Variational quantum simulations of stochastic differential equations

Kenji Kubo,1,2,† Yuya O. Nakagawa,3‡ Suguru Endo,4§ and Shota Nagayama1∥

1R&D, Mercari Inc., Roppongi Hills Mori Tower 18F, 6-10-1, Roppongi, Minato-ku, Tokyo 106-6118, Japan
2Graduate School of Engineering Science, Osaka University, 1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan
3QunaSys Inc., Aqua Hakusan Building 9F, 1-13-7 Hakusan, Bunkyo, Tokyo 113-0001, Japan
4NTT Secure Platform Laboratories, NTT Corporation, Musashino, Tokyo 180-8585, Japan

(Dated: March 18, 2021)

Stochastic differential equations (SDEs), which models uncertain phenomena as the time evolution of random variables, are exploited in various fields of natural and social sciences such as finance. Since SDEs rarely admit analytical solutions and must usually be solved numerically with huge classical-computational resources in practical applications, there is strong motivation to use quantum computation to accelerate the calculation. Here, we propose a quantum-classical hybrid algorithm that solves SDEs based on variational quantum simulation (VQS). We first approximate the target SDE by a trinomial tree structure with discretization and then formulate it as the time-evolution of a quantum state embedding the probability distributions of the SDE variables. We embed the probability distribution directly in the amplitudes of the quantum state while the previous studies did the square-root of the probability distribution in the amplitudes. Our embedding enables us to construct simple quantum circuits that simulate the time-evolution of the state for general SDEs. We also develop a scheme to compute the expectation values of the SDE variables and discuss whether our scheme can achieve quantum speed-up for the expectation-value evaluations of the SDE variables. Finally, we numerically validate our algorithm by simulating several types of stochastic processes. Our proposal provides a new direction for simulating SDEs on quantum computers.

I. INTRODUCTION

Stochastic differential equations (SDEs), which describe the time evolution of random variables, are among the most important mathematical tools for modeling uncertain systems in diverse fields, such as finance [1], physics [2], and biology [3]. From the expectation values of the simulated random variables, we can often extract information about the system of interest. Since the expectation values rarely admit analytical solutions, they are usually obtained by numerical methods such as the Monte Carlo method [4]. However, those numerical methods incur high computational costs, especially in high-dimensional problems such as the SDEs of financial applications [5–7]. Therefore, a method that can speed-up SDE simulations is urgently demanded.

Such a speed up can be achieved on quantum computers. Throughout the past decade, technological developments have realized a primitive form of quantum computers called noisy intermediate-scale quantum (NISQ) devices [8], which can handle problems that are intractably large for classical computers [9]. NISQ devices can operate only a few tens to hundreds of qubits without error correction, so they cannot run quantum algorithms requiring deep and complicated quantum circuits. Although quantum algorithms are expected to outperform classical ones on specific computing tasks [10–13], they usually exceed the capability of NISQ devices. Accordingly, NISQ devices have been leveraged with heuristic algorithms that solve real-world problems. For example, in quantum chemistry and condensed matter physics, the variational quantum eigensolver (VQE) algorithm [14, 15] can calculate the ground-state energies of given Hamiltonians [16, 17]. Another example is quantum machine learning with variational quantum circuits [18–21]. Both algorithms variationally optimize the tuneable classical parameters in quantum circuits, so the speedups of the computation over classical computers and the precision of the obtained results are not guaranteed in general.

Several quantum-computing-based methods obtain the expectation value of a function that takes an SDE solution as its argument. However, all of these methods require prerequisite knowledge of the SDE solution. In [22], the partial differential equation describing the time evolution of the expectation value was simulated by a variational quantum computation, which requires pre-derivation of the partial differential equation of the expectation value. In [23] and [24], the probability distribution of the SDE solution was embedded in the quantum state, and the expectation value was calculated by a quantum amplitude estimation algorithm (QAE). In this case, the probability distribution of the SDE solutions must be known in advance. As the solution to the SDE is not found, the partial differential equation of the expectation value must also be derived, or the SDE solved beforehand.

In this study, to solve an SDE with quantum algo-
rithms, we apply a tree model approximation [25], and hence obtain a linear differential equation describing the probability distribution of SDE solutions. This differential equation is then solved by a variational quantum simulation (VQS) [26-29]. Note that linear differential equations can be solved by a quantum linear solver algorithm (QLSA) [12, 31, 32], which is expected to be quantum-accelerated. However, the QLSA requires a large number of ancilla qubits and deep circuits and is possibly executable only on quantum computers with error correction. Our proposed method possesses several desirable features. First, the probability distribution is simulated by the tree-model approximation, so the model requires only the SDE. No prior knowledge of the probability distribution or expectation value is required. Therefore, our method is applicable to more general SDEs than previous methods. Second, once the VQS is performed, the variational parameters are obtained as classical information, and the probability distribution of the simulation results can be used to compute various expectation values. We can also compute path-dependent expectation values because the time series of the probability distribution is obtained. Third, the algorithm is less resource-intensive than the QLSA. Since VQS is a variational algorithm, it is difficult to estimate the exact computational cost, but VQS requires only a few ancilla qubits and calculates the expectation value for relatively shallow unitary gates at each time step. The number of qubits and the depth of the circuit are expected to be much smaller than QLSA. As our method uses a new scheme for embedding probability distributions in quantum states, the method for computing expectation values is also new. We additionally found that the expectation values are more simply determined by our method than by the QAE. The proposed method facilitates the application of SDEs in quantum computing simulations and is expected to impact various scientific fields.

