.11)$$

When we input n , we just compare n with n_j^* , and determine the payoff (8.9). Under this setting, the complexity of (8.10) is independent on n , by avoiding n times multiplication for calculating $S_n = U^k D^{n-k} S_0$.

We follow Theorem 12 of [46], in which we replace the fast random walk by BOPM.

Proposition 4. *We consider Problem 1 given by a Geometric Brownian Motion (5.1) with an initial condition S_0 . We are given the zeroth-order classical oracle (8.10) for a final piecewise constant payoff as defined in (8.9), which has a bounded variance independent of h . There exists a classical algorithm that estimates $\mathbb{E}[P]$ up to additive error ϵ with probability at least 0.99 in cost*

$$\tilde{O}(\epsilon^{-2}). \quad (8.12)$$

Proof. We aim to perform $n = O(1/\epsilon^2)$ steps BOPM (8.1) to obtain S_T within error ϵ , and then calculate $P(S_T)$ as one sample of Monte Carlo simulation. Given U, D, S_0 sampled from π_0 in $O(1)$, and the oracle (8.10) for (8.9), we do the sampling procedure as follows.

Inspired by Lemma 11 and Theorem 12 of [46], we determine n by sampling a binomial distribution $B(n, p)$, and then compare to n_j^* to determine the payoff by (8.10). The sampling from a binomial distribution requires $O(\log n)$ expected samples and $O(\log n)$ expected time [9, 25]. Thus, the cost of each iteration is $O(\log n)$.

By Monte Carlo simulation, we repeat the above sampling procedure $O(\epsilon^{-2})$ times. Thus, we can estimate $\mathbb{E}[P]$ with the final complexity $O(\epsilon^{-2} \log n) = O(\epsilon^{-2} \log 1/\epsilon)$ in time. $\square$

Similarly, we are given a zeroth-order quantum oracle by modifying (5.7) to be

$$U_\psi(|S_0\rangle|k\rangle|0\rangle) = |S_0\rangle|k\rangle|\psi(U^k D^{n-k} S_0)\rangle. \quad (8.13)$$

where ψ is the final payoff (8.9). The same as the procedure of (8.10), the complexity of (8.13) is independent on N .

We follow Theorem 22 of [46], in which we replace the quantum walk by BOPM.

Theorem 4. *We consider Problem 1 given by a Geometric Brownian Motion (5.1) with an initial condition S_0 . We are given the zeroth-order quantum oracle (8.13) for a final piecewise constant payoff as defined in (8.9), which has a bounded variance independent of h . There exists a quantum algorithm that estimates $\mathbb{E}[P]$ up to additive error ϵ with probability at least 0.99 in cost*

$$\tilde{O}(\epsilon^{-1}). \quad (8.14)$$

Proof. Inspired by Theorem 22 of [46], we apply QA-MC to the random seed used as input to a procedure for sampling from the binomial distributions. According to Proposition 4, the cost of each iteration is $O(\log n)$ with $n = O(1/\epsilon^2)$, and amplitude estimation gives the final complexity $O(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log \log 1/\epsilon)^2)$ in time. $\square$

9 Discussion

We have presented quantum-accelerated multilevel Monte Carlo methods for stochastic processes. We apply our algorithm to several applications arising in mathematical finance, in which we classify different financial models corresponding with different payoffs in detail. We have shown a quadratically improved dependence on precision can be achieved by our algorithm.

This work raises several natural open problems. First, from the PDE perspective, we only deal with parabolic PDEs as an application of simulating SDEs. However, Poisson's equation and elliptic PDEs can be solved by classical multilevel Monte Carlo methods [17]. Can we apply our algorithm to Poisson's equation, elliptic PDEs, or more general PDEs?

Second, we consider several types of financial models that can be presented as a SDE model Problem 1. However, we only consider time-independent payoffs $\mathcal{P}(X_T)$ that only rely on X_T . For more general time-dependent payoffs $\mathcal{P}(X_t)$ where X_t is the stochastic path in time t , such as the Lookback option [29], can we achieve such a quadratic speed-up for this general model? Furthermore, there are some financial models that can be solved by variational inequalities, such as American option. Can we develop corresponding quantum algorithms for such a generalization?

Finally, we aim to output a classical value for estimating the mean of a payoff (Problem 1). Can we provide other meaningful characteristics of stochastic processes, or some processes beyond the SDE modelling by quantum computer? And can we find more practical quantum input-output models for potential applications in finance or other fields?

Acknowledgments

The authors thank Andrew M. Childs, Lin Lin, and Nick Whiteley for valuable discussions and comments. JPL did part of this work while visiting the Simons Institute for the Theory of Computing in Berkeley and gratefully acknowledge its hospitality. We thank the National Energy Research Scientific Computing (NERSC) center and the Berkeley Research Computing (BRC) program at the University of California, Berkeley for making computational resources available.

The authors acknowledge support from National Science Foundation grant CCF-1813814, the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams and Accelerated Research in Quantum Computing programs. This work was also partially supported by the Department of Energy under Grant No. DE-SC0017867 and the National Science Foundation under the Quantum Leap Challenge Institutes (D.A., J.W.). We acknowledge support from the QuantERA ERA-NET Cofund in Quantum Technologies implemented within the European Union's Horizon 2020 Programme (QuantAlgo project) and EPSRC grants EP/R043957/1 and EP/T001062/1. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 817581).