The remainder of this paper is organized as follows. Section II reviews the main ingredients of this paper: the trinomial tree-model approximation of the SDE [25] and the VQS algorithm [26-29]. In Sec. III we combine both ingredients into a method that simulates the SDE by the VQS.

A. Trinomial tree-model approximation of the stochastic differential equation

Let us consider a random variable $X(t)$ taking values on an interval $I \subset \mathbb{R}$. We refer to $I$ as an event space. The SDE of a single process \{\(X(t)\)\} \(t \in [0,T]\), which is a time-series of random variables from \(t = 0\) to \(t = T\), is defined as [1]

\[
dX(t) = \mu(X(t), t)dt + \sigma(X(t), t)dW, \quad X(0) = x_{\text{ini}} \in I,
\]

where \(\mu(X(t), t), \sigma(X(t), t)\) are real valued functions of time \(t\) and the variable \(X(t)\), and \(W\) denotes the Brownian motion. In the main text, our proposal is applied to a single process (extensions to multi-variables cases are described in Appendix B).

The tree model numerically simulates the time evolution of an SDE. Let us consider an SDE simulation of the process with event space \([0, x_{\text{max}}]\) from \(t = 0\) to \(t = T\). We discretize the time as \(t_i \equiv i\Delta t\) \((i = 0, 1, \ldots, N_t)\) and the event space as \(x_i \equiv i\Delta x\) \((i = 0, 1, \ldots, N_x)\), where \(N_t\Delta t = T\) and \(N_x\Delta x = x_{\text{max}}\). In this discretization scheme, we define a \((N_x + 1) \times (N_t + 1)\) lattice on which each node \((i,j)\) is associated with a probability \(\text{Prob}(X(t_j) = x_i)\) and each edge represents a transition.
between two nodes, as shown in Fig. 1. Here, we adopt the trinomial tree model, which has three transition probabilities as follows:

\[ p_u(x,t) = \text{Prob}[X(t + \Delta t) = x + \Delta x \mid X(t) = x], \]
\[ p_d(x,t) = \text{Prob}[X(t + \Delta t) = x - \Delta x \mid X(t) = x], \]
\[ p_m(x,t) = \text{Prob}[X(t + \Delta t) = x \mid X(t) = x]. \]

These probabilities were chosen to reproduce the first and second moments (mean and variance, respectively) of the random variable \( X(t) \) in Eq. (1). Following the Euler-Maruyama method [2], the SDE is discretized as

\[ X(t_{j+1}) - X(t_j) = \mu(X(t_j), t) \Delta t + \sigma(X(t_j), t) \sqrt{\Delta t} z, \quad (2) \]

where \( z \sim N(0,1) \) and \( O(\Delta t^2) \) terms are ignored. The conditional expectation value and variance are respectively expressed as

\[ E[X(t_{j+1}) - X(t_j) | X(t_j) = x] = \mu(x,t_j) \Delta t, \]
\[ \text{Var}[X(t_{j+1}) - X(t_j) | X(t_j) = x] = \sigma^2(x,t_j) \Delta t. \]

The corresponding moments on the trinomial tree model are

\[ E[X(t_{j+1}) - X(t_j) | X(t_j) = x_i] = (p_u(x_i,t_j) - p_d(x_i,t_j)) \Delta x, \]
\[ \text{Var}[X(t_{j+1}) - X(t_j) | X(t_j) = x_i] = (p_u(x_i,t_j) + p_d(x_i,t_j)) \Delta x^2. \]

Equating these moments and considering the normalization condition \( p_u(x,t) + p_m(x,t) + p_d(x,t) = 1 \), we obtain

\[ p_u(x_i,t_j) = \frac{1}{2} \left( \frac{\sigma^2(x_i,t_j) + \mu(x_i,t_j)}{\Delta x^2} \right) \Delta t, \quad (3) \]
\[ p_d(x_i,t_j) = \frac{1}{2} \left( \frac{\sigma^2(x_i,t_j) - \mu(x_i,t_j)}{\Delta x^2} \right) \Delta t, \quad (4) \]
\[ p_m(x_i,t_j) = 1 - \frac{\sigma^2(x_i,t_j)}{\Delta x^2} \Delta t. \quad (5) \]

In summary, the trinomial tree model approximates the original SDE by discretizing it on the lattice and setting the transition probabilities between the nodes to reproduce the first and second moments of the process.