References

- [1] A. Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In *29th Symposium on Theoretical Aspects of Computer Science*, volume 14, pages 636–647. LIPIcs, 2012. doi:10.4230/LIPIcs.STACS.2012.636.
- [2] D. An and L. Lin. Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, 2019. arXiv:1909.05500.
- [3] K. E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, 2008.
- [4] D. W. Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301, 2014. doi:10.1088/1751-8113/47/10/105301.
- [5] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics*, 356(3):1057–1081, 2017. doi:10.1007/s00220-017-3002-y.
- [6] F. Black and M. Scholes. The pricing of options and corporate liabilities, 1973. doi:10.1086/260062.
- [7] Z. Bodie. *Investments*. Tata McGraw-Hill Education, 2009.
- [8] A. Bouland, W. van Dam, H. Joorati, I. Kerenidis, and A. Prakash. Prospects and challenges of quantum finance, 2020. arXiv:2011.06492.
- [9] K. Bringmann, F. Kuhn, K. Panagiotou, U. Peter, and H. Thomas. Internal DLA: Efficient simulation of a physical growth model. In *International Colloquium on Automata, Languages, and Programming*, pages 247–258. Springer, 2014. doi:10.1007/978-3-662-43948-7_21.
- [10] S. Burgos and M. B. Giles. Computing greeks using multilevel path simulation. In: *Plaskota L., Woźniakowski H. (eds) Monte Carlo and Quasi-Monte Carlo Methods 2010. Springer Proceedings in Mathematics & Statistics*, 23:281–296, 2012. doi:10.1007/978-3-642-27440-4_13.
- [11] K. Burrage, P. M. Burrage, and T. Tian. Numerical methods for strong solutions of stochastic differential equations: An overview. *Proc. R. Soc. A Math. Phys. Eng. Sci.*, 460:373–402, 2004. doi:10.1098/rspa.2003.1247.
- [12] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais. Quantum algorithm and circuit design solving the Poisson equation. *New Journal of Physics*, 15(1):013021, 2013. doi:10.1088/1367-2630/15/1/013021.
- [13] A. M. Childs, R. Kothari, and R. D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017. doi:10.1137/16M1087072.
- [14] A. M. Childs and J.-P. Liu. Quantum spectral methods for differential equations. *Communications in Mathematical Physics*, 375:1427–1457, 2020. doi:10.1007/s00220-020-03699-z.
- [15] A. M. Childs, J.-P. Liu, and A. Ostrander. High-precision quantum algorithms for partial differential equations, 2020. arXiv:2002.07868.
- [16] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. Preconditioned quantum linear system algorithm. *Physical Review Letters*, 110(25):250504, 2013. doi:10.1103/PhysRevLett.110.250504.
- [17] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo

- methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3, 2011. doi:10.1007/s00791-011-0160-x.
- [18] P. C. S. Costa, S. Jordan, and A. Ostrander. Quantum algorithm for simulating the wave equation. *Physical Review A*, 99(1):012323, 2019. doi:10.1103/PhysRevA.99.012323.
- [19] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of financial Economics*, 7(3):229–263, 1979. doi:10.1016/0304-405X(79)90015-1.
- [20] E. Derman and I. Kani. Riding on a smile. *Risk*, 7(2):32–39, 1994.
- [21] J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133, 2013. doi:10.1017/S0962492913000044.
- [22] B. Dupire. Pricing with a smile. *Risk*, 7(1):18–20, 1994.
- [23] A. Engel, G. Smith, and S. E. Parker. Quantum algorithm for the Vlasov equation. *Physical Review A*, 100(6):062315, 2019. doi:10.1103/PhysRevA.100.062315.
- [24] L. C. Evans. *Partial differential equations (2nd ed.)*. American Mathematical Society, 2010. doi:10.1090/gsm/019.
- [25] M. Farach-Colton and M.-T. Tsai. Exact sublinear binomial sampling. *Algorithmica*, 73(4):637–651, 2015. doi:10.1007/s00453-015-0077-8.
- [26] F. Fontanela, A. J. Jacquier, and M. Oumgari. A quantum algorithm for linear PDEs arising in finance, 2019. doi:10.2139/ssrn.3499134.
- [27] E. Fournié, J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi. Applications of malliavin calculus to Monte Carlo methods in finance. *Finance and Stochastics*, 3:391–412, 1999. doi:10.1007/s007800050068.
- [28] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008. doi:10.1287/opre.1070.0496.
- [29] M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015. doi:10.1017/S096249291500001X.
- [30] M. B. Giles, D. J. Higham, and X. Mao. Analysing multi-level Monte Carlo for options with non-globally lipschitz payoff. *Finance and Stochastics*, 13(3):403–413, 2009. doi:10.1007/s00780-009-0092-1.
- [31] M. B. Giles and B. J. Waterhouse. Multilevel quasi-Monte Carlo path simulation. *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics*, 8:165–181, 2009. doi:10.1515/9783110213140.165.
- [32] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. doi:10.1145/3313276.3316366.
- [33] J. Gonzalez-Conde, A. Rodríguez-Rozas, E. Solano, and M. Sanz. Pricing financial derivatives with exponential quantum speedup, 2021. arXiv:2101.04023.
- [34] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. doi:10.1103/PhysRevLett.103.150502.
- [35] S. Heinrich. Multilevel Monte Carlo methods. In *International Conference on Large-Scale Scientific Computing*, pages 58–67. Springer, 2001. doi:10.1007/3-540-45346-6_5.
- [36] S. Heston and G. Zhou. On the rate of convergence of discrete-time contingent claims. *Mathematical Finance*, 10(1):53–75, 2000. doi:10.1111/1467-9965.00080.
- [37] J. C. Hull. *Options futures and other derivatives*. Pearson Education India, 2003.
- [38] R. A. Jarrow and A. T. Rudd. *Option pricing*. Richard D. Irwin, 1983.

- [39] M. H. Kalos and P. A. Whitlock. *Monte Carlo methods*. John Wiley & Sons, 2009. doi:10.1002/9783527626212.
- [40] K. Kaneko, K. Miyamoto, N. Takeda, and K. Yoshino. Quantum pricing with a smile: Implementation of local volatility model on quantum computer, 2020. arXiv:2007.01467.
- [41] K. Kaneko, K. Miyamoto, N. Takeda, and K. Yoshino. Quantum speedup of Monte Carlo integration in the direction of dimension and its application to finance. *Quantum Inf Processing*, 20:185, 2021. doi:10.1007/s11128-021-03127-8.
- [42] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer Science & Business Media, 2013. doi:10.1007/978-3-662-12616-5.
- [43] D. F. Kuznetsov. Strong numerical methods of orders 2.0, 2.5, and 3.0 for ito stochastic differential equations, based on the unified stochastic taylor expansions and multiple fourier-legendre series, 2018. arXiv:1807.02190.
- [44] D. P. J. Leisen and M. Reimer. Binomial models for option valuation-examining and improving convergence. *Applied Mathematical Finance*, 3(4):319–346, 1996. doi:10.1080/13504869600000015.
- [45] L. Lin and Y. Tong. Optimal quantum eigenstate filtering with application to solving quantum linear systems. *Quantum*, 4:361, 2020. doi:10.22331/q-2020-11-11-361.
- [46] N. Linden, A. Montanaro, and C. Shao. Quantum vs. classical algorithms for solving the heat equation, 2020. arXiv:2004.06516.
- [47] G. Liu and L. J. Hong. Kernel estimation of the greeks for options with discontinuous payoffs. *Operations Research*, 59:iii–266, 2010. doi:10.1287/opre.1100.0844.
- [48] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations, 2020. arXiv:2011.03185.
- [49] P. Malliavin and A. Thalmaier. *Stochastic Calculus of Variations in Mathematical Finance*. Springer-Verlag Berlin Heidelberg, 2006. doi:10.1007/3-540-30799-0.
- [50] X. Mao. *Stochastic differential equations and applications*. Elsevier, 2007. doi:10.1533/9780857099402.
- [51] A. Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015. doi:10.1098/rspa.2015.0301.
- [52] A. Montanaro and S. Pallister. Quantum algorithms and the finite element method. *Physical Review A*, 93(3):032324, 2016. doi:10.1103/PhysRevA.93.032324.
- [53] R. Orus, S. Mugel, and E. Lizaso. Quantum computing for finance: overview and prospects. *Reviews in Physics*, 4:100028, 2019. doi:10.1016/j.revip.2019.100028.
- [54] S. H. Paskov and J. F. Traub. Faster valuation of financial derivatives. *The Journal of Portfolio Management*, 22(1):113–123, 1995. doi:10.3905/jpm.1995.409541.
- [55] P. Rebentrost, B. Gupt, and T. R. Bromley. Quantum computational finance: Monte Carlo pricing of financial derivatives. *Physical Review A*, 98(2):022321, 2018. doi:10.1103/PhysRevA.98.022321.
- [56] R. J. Rendleman. Two-state option pricing. *The Journal of Finance*, 34(5):1093–1110, 1979. doi:doi.org/10.2307/2327237.
- [57] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner. Option pricing using quantum computers. *Quantum*, 4:291, 2020. doi:10.22331/q-2020-07-06-291.

- [58] Y. Subaşı, R. D. Somma, and D. Orsucci. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Physical Review Letters*, 122(6):060504, 2019. [doi:10.1103/PhysRevLett.122.060504](https://doi.org/10.1103/PhysRevLett.122.060504).
- [59] Y. Tong, D. An, N. Wiebe, and L. Lin. Fast inversion, preconditioned quantum linear system solvers, and fast evaluation of matrix functions, 2020. [arXiv:2008.13295](https://arxiv.org/abs/2008.13295).
- [60] G. Xu, A. J. Daley, P. Givi, and R. D. Somma. Turbulent mixing simulation via a quantum algorithm. *AIAA Journal*, 56(2):687–699, 2018. [doi:10.2514/1.J055896](https://doi.org/10.2514/1.J055896).
- [61] G. Xu, A. J. Daley, P. Givi, and R. D. Somma. Quantum algorithm for the computation of the reactant conversion rate in homogeneous turbulence. *Combustion Theory and Modelling*, 23(6):1090–1104, 2019. [doi:10.1080/13647830.2019.1626025](https://doi.org/10.1080/13647830.2019.1626025).

A Different types of errors

Let a random variable Y be an estimator of some unknown quantity a . In this paper we have considered two different types of errors, mean-squared error $\mathbb{E}[(Y - a)^2]$ and additive error $|Y - a|$. Here we will show that these two types of errors are indeed almost equivalent. More precisely, we will show that, up to some absolute pre-constants in the errors and logarithmic factors in the cost, the situation that mean-squared error is on the level of ϵ^2 indicates that the additive error is on the level of ϵ with probability at least 0.99, and *vice versa*.

Proposition 5. *Let $\mathcal{A}$ be a (classical or quantum) algorithm that generates a random variable Y to estimate some unknown quantity a . Assuming that $\mathbb{E}[|Y|^{2+\delta}] < \infty$ for some $\delta > 0$, we have:*

1. *If $\mathbb{E}[(Y - a)^2] \leq \epsilon^2$, then there exists an algorithm which repeats $\mathcal{A}$ a constant number of times and outputs $\hat{Y}$ such that $|\hat{Y} - a| \leq 3\epsilon$ with probability at least 0.99.*
2. *If $|Y - a| \leq \epsilon$ holds with probability at least 0.99, then there exists an algorithm which repeats $\mathcal{A}$ $O(\log(1/\epsilon))$ times and outputs $\hat{Y}$ such that $\mathbb{E}[(\hat{Y} - a)^2] \leq 2\epsilon^2$.*