The trinomial tree model simulates the SDE as follows. First, the closest value to \( x_{ini} \) in \( \{x_i\}_{i=0, N_x} \) is set to \( x_{ini} \), and the probabilities are set as \( \text{Prob}[X(t_0) = x_{ini}] = 1, \text{Prob}[X(t_0) = x_{i \neq ini}] = 0 \). Next, the probability distribution of \( X(t_1 = \Delta t) \) is calculated using the transition probabilities given by Eqs. (3) (4) (5). Repeating this step for \( X(t_j) (j = 2, 3, ..., N_t - 1) \) yields all probabilities \( \text{Prob}[X(t_j) = x_i] \) at node \( (i,j) \), from which any properties related to the process \( X(t) \), such as the expectation values of \( X(T) \) under some function \( f \), \( E[f(X(T))] \), can be determined. In option-pricing financial problems, the nodes of the tree model denote the prices of the option, and the problems are sometimes to be solved in the backward direction from time \( t \). In such cases, the boundary condition is set at \( t = T \).

**B. Variational quantum simulation (VQS)**

This subsection introduces the VQS algorithm [26–29], a quantum-classical hybrid algorithm that simulates both unitary and non-unitary time evolution with possibly shallow quantum circuits. Therefore, the VQS algorithms is especially suitable for NISQ devices.

We are interested in the non-unitary time evolution of an unnormalized quantum state \( |\psi(t)\rangle \) on an \( n \)-qubit system, defined as

\[ \frac{d}{dt} |\tilde{\psi}(t)\rangle = L(t) |\tilde{\psi}(t)\rangle, \quad (6) \]

where \( L(t) \) is a time-dependent (possibly non-Hermitian) linear operator. To simulate the dynamics of \( |\tilde{\psi}(t)\rangle \), let us introduce the following ansatz quantum state \( |\tilde{\psi}(\theta(t))\rangle \):

\[ |\tilde{\psi}(\theta(t))\rangle \equiv \alpha(t)R(\theta_1(t))|0\rangle \quad (7) \]

where \( \alpha(t) \) is a real number, \( \theta(t) \equiv (\alpha(t), \theta_1(t), ... , \theta_M(t)) \) are variational parameters of the ansatz, \( |0\rangle \) is some reference state, and \( R(\theta_1) \equiv R_1(\theta_1)R_2(\theta_2) \cdots R_M(\theta_M) \) is a product of \( M \) parametric circuits (unitaries) composed of one parametric rotation gates \( e^{i\theta_k}G_k(G_k^1 = G_k) \). The gates depend on their parameters and on other non-parametric gates. In particular, \( G_k \) is assumed as a multi-qubit Pauli gate \( \{I, X, Y, Z\}^{\otimes n} \).

The VQS algorithm maps the dynamics of the quantum state, Eq. (6), to those of the variational parameters \( \theta(t) \) of the ansatz. The mapping is performed by McLachlan’s variational principle [33]

\[ \min_{\theta(t)} \left\| \frac{d}{dt} |\tilde{\psi}(\theta(t))\rangle - L(t) |\tilde{\psi}(\theta(t))\rangle \right\|, \quad (8) \]

for \( k = 0, \ldots, M \) where

\[ M_{k,j} \equiv \mathbb{R} \left( \frac{\partial (\tilde{\psi}(\theta(t))) \partial (\tilde{\psi}(\theta(t)))}{\partial \theta_k} \right), \quad (9) \]

\[ V_k \equiv \mathbb{R} \left( \frac{\partial (\tilde{\psi}(\theta(t))) \partial (\tilde{\psi}(\theta(t)))}{\partial \theta_k} \right). \quad (10) \]

We define \( \theta_0(t) \equiv \alpha(t) \) for notational simplicity. When simulating the dynamic Eq. (6), one starts from the initial parameters \( \theta_{ini} \) corresponding to the initial state \( |\psi(t=0)\rangle = |\tilde{\psi}(\theta_{ini})\rangle \). The time derivative \( \dot{\theta}(t) = 0 \) is calculated by Eq. (9) with \( |\tilde{\psi}(\theta(t))\rangle \) in Eqs. (10) and (11). After a small time step \( \delta t \), the parameters are obtained as \( \theta(t) = \theta(t) + \delta t \cdot \dot{\theta}(t = 0) \). Repeating this procedure obtains the dynamics of \( \theta(t) \) and the state \( |\tilde{\psi}(\theta(t))\rangle \).

The terms \( M_{k,j} \) and \( V_k \) can be evaluated by the quantum circuits depicted in Fig. 2 [29]. The normalized
state \(|\psi(\theta(t))\rangle \equiv R(\theta(t))|0\rangle\) is actually prepared on quantum computers and is multiplied by the normalization constant \(\alpha(t)\) in post-processes of the result of the circuit measurements. Decomposing the operator \(L(t) = L(t) = \sum_{k=1}^{k_{\text{term}}(t)} \lambda_k U_k\), where \(U_k\) is an easily-implementable unitary operator (e.g., a multi-qubit Pauli operator) and \(\lambda_k\) is a complex coefficient, we must evaluate \(O(M^2) + O(Mk_{\text{term}}(t))\) distinct quantum circuits. The circuits need an ancilla qubit other than the qubits of the system of interest, along with control operations of \(G_k\) and \(U_k\). Therefore, to ensure a feasible VQS algorithm, both \(M, k_{\text{term}}(t)\), and the depth of the unitaries \(U_k\) must be \(O(\text{poly}(n))\).