Proof. 1. By $\mathbb{E}[(Y - a)^2] \leq \epsilon^2$ and the bias-variance decomposition

$$\mathbb{E}[(Y - a)^2] = \mathbb{E}[(Y - \mathbb{E}[Y])^2] + (\mathbb{E}[Y] - a)^2, \quad (\text{A.1})$$

we have $\mathbb{E}[(Y - \mathbb{E}[Y])^2] \leq \epsilon^2$ and $|\mathbb{E}[Y] - a| \leq \epsilon$. Since

$$|Y - \mathbb{E}[Y]| \geq |Y - a| - |a - \mathbb{E}[Y]| \geq |Y - a| - \epsilon, \quad (\text{A.2})$$

the event $\{|Y - a| \geq 3\epsilon\}$ is a subset of the event $\{|Y - \mathbb{E}[Y]| \geq 2\epsilon\}$. Together with Chebyshev's inequality, we have

$$\begin{aligned} \mathbb{P}(|Y - a| \geq 3\epsilon) &\leq \mathbb{P}(|Y - \mathbb{E}[Y]| \geq 2\epsilon) \\ &\leq \frac{\mathbb{E}[(Y - \mathbb{E}[Y])^2]}{(2\epsilon)^2} \\ &\leq \frac{1}{4}. \end{aligned} \quad (\text{A.3})$$

This indicates that a single sample of Y can estimate a up to additive error 3ϵ with probability at least 3/4. Then by Lemma 2, it suffices to repeat $\mathcal{A}$ a constant number of times to boost the success probability to 0.99. This completes the proof of the first part.

2. First, according to Lemma 2, we can construct an estimator $\hat{Y}$ such that $|\hat{Y} - a| \leq \epsilon$ with probability at least

$$1 - 2^{-2-2/\delta} \left(\mathbb{E}|Y|^{2+\delta} + |a|^{2+\delta} \right)^{-2/\delta} \epsilon^{2+4/\delta}$$

by repeating the algorithm $\mathcal{A}$ $O(\log(1/\epsilon))$ times. Let ψ_A be the indicator function of a set A , i.e. $\psi_A(x) = 1$ if $x \in A$ and $\psi_A(x) = 0$ if $x \notin A$. We split the mean-squared error into two parts,

$$\begin{aligned} \mathbb{E}[(\hat{Y} - a)^2] &= \int (y - a)^2 d\mathbb{P}_{\hat{Y}}(y) \\ &= \int (y - a)^2 \psi_{\{|y-a|<\epsilon\}}(y) d\mathbb{P}_{\hat{Y}}(y) + \int (y - a)^2 \psi_{\{|y-a|\geq\epsilon\}}(y) d\mathbb{P}_{\hat{Y}}(y). \end{aligned} \quad (\text{A.4})$$

The first integral can be bounded as

$$\int (y-a)^2 \psi_{\{|y-a|<\epsilon\}}(y) d\mathbb{P}_{\hat{Y}}(y) \leq \int \epsilon^2 \psi_{\{|y-a|<\epsilon\}}(y) d\mathbb{P}_{\hat{Y}}(y) \leq \int \epsilon^2 d\mathbb{P}_{\hat{Y}}(y) = \epsilon^2. \quad (\text{A.5})$$

By Hölder's inequality, the second integral can be bounded as

$$\begin{aligned} & \int (y-a)^2 \psi_{\{|y-a|\geq\epsilon\}}(y) d\mathbb{P}_{\hat{Y}}(y) \\ & \leq \left[\int |y-a|^{2+\delta} d\mathbb{P}_{\hat{Y}}(y) \right]^{\frac{2}{2+\delta}} \left[\int \psi_{\{|y-a|\geq\epsilon\}}(y) d\mathbb{P}_{\hat{Y}}(y) \right]^{\frac{\delta}{2+\delta}} \\ & = \left[\int |y-a|^{2+\delta} d\mathbb{P}_{\hat{Y}}(y) \right]^{\frac{2}{2+\delta}} \left[\mathbb{P}(|\hat{Y}-a| \geq \epsilon) \right]^{\frac{\delta}{2+\delta}} \\ & \leq 2^{-\frac{2+2\delta}{2+\delta}} \left(\mathbb{E}|Y|^{2+\delta} + |a|^{2+\delta} \right)^{-\frac{2}{2+\delta}} \epsilon^2 \left[\int |y-a|^{2+\delta} d\mathbb{P}_{\hat{Y}}(y) \right]^{\frac{2}{2+\delta}} \\ & \leq 2^{-\frac{2+2\delta}{2+\delta}} \left(\mathbb{E}|Y|^{2+\delta} + |a|^{2+\delta} \right)^{-\frac{2}{2+\delta}} \epsilon^2 \left[\int (|y|+|a|)^{2+\delta} d\mathbb{P}_{\hat{Y}}(y) \right]^{\frac{2}{2+\delta}} \\ & \leq \left(\mathbb{E}|Y|^{2+\delta} + |a|^{2+\delta} \right)^{-\frac{2}{2+\delta}} \epsilon^2 \left[\int |y|^{2+\delta} d\mathbb{P}_{\hat{Y}}(y) + \int |a|^{2+\delta} d\mathbb{P}_{\hat{Y}}(y) \right]^{\frac{2}{2+\delta}} \\ & = \left(\mathbb{E}|Y|^{2+\delta} + |a|^{2+\delta} \right)^{-\frac{2}{2+\delta}} \epsilon^2 \left[\mathbb{E}|Y|^{2+\delta} + |a|^{2+\delta} \right]^{\frac{2}{2+\delta}} \\ & = \epsilon^2. \end{aligned} \quad (\text{A.6})$$

Therefore the mean-squared error $\mathbb{E}[(\hat{Y}-a)^2]$ is bounded by $2\epsilon^2$. This completes the proof of the second part. $\square$

B Numerical Results

In this part, we test several numerical schemes of SDE on Black-Scholes option pricing model to obtain a classical estimate of the parameters α and β in [Proposition 2](#). We consider five numerical schemes, including Euler-Maruyama scheme (4.4), Milstein scheme (4.5), strong order 1.5 scheme (B.1), strong order 2 scheme (B.5) (B.9) and strong order 3 scheme (B.11). The strong order 1.5 scheme is based on Taylor-Itô expansion

$$\begin{aligned} \hat{X}_{k+1} = & \hat{X}_k + \mu_k h + \sigma_k \Delta W_k + \sigma_k \sigma'_k I_{(1,1)} + \sigma_k \mu'_k I_{(1,0)} + (\mu_k \mu'_k + \frac{1}{2} \sigma_k^2 \mu_k) \frac{h^2}{2} \\ & + (\mu_k \sigma'_k + \frac{1}{2} \sigma_k^2 \sigma''_k) I_{(0,1)} + \sigma_k (\sigma_k \sigma''_k + (\sigma'_k)^2) I_{(1,1,1)}. \end{aligned} \quad (\text{B.1})$$

Here,

$$\begin{aligned} \mu_k &:= \mu(\hat{X}_k, t), & \mu'_k &:= \partial_X \mu(\hat{X}_k, t), \\ \sigma_k &:= \sigma(\hat{X}_k, t), & \sigma'_k &:= \partial_X \sigma(\hat{X}_k, t), & \sigma''_k &:= \partial_X^2 \sigma(\hat{X}_k, t). \end{aligned} \quad (\text{B.2})$$

Without other notice, for any function $f := f(x, t)$, f' denote the partial derivative of function f with respect to x . $I_{(1,1)}$, $I_{(1,0)}$, $I_{(0,1)}$ and $I_{(1,1,1)}$ are multiple Itô integrals:

$$\begin{aligned}
I_{(1)} &= \int_{t_n}^{t_{n+1}} dW_{s_1} = \Delta W_k, \\
I_{(1,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} dW_{s_1} dW_{s_2} = \frac{1}{2}(I_{(1)}^2 - h), \\
I_{(1,0)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} dW_{s_1} ds_2, \\
I_{(0,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} ds_1 dW_{s_2} = hI_{(1)} - I_{(1,0)}, \\
I_{(1,1,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_3} \int_{t_n}^{s_2} dW_{s_1} dW_{s_2} dW_{s_3} = \frac{1}{2}(\frac{1}{3}I_{(1)}^3 - h)I_{(1)}.
\end{aligned} \tag{B.3}$$

In practice, we use two independent $\mathcal{N}(0,1)$ distributed random variables U_1 and U_2 to approximate $I_{(1)}$ and $I_{(1,0)}$ with

$$I_{(1)} = \sqrt{h}U_1, \quad I_{(1,0)} = \frac{1}{2}h^{\frac{3}{2}}(U_1 + \frac{1}{\sqrt{3}}U_2). \tag{B.4}$$

The integer strong order Taylor schemes can be conveniently derived from a Taylor-Stratonovich expansion, and thus we use the following strong order 2 scheme [43]:

$$\begin{aligned}
\hat{X}_{k+1} &= \hat{X}_k + \bar{\mu}_k h + \sigma_k \Delta W_k + \sigma_k \sigma'_k J_{(1,1)} + \sigma_k \bar{\mu}'_k J_{(1,0)} + \bar{\mu}_k \sigma'_k J_{(0,1)} + \bar{\mu}_k \bar{\mu}'_k \frac{h^2}{2} \\
&+ \sigma_k (\sigma_k \sigma'_k)' J_{(1,1,1)} + \bar{\mu}_k (\sigma_k \sigma'_k)' J_{(0,1,1)} + \sigma_k (\bar{\mu}_k \sigma'_k)' J_{(1,0,1)} + \sigma_k (\sigma_k \bar{\mu}'_k)' J_{(1,1,0)} \\
&+ \sigma_k (\sigma_k (\sigma_k \sigma'_k)')' J_{(1,1,1,1)},
\end{aligned} \tag{B.5}$$

where

$$\bar{\mu} = \mu - \frac{1}{2}\sigma\sigma', \tag{B.6}$$

and $J_{(1,1)}, J_{(1,0)}, J_{(0,1)}, J_{(1,1,1)}, J_{(1,1,0)}, J_{(1,0,1)}, J_{(0,1,1)}$ and $J_{(1,1,1,1)}$ are multiple Stratonovich integrals:

$$\begin{aligned}
J_{(1)} &= \int_{t_n}^{t_{n+1}} \circ dW_{s_1} = I_{(1)} = \Delta W_k, \\
J_{(1,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} \circ dW_{s_1} \circ dW_{s_2} = \frac{1}{2!}(J_{(1)})^2, \\
J_{(1,0)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} \circ dW_{s_1} ds_2, \\
J_{(0,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} ds_1 \circ dW_{s_2}, \\
J_{(1,1,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_3} \int_{t_n}^{s_2} \circ dW_{s_1} \circ dW_{s_2} \circ dW_{s_3} = \frac{1}{3!}(J_{(1)})^3, \\
J_{(1,1,0)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_3} \int_{t_n}^{s_2} ds_1 \circ dW_{s_2} \circ dW_{s_3}, \\
J_{(1,0,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_3} \int_{t_n}^{s_2} \circ dW_{s_1} ds_2 \circ dW_{s_3}, \\
J_{(0,1,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_3} \int_{t_n}^{s_2} \circ dW_{s_1} \circ dW_{s_2} ds_3, \\
J_{(1,1,1,1)} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_3} \int_{t_n}^{s_3} \int_{t_n}^{s_2} \circ dW_{s_1} \circ dW_{s_2} \circ dW_{s_3} \circ dW_{s_4} = \frac{1}{4!}(J_{(1)})^4.
\end{aligned} \tag{B.7}$$

We remark that there also exists the strong order 2 scheme derived from Taylor-Itô expansion [42]. Typically, we approximate multiple Stratonovich integrals and multiple Itô integrals by introducing additional random variables. However, the approximation of multiple Stratonovich integrals is simpler and requires minimal set of random variables. That is also why we prefer high order numerical schemes derived from Taylor-Stratonovich expansion, such as the strong order 2 scheme and strong order 3 scheme in the following.