### III. Solving Stochastic Differential Equations by Variational Quantum Simulation

This section presents one of our main results. The SDE simulated by the above-described trinomial tree model is reformulated as the non-unitary dynamics of a quantum state \(|\psi(t)\rangle\) embedding the probability distribution of the random variable \(X(t)\). We explicitly state for the \(L(t)\) operator of the VQS and decompose it by the polynomial number of the sum of easily-implementable unitaries.

#### A. Embedding the probability distribution into a quantum state

To simulate the trinomial tree model of the target SDE by VQS, we define an unnormalized quantum state containing the discretized probability distribution of the random variable \(X(t_j)\):

\[
|\tilde{\psi}(t)\rangle \equiv \sum_{i=0}^{N_x} \text{Prob}[X(t) = x_i] |i\rangle, \tag{12}
\]

where \(|i\rangle\big|_{i=0}^{N_x}\) is the computational basis. We call this state a directly embedded state. For simplicity, we assume that \(N_x = 2^n - 1\), where \(n\) is the number of qubits.

Note that this embedding of the probability distribution into the quantum state differs from most of the literature, in which (aiming for a quantum advantage) the expectation values of a probability distribution are calculated using QAE [34]. In the literature, the probability distribution is expressed as a normalized quantum state

\[
|\psi_{\text{sqrt}}\rangle \equiv \sum_i \sqrt{\text{Prob}[X(t_j) = x_i]} |i\rangle. \tag{13}
\]

The expectation value of the distribution, \(E[f(X(t))] = \sum_j f(x_i)\text{Prob}[X(t_j) = x_i]\) for some function \(f\), is computed by the QAE. In this embedding method, VQS cannot be used because the differential equation describing the time evolution of the probability distribution is nonlinear. There are ways to solve the nonlinear differential equation with a quantum algorithm [35, 38], but they require more complicated quantum circuits.

Because our embedding [12] differs from this embedding scheme, we also developed a method for evaluating its expectation values (see Sec. IV). Note that the normalization constant \(\alpha\) in Eq. (4) may be exponentially small. In fact, for a uniform distribution \(\text{Prob}[X(t_j) = x_i] = 1/2^n\), the normalization constant is \(2^{-n/2}\).

#### B. Reformulating the trinomial tree model and applying the variational quantum simulation

In the trinomial tree model, the probability \(\text{Prob}[X(t_{j+1}) = x_i]\) is calculated as

\[
\text{Prob}[X(t_{j+1}) = x_i] = p_u(x_{i-1}, t_j)\text{Prob}[X(t_j) = x_{i-1}] + p_d(x_{i+1}, t_j)\text{Prob}[X(t_j) = x_{i+1}] + p_m(x_i, t_j)\text{Prob}[X(t_j) = x_i]. \tag{14}
\]

Substituting the transition probabilities [3, 4] and [5] into this expression and denoting \(P(x, t) \equiv \text{Prob}[X(t) = x]\)

![Quantum circuits for evaluating (a) \(M_{k,j}\) and (b) \(V_k\).](image-url)
evaluations are important for a feasible VQS. This sub-
section presents the VQS architecture for single-qubit, two-qubit, and few-qubit gates. These gates can be implemented as a sum of easily-implementable unitaries (composed of single-qubit, two-qubit, and few-qubit operators).

We evaluate Eqs. (10) and (11), and decomposes
\[ \tilde{\psi}(t) \equiv \sum_{i,k=0}^{2^n-1} (L(t))_{i,k} |i \rangle |k \rangle, \]
where \( \tilde{\psi}(t) \) is the time-evolved quantum state |\tilde{\psi}(t)\rangle.

We obtain the time-evolved probability distribution \( \mathrm{Prob}[X(t) = x_j] \) for each qubit. Equation (13) is the essence of our proposal to simulate VQS-based SDE simulation: specifically, the VQS algorithm applied to Eq. (13) obtains the time-evolved probability distribution as the quantum state |\tilde{\psi}(t)\rangle. Hereafter, when the distinction is clear in context, we denote the operator \( \hat{L}(t) \) by \( L(t) \) as in Eq. (13).

C. Construction of \( L(t) \)

As explained in the previous section, in the VQS, we evaluate Eqs. (10) and (11), and decomposes \( L(t) \) into a sum of easily-implmentable unitaries (composed of single-qubit, two-qubit, and few-qubit gates). These evaluations are important for a feasible VQS. This subsection discusses the explicit decomposition of \( L(t) \) given by Eq. (13).

To express the operator \( L(t) \) in Eq. (13), we define operators
\[ V_+(n) \equiv \sum_{i=0}^{2^n-2} |i+1 \rangle |i \rangle, \quad V_-(n) \equiv \sum_{i=1}^{2^n-1} |i-1 \rangle |i \rangle. \]

These operators can be constructed from the \( n \)-qubit cyclic increment/decrement operator
\[ \text{CycInc}(n) = \sum_{i=0}^{2^n-1} |i+1 \rangle |i \rangle, \quad \text{CycDec}(n) = \sum_{i=0}^{2^n-1} |i-1 \rangle |i \rangle, \]
where \( |-1 \rangle, |2^n \rangle \) are identified with \( |2^n - 1 \rangle, |0 \rangle \), respectively. These gates are implemented as a product of \( O(n) \) Toffoli, CNOT, and X gates with \( O(n) \) ancilla qubits. \( V_+(n) \) is constructed from \( \text{CycInc}(n) \) and \( \text{CycDec}(n) \) and an \( n \)-qubit-control Z gate \( C^n Z \equiv \sum_{i=0}^{2^n-2} |i \rangle |i - 2^n - 1 \rangle |2^n - 1 \rangle \) which can be implemented as a product of \( O(n^2) \) Toffoli, CNOT, and single qubit gates. Using \( \frac{1}{2} (C^n Z + I^{\otimes n}) = \sum_{i=0}^{2^n-2} |i \rangle |i \rangle \), we can show that
\[ V_+(n) = \text{CycInc}(n) \cdot \frac{1}{2} (C^n Z + I^{\otimes n}), \]
\[ V_-(n) = \frac{1}{2} (C^n Z + I^{\otimes n}) \cdot \text{CycDec}(n), \]
meaning that \( V_+(n) \) can be decomposed into a sum of two unitaries composed of \( O(n^2) \) few-qubit gates. Finally, we define the operator \( D(n) \) by
\[ D(n) = \sum_{i=0}^{2^n-1} i |i \rangle |i \rangle = \frac{2^n - 1}{2} I^{\otimes n} - \sum_{i=1}^{n} 2^{n-i-1} Z_i, \]
where \( Z_i \) is a Z gate acting on the \( i \)-th qubit. Therefore, \( D(n) \) is a sum of \( O(n) \) unitaries composed of a single-qubit gate. It follows that
\[ V_+(n)(D(n))^m = \sum_{i=0}^{2^n-2} i^m |i + 1 \rangle |i \rangle, \]
\[ V_-(n)(D(n))^m = \sum_{i=1}^{2^n-1} i^m |i - 1 \rangle |i \rangle. \]

Let us recall
\[ L(t) = \sum_{i=0}^{2^n-2} \frac{1}{2} \left( \frac{\sigma^2(x_i,t)}{\Delta x^2} + \frac{\mu(x_i,t)}{\Delta x} \right) |i+1 \rangle |i \rangle + \sum_{i=1}^{2^n-1} \frac{1}{2} \left( \frac{\sigma^2(x_i,t)}{\Delta x^2} - \frac{\mu(x_i,t)}{\Delta x} \right) |i-1 \rangle |i \rangle - \sum_{i=0}^{2^n-1} \sigma^2(x_i,t) |i \rangle |i \rangle \].

Expanding \( \sigma^2(x_i,t) \) and \( \mu(x_i,t) \) as
\[ \sigma^2(x_i,t) = \sum_{m=0}^{m_a} a_{\sigma,m}(t)x_i^m, \quad \mu(x_i,t) = \sum_{m=0}^{m_a} a_{\mu,m}(t)x_i^m, \]
(27)
we can decompose \( L(t) \) as follows:

\[
L(t) = \sum_{m=0}^{m_n} a_{\sigma,m}(t) (\Delta x)^{m-2} \left( \frac{V_+(n) + V_-(n)}{2} - I \right) (D(n))^m \\
+ \sum_{m=0}^{m_n} a_{\mu,m}(t) (\Delta x)^{m-1} \left( \frac{V_+(n) - V_-(n)}{2} - I \right) (D(n))^m.
\]

\( V_+(n)(D(n))^m, \ V_-(n)(D(n))^m \) and \( (D(n))^m \) are composed of the sum of \( O(n^m) \) unitaries, each composed of \( O(n^2) \) few-qubit gates. In typical SDEs, the orders \( m_\sigma, m_\mu \) can be set to small values. For example, geometric Brownian motion case, \( m = 1 \) (see Sec. VI). Therefore, the \( L(t) \) decomposition realizes a feasible VQE (Eq. 15).

IV. CALCULATION OF EXPECTATION VALUES

In the previous section, we propose a method to simulate the SDE by calculating the dynamics of the probability distribution of a random variable \( X(t) \) using VQS. However, in many cases, the goal of the SDE simulation is not the probability distribution of \( X(t) \), but the expectation value \( E[f(X(t))] \) of \( X(t) \) for some function \( f \). In this section, we introduce a means of calculating this expectation value.