Since we consider Black-Scholes model,

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \quad (\text{B.8})$$

the strong order 2 scheme can be simplified to

$$\begin{aligned} \hat{X}_{k+1} = & \hat{X}_k + \bar{\mu} \hat{X}_k h + \sigma \hat{X}_k \Delta W_k + \sigma^2 \hat{X}_k J_{(1,1)} + \bar{\mu} \sigma \hat{X}_k h \Delta W_k + \bar{\mu}^2 \hat{X}_k \frac{h^2}{2} \\ & + \sigma^3 \hat{X}_k J_{(1,1,1)} + \bar{\mu} \sigma^2 \hat{X}_k h J_{(1,1)} + \sigma^4 \hat{X}_k J_{(1,1,1,1)}, \end{aligned} \quad (\text{B.9})$$

where $\bar{\mu} = \mu - \frac{1}{2}\sigma^2$ and we use a nice property of multiple Stratonovich integral that

$$J_{(1,0)} + J_{(0,1)} = hJ_{(1)}, \quad J_{(1,1,0)} + J_{(1,0,1)} + J_{(0,1,1)} = hJ_{(1,1)}. \quad (\text{B.10})$$

The strong order 3 schemes for general SDEs are thoroughly discussed in [43]. We choose the strong order 3.0 scheme based on Taylor-Stratonovich expansion and the corresponding discretization scheme for (B.8) is as follow.

$$\begin{aligned} \hat{X}_{k+1} = & \hat{X}_k + \bar{\mu} \hat{X}_k h + \sigma \hat{X}_k \Delta W_k + \sigma^2 \hat{X}_k J_{(1,1)} + \bar{\mu} \sigma \hat{X}_k h \Delta W_k + \bar{\mu}^2 \hat{X}_k \frac{h^2}{2} + \sigma^3 \hat{X}_k J_{(1,1,1)} \\ & + \bar{\mu} \sigma^2 \hat{X}_k h J_{(1,1)} + \sigma^4 \hat{X}_k J_{(1,1,1,1)} + \bar{\mu}^2 \sigma \hat{X}_k \frac{h^2}{2} \Delta W_k + \bar{\mu} \sigma^3 \hat{X}_k h J_{(1,1,1)} + \sigma^5 \hat{X}_k J_{(1,1,1,1,1)} \\ & + \bar{\mu}^3 \hat{X}_k \frac{h^3}{6} + \sigma^6 \hat{X}_k J_{(1,1,1,1,1,1)} + \bar{\mu}^2 \sigma^2 \hat{X}_k \frac{h^2}{2} J_{(1,1)} + \bar{\mu} \sigma^4 \hat{X}_k h J_{(1,1,1,1)} \end{aligned} \quad (\text{B.11})$$

where $J_{(1,1,1,1,1)}$ and $J_{(1,1,1,1,1,1)}$ are multiple Stratonovich integrals with

$$J_{(1,1,1,1,1)} = \frac{J_{(1)}}{5!}, \quad J_{(1,1,1,1,1,1)} = \frac{J_{(1)}}{6!}. \quad (\text{B.12})$$

As for options, we consider European option and Digital option pricing. For European option, the payoff function is

$$\psi(S_T) = \exp(-\mu T) \max\{S_T - K, 0\} \quad (\text{B.13})$$

For Digital option, the payoff function is

$$\psi(S_T) = 5 \exp(-\mu T) (1 + \mathcal{H}(S_T - K)) \quad (\text{B.14})$$

where $\mathcal{H}$ is Heaviside step function.

Table 2, 3 and Figure 1, 2, 3, 4, 5 show our numerical results on estimating α and β as well as the convergence performance of the errors at each layer of MLMC. The five schemes mentioned above are tested by running respectively 10^6 , 10^6 , 10^7 , 10^8 , 10^9 independent simulations with the following choice of parameters:

$$\mu = 0.05, \quad \sigma = 0.2, \quad T = 1, \quad S_0 = 100, \quad K = 100. \quad (\text{B.15})$$

Option	Euler Maruyama	Milstein	strong order 1.5	strong order 2	strong order 3
European	0.976999	1.962848	2.970166	3.964626	5.958417
Digital	0.473426	0.869393	1.452448	1.775679	NAN

Table 2: numerical estimates of β based on linear regression for five schemes

Option	Euler Maruyama	Milstein	strong order 1.5	strong order 2	strong order 3
European	1.136214	0.979572	1.747239	1.970829	2.961041
Digital	1.023176	0.791818	1.853158	1.827618	NAN

Table 3: numerical estimates of α based on linear regression for five schemes

The numerical scalings of α and β are estimated using linear regression based on the last four points, in which the time step size h is small enough such that the corresponding numerical scheme has already well converged (as shown in the figures) and the numerical errors are dominated by the leading order term.

Table 2 and Table 3 shows the estimate of β and α of these schemes respectively. In the case of European option, the estimates of β for five schemes all approximately equal twice the corresponding strong order, which implies that our theoretical estimate for β in Proposition 2 is sharp for Lipschitz continuous functions. The estimates of α with higher order schemes are approximately equal to the corresponding strong order, which also agrees well with our theoretical estimate for α . In the case of Digital option, other than the strong order 3 scheme (which will be discussed later), the estimates of β are roughly equal to the corresponding strong order of the schemes, which again verifies our theoretical results for non-Lipschitz payoff functions. We remark that, similarly to the European option case, some of the estimates for α are larger than the strong order. This is because our theoretical estimate for α is quite conservative that we only employ properties of strong order, while the convergence of payoff function is more related to the weak order, which describes the convergence order of the moments of the stochastic process. This also implies that our theoretical estimate for α might be improvable for particular schemes and payoff functions.

We notice that for Digital option, strong order 3 scheme cannot output the estimate of α and β . That is because strong order 3 scheme is of pretty high accuracy. When we compute the option price, since the expectation of S_T under our parameter choices is larger than K , the Heaviside step function $\mathcal{H}$ outputs 1 with extremely high probability and the option price is nearly deterministic. The outcome of numerical experiment has such a small variance that the influence of the increase in level l on the mean and variance is hard to distinguish, which explains the results shown in Figure 5.

So we make some changes to the choice of parameters. We increase σ to 1.5, which increase the randomness, and choose K to be exactly the theoretical expectation $100e^{-0.05}$. The numerical results are shown in Figure 6. The estimate of β is 2.957982, which agrees well with theoretical result, and the estimate of α is 2.438328, which might be estimated more accurately by using more samplings as well as smaller time step sizes.

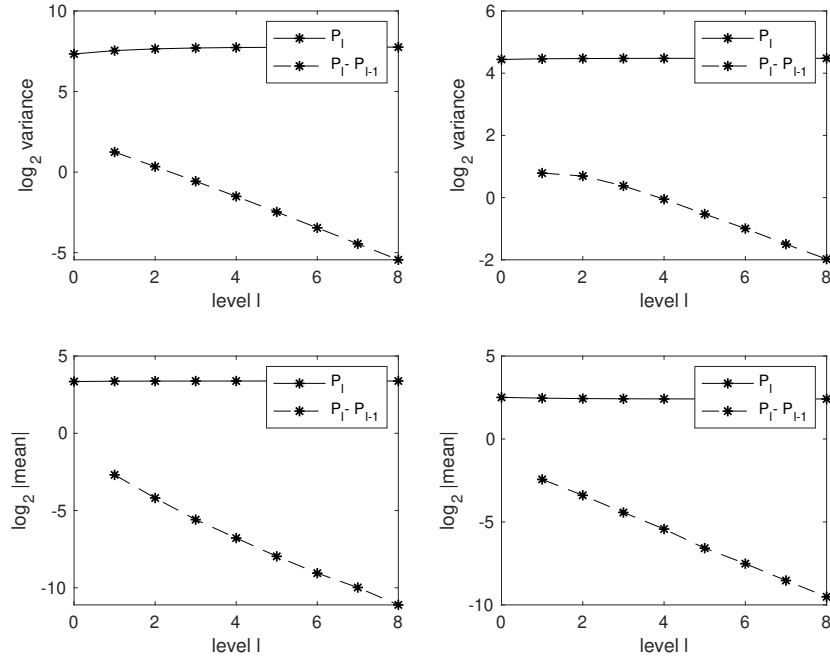


Figure 1: estimates of β and α when using Euler Maruyama Scheme for European option (left) and Digital option (right). The top two plots show the variance versus level l , and the slope gives an estimate of β . The bottom two plots show the mean versus level l and the slope gives an estimate of α .

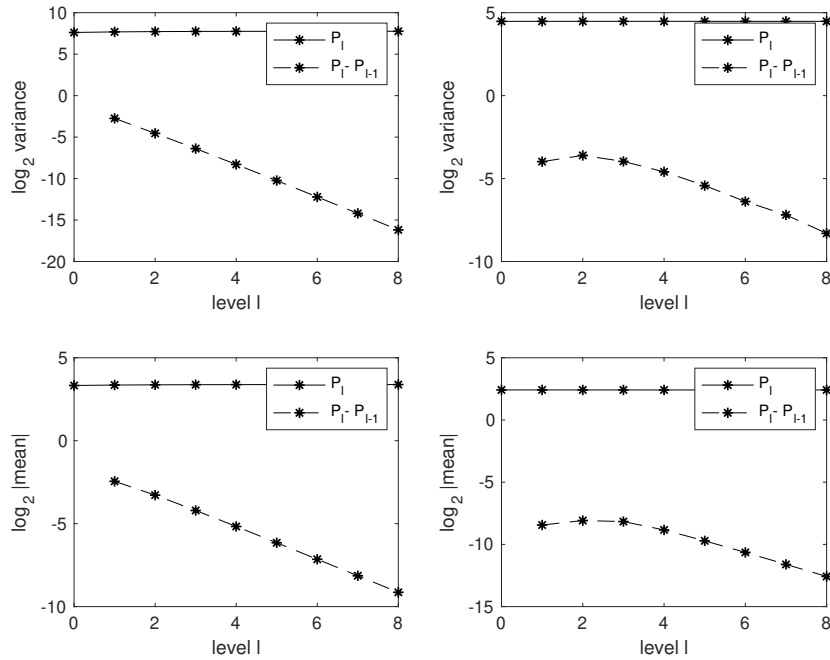


Figure 2: estimates of β and α when using Milstein Scheme for European option (left) and Digital option (right). The top two plots show the variance versus level l , and the slope gives an estimate of β . The bottom two plots show the mean versus level l and the slope gives an estimate of α .

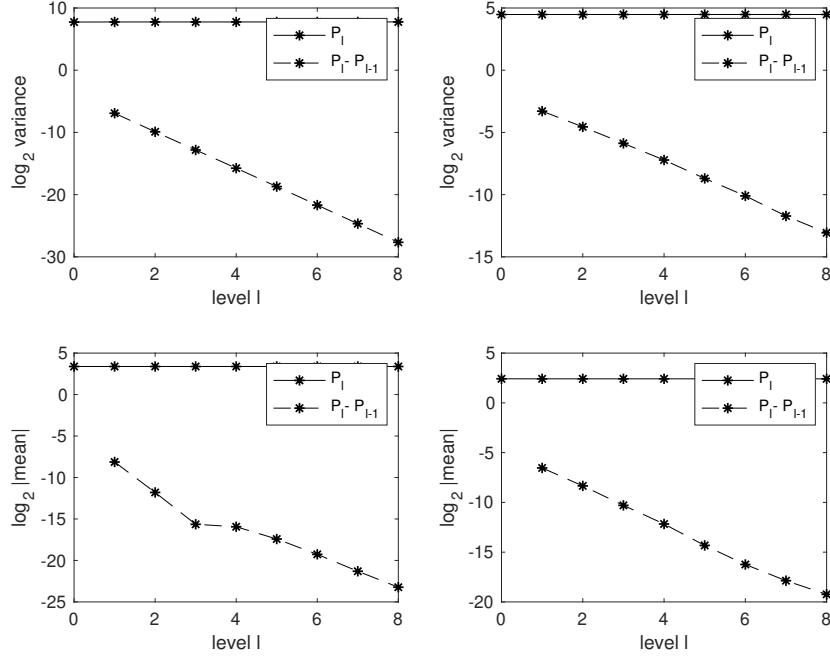


Figure 3: estimates of β and α when using Strong order 1.5 Scheme for European option (left) and Digital option (right). The top two plots show the variance versus level l , and the slope gives an estimate of β . The bottom two plots show the mean versus level l and the slope gives an estimate of α .