A. Problem Setting

Given a function \( f(x) : \mathbb{R} \to \mathbb{R} \), we try to calculate the expectation value \( E[f(X(T))] \) of the SDE (11) at time \( t = T \). The expectation value can be explicitly written as

\[
E[f(X(T))] = \sum_{i=0}^{2^n-1} f(x_i) \text{Prob}[X(T) = x_i]. \tag{28}
\]

Here, we assume that \( f(x) \) in the interval \([a_k, a_{k+1}] \subseteq \{[0, a_1], [a_1, a_2], \ldots, [a_d-1, a_{d-1}] \} \) is well approximated by \( L \)th order polynomials \( f_k(x) = \sum_{m=0}^{L} a_m^{(k)} x^m \). The additional error from this piecewise polynomial approximation is evaluated in Appendix C. As \( x \) is finite, the range of \( f \) is also finite. Thus, by shifting the function \( f \) by a constant, we can ensure that the range of \( f \) is positive and that the expectation value is also positive, i.e. \( E[f(X(T))] \geq 0 \). In most situations (such as pricing of European call options as we see in Sec. V.C) the number of intervals \( d \) does not scale with the number of qubits \( n \).

B. General formula for calculating expectation values

We now compute the expectation value \( \langle \tilde{\psi}(t)|S_f|\tilde{\psi}(t)\rangle \) using the quantum state \( |\tilde{\psi}(t)\rangle \) (Eq. (12)). First, we consider a non-unitary operator satisfying

\[
S_f |0\rangle = \sum_{i=0}^{2^n-1} f(x_i) |i\rangle, \tag{29}
\]

and decompose \( S_f \) into a sum of easily-implementable unitaries as \( S_f = \sum_i \xi_i Q_i \) with complex coefficients \( \xi_i \). It follows that

\[
\langle \tilde{\psi}(t)| \left( S_f |0\rangle \langle 0 | S_f^\dagger \right) |\tilde{\psi}(t)\rangle = \langle E[f(X(T))] \rangle^2. \tag{30}
\]

As \(|0\rangle \langle 0 | = I - C^n Z \cdot X^\otimes \) is also a sum of easily-implementable unitaries as explained in the previous subsection, the Hermitian observable \( S_f |0\rangle \langle 0 | S_f^\dagger \) is decomposed as

\[
S_f |0\rangle \langle 0 | S_f^\dagger = \sum_{i,i'} \xi_{i,i'} \left( Q_i Q_i^\dagger \right) \left( Q_{i'} C^n Z \cdot X^\otimes Q_{i'}^\dagger \right), \tag{31}
\]

which is again a sum of unitaries. With this decomposition, the left-hand side of Eq. (30) is computed by evaluating \( \langle \tilde{\psi}(t)|Q_i Q_i^\dagger |\tilde{\psi}(t)\rangle \) and \( \langle \tilde{\psi}(t)|Q_{i'} C^n Z \cdot X^\otimes Q_{i'}^\dagger |\tilde{\psi}(t)\rangle \). Because we set \( \langle E[f(X(T))] \rangle \geq 0 \), the left-hand side of Eq. (30) will determine the expectation value.

There are two options to evaluate the quantities \( \langle \tilde{\psi}(t)|Q_i Q_i^\dagger |\tilde{\psi}(t)\rangle \) and \( \langle \tilde{\psi}(t)|Q_{i'} C^n Z \cdot X^\otimes Q_{i'}^\dagger |\tilde{\psi}(t)\rangle \). The first one is to use the Hadamard test depicted in Fig. 3. The second one is to use quantum phase estimation \([49, 50]\). The former one requires shallower quantum circuits but is inefficient in terms of the number of measurements to determine the quantities with fixed precision. The detailed computational complexity of these methods is given in Sec. VII and Appendix A.

Next, we explain the construction of the operator \( S_f \) in Eq. (29) and its decomposition. We first define an operator

\[
S_{\chi_{[0,a]}} |0\rangle = \sum_{i=0}^{2^n-1} \chi_{[0,a]}(x_i) |i\rangle = \sum_{x_i \in [0,a]} |i\rangle, \tag{32}
\]

where \( \chi_{[0,a]}(x) \) is the indicator function valued as 1 for \( x \in [0, a] \) and 0 for otherwise. Using the binary expansion of \( a/\Delta x \), we can obtain the decomposition of \( S_{\chi_{[0,a]}} \) Figure 3. Quantum circuit for evaluating the real part of an expectation value \( \Re(\langle \tilde{\psi}(t)|U |\tilde{\psi}(t)\rangle) \) of a unitary operator \( U = Q_i Q_i^\dagger, Q_i C^n Z \cdot X^\otimes Q_i^\dagger \). The imaginary part of the expectation value \( \Im(\langle \tilde{\psi}(t)|U |\tilde{\psi}(t)\rangle) \) is evaluated by the circuit with an \( S^\dagger \) gate inserted to the left of the second \( H \) gate.
hence the decomposition of $S_f$. As $a \in [0, x_{\max}]$, there exists $k_a \in \mathbb{N}$ such that $\Delta x 2^{k_a - 1} \leq a < \Delta x 2^{k_a}$, $0 < k_a \leq n$. The binary expansion of $a/\Delta x$ is given by $a/\Delta x = \sum_{j=0}^{k_a-1} s_j 2^j$, $s_j \in \{0, 1\}$. We define the list of $l$ as $l_1, l_2, \ldots, l_B (= k_a - 1)$ satisfying $s_l = 1$ in ascending order, and also define an interval