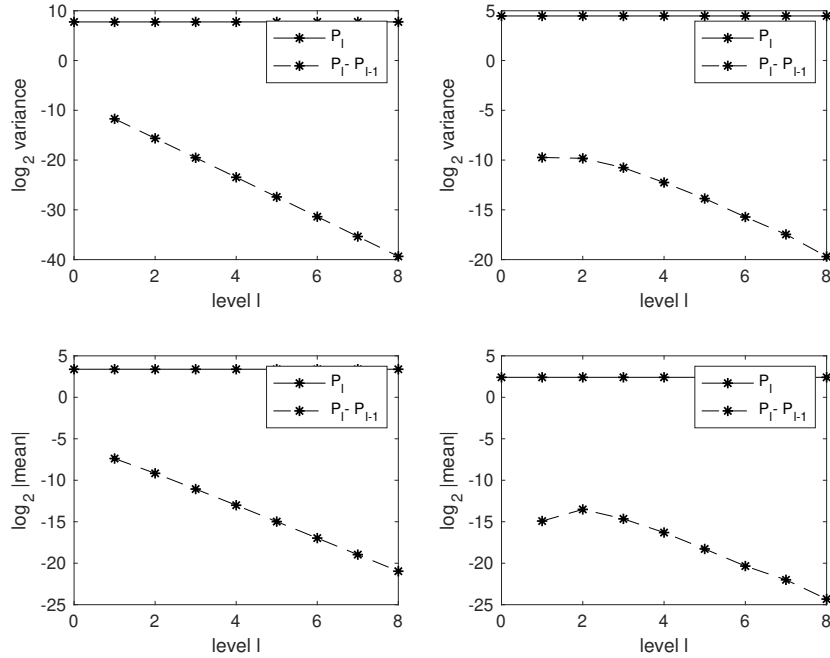


Figure 4: estimates of β and α when using Strong order 2 Scheme for European option (left) and Digital option (right). The top two plots show the variance versus level l , and the slope gives an estimate of β . The bottom two plots show the mean versus level l and the slope gives an estimate of α .

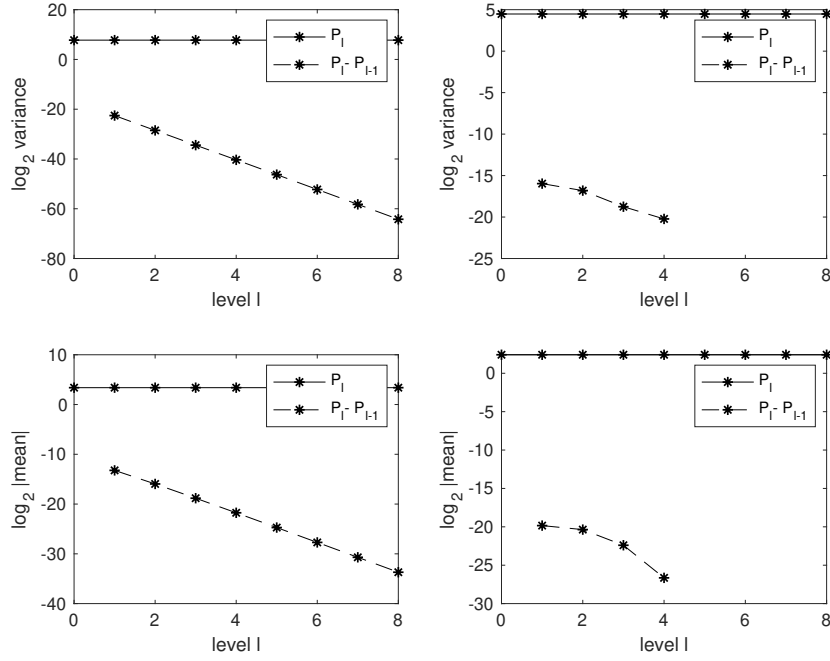


Figure 5: estimates of β and α when using Strong order 3 Scheme for European option (left) and Digital option (right). The top two plots show the variance versus level l , and the slope gives an estimate of β . The bottom two plots show the mean versus level l and the slope gives an estimate of α .

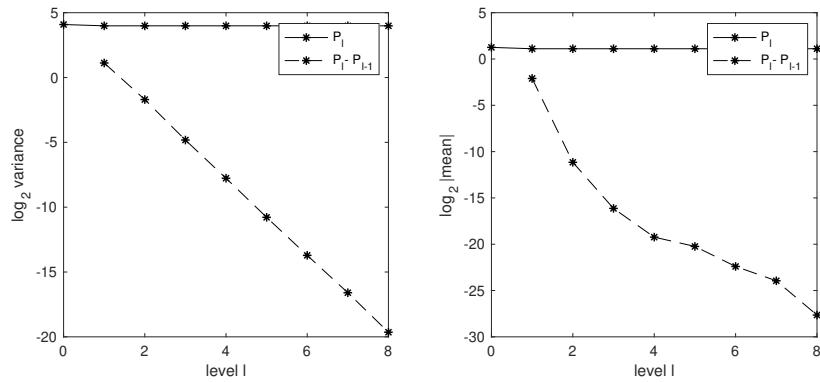


Figure 6: estimates of β and α when using Strong order 3 Scheme for Digital option with new parameters $\sigma = 1.5$ and $K = 100e^{-0.05}$. The left plot shows the variance versus level l , and the slope gives an estimate of β . The right plot shows the mean versus level l and the slope gives an estimate of α .