$$\chi_l^a = \left[ 2^{lB} + \sum_{j=0}^{l-1} s_j 2^j + 1, 2^{lB} + \sum_{j=0}^{l} s_j 2^j \right] (33)$$

for $l \in \{l_1, l_2, \ldots, l_B\}$. Using $\chi_l^a$, we devise $[0, a/\Delta x]$ into disjoint intervals as follows:

$$[0, a/\Delta x] = \bigcup_{l} (\chi_l^a) \cup \cdots \cup \chi_l^B. \quad (34)$$

The indicator operator $S_{ \chi_l^a }$ is obtained by summing the indicator operators on each interval. In binary expansion, the $k_a$th and the $l$th bit of $i \in \chi_l^a$ are 1, and the bit below $l$ is either 0 or 1. Accordingly, $X$ should act on the bit taking 1, and $H$ should act on the bit taking either of $(0,1)$. The indicator operator $S_{ \chi_l^a }$ on $\chi_l^a$ is defined as follows:

$$S_{ \chi_l^a \mid 0 } = |0 \rangle \otimes |n - k_a - l - 1 \rangle \otimes \sum_{j=0}^{l} |s_n - k_a - j \rangle \otimes |j \rangle \otimes \sum_{j=0}^{l} |X \rangle$$

where

$$X_{sa} = \begin{cases} X & (s = 1) \\ I & (s = 0) \end{cases}. \quad (36)$$

In addition, we define

$$S_{ \chi_{[0,2^{k_a-1}-1]} \mid 0 } = 2^{(k_a-1)/2} I \otimes H^{\otimes n-k_a-1} \otimes |0 \rangle \otimes \sum_{j=0}^{l} |X \rangle$$

We can construct $S_{ \chi_0^a }$ by summing Eqs. (35) and (37) on each interval. $S_{ \chi_{a_k} }$ on interval $a_k \equiv [a_k, a_{k+1}]$ is

$$S_{ \chi_{a_k} } = S_{ \chi_{[a_k,a_{k+1}]-1} } - S_{ \chi_{[0,a_k]-1} } \quad (38)$$

which is a sum of at most $O(n)$ unitaries composed of $O(n)$ gates. Using $S_{ \chi_{a_k} }$, we obtain

$$S_f \mid 0 \rangle = \sum_{k=0}^{d-1} \sum_{x \in a_k} f(x) S_{ \chi_{a_k} } \mid 0 \rangle \quad (39)$$

and $S_f$ is constructed as

$$S_f = \sum_{k=0}^{d-1} \sum_{m=0}^{L} a_m^{(k)} (D(n))^m S_{ \chi_{a_k} } \quad (40)$$

In summary, evaluation of the expectation value is calculated by the following steps.

1. Divide the domain of the target function $[0, x_{\max}]$ into intervals $[a_k, a_{k+1}] \in \{[0, a_1], [a_1, a_2], \ldots, [a_{d-1}, x_{\max}]\}$.

2. Approximate the function in each interval $[a_k, a_{k+1}]$ by Eq. (40).

3. Decompose $S_f \mid 0 \rangle \otimes |0 \rangle S_f^\dagger$ into a sum of unitary terms and calculate each term using the circuits in Fig. 4.

As $S_{ \chi_{[a_k,a_{k+1}]}} \mid (D(n))^m \mid 0 \rangle \otimes |0 \rangle$ is the sum of $O(n)$, $O(n^m)$ and $O(1)$ unitaries composed of $O(n)$, $O(1)$ and $O(n^2)$ gates, respectively, $S_f \mid 0 \rangle \otimes |0 \rangle S_f^\dagger$ is the sum of $O(d^2 n^{2L+2})$ unitaries and each $Q_i$ is composed of at most $O(n^4)$ gates.

When the target function $f$ on each interval is written by a low-degree polynomial (i.e., $L$ is small), especially by a linear function (as in the pricing of European call options shown below), our algorithm can efficiently calculate the expectation value because the number of unitaries $O(d^2 n^{2L+2})$ gets not so large. When the function $f$ is approximated by the polynomial, we can estimate the error of the expectation value stemming from that approximation. If we want to suppress the error below $\epsilon$, the number of unitaries becomes $O(x_{\max}^2 \epsilon^{-2}/22d^2 n^{2L+2})$ (the derivation is presented in Appendix C). Note that as $L$ is increased, $\epsilon^{-22d^2 n^{2L+2}}$ becomes smaller while $n^{2L+2}$ becomes larger. The number of unitaries, therefore, is not monotonic with respect to $L$, and there may be an optimal $L$ for the desired accuracy. We note that evaluation of expectation values of those unitaries can be performed completely in parallel by independent quantum devices.

C. Pricing of The European Call Option

As a concrete example, we present the pricing of a European call option with the Black-Scholes (BS) model, which is one of the simplest financial derivatives. The holder of a European call option is entitled to buy the asset at a predetermined strike price at maturity. The price of a European call option with strike price $K \geq 0$, interest rate $r \geq 0$, and maturity $T \geq 0$ is defined by the conditional probability

$$e^{-rT} E_Q [ \max(X_T - K, 0) | X_0 = x_0 ] \quad (41)$$

Here, $E_Q$ denotes the expectation value under the risk-neutral probability measure. Stochastic processes are assumed to follow geometric Brownian motion in the BS model, but are described by more complex mechanisms in other models. Even in these models, the expression Eq. (41) of the price of the European call option is the same with the present case.

Setting the probability distribution of $X_T$ conditioned by $X_0 = x_0$ as \{Prob\ $[X_T = x_i | X_0 = x_0] \}^{2^n - 1}$, the expectation value is

$$e^{-rT} E_Q [ \max(X_T - K, 0) | X_0 = x_0 ]$$

$$= e^{-rT} \sum_{i=0}^{2^n - 1} \text{Prob} [X_T = x_i | X_0 = x_0] \max(x_i - K, 0). \quad (42)$$
For simplicity, we assume $\Delta x = 1$ and $K = 2^k < 2^n - 1, k \in \mathbb{N}$. We thus obtain

$$S_{\text{max}(i-K, n)} = (D(n) - KT)S_{(K, x^n - 1)}.$$  

(43)

In this case, there are only two intervals $[0, K - 1]$ and $[K, 2^{n-1}]$, and the polynomial in each interval is of first-order degree at most. Therefore, we can calculate the price of the European call option by Eq. (50).

V. POSSIBLE ADVANTAGES OF OUR METHOD

In this section, we discuss the advantages of our method compared to previous studies, as well as the possible quantum advantages.

In general, the SDEs addressed in this paper can be transformed into a partial differential equation (PDE) of the function $e_f(x, t)$, where $e_f(x, t)$ gives the expectation value $E[f(X(T - t))]|X(0) = x|$, by Feynman-Kac formula \[1].\] In fact, the authors of \[22\] performed a variational quantum computation of a PDE of this function. We point out two advantages of our method compared with previous studies, as well as the possibility of quantum computation of a PDE of this function.

The authors of \[23\] embedded the probability distribution using a quantum generative adversarial network, which requires only $O(\text{Poly}(n))$ gates. The probability distribution function can also be approximated by a 4th-order piecewise polynomial, which can be embedded with $O(\ln n^2)$ gates even in quantum arithmetic \[43\]. However, both methods require prior knowledge of the probability distribution to be embedded. In contrast, our method does not require prior knowledge of the embedding probability distribution since our method simulates the time evolution of a given SDE.

We now compare the computational cost to calculate expectation values with previous studies. In \[23\] and \[24\], by employing QAE, the expectation value (Eq. (28)) was calculated by using an oracle that is complex quantum gate reflecting the functional form of $f$ for $O(1/\epsilon)$ times, where $\epsilon$ is the precision for the expectation values. The classical Monte Carlo method requires $O(1/\epsilon^2)$ sampling for precision $\epsilon$, so their methods provide a second-order acceleration. On the other hand, our method measures the expectation value of each term of Eq. (31) using the Hadamard test \([10, 11]\) or the quantum phase estimation (QPE) \([40, 41]\). As shown in Appendix \[A\] the total number of measurements to obtain the expectation value with precision $\epsilon$ is $O(1/\epsilon^2)$ for the Hadamard test and $O(\log(1/\epsilon^2))$ for QPE, where $\gamma$ is some factor. We note that the depth of the circuit is $O(1/\epsilon)$ in QPE, which is in the same order as the QAE whereas our method requires not an complicated oracle but a relatively-small unitary. Hence, when the factor $\gamma$ is not too small, our method combined with QPE can also exhibit quantum advantage for the evaluation of the expectation values. The factor $\gamma$ depends on the parameters of the polynomial approximation $(a_k^{(m)}, d, L)$, the domain of the approximated function $x_{\text{max}}$, and the probability distribution \{$\text{Prob}[X = x_i]|_{i=0}^{2^n-1}$\}. The detailed evaluation of $\gamma$ is described in Appendix \[A\].

VI. NUMERICAL RESULTS

In this section, our algorithm is applied to two stochastic processes, namely, geometric Brownian motion and an Ornstein–Uhlenbeck process, which are commonly assumed in financial engineering problems. Geometric Brownian motion simply models the fluctuations of asset prices, and the Ornstein–Uhlenbeck process is a popular model of interest rates.

A. Models

Geometric Brownian motion is equivalent to setting $\mu(X(t), t) = rX(t), \sigma(X(t), t) = \sigma X(t)$ in Eq. (1), where $r$ and $\sigma$ are positive constants.

The Ornstein–Uhlenbeck process is equivalent to setting $\mu(X(t), t) = -\eta(X(t) - r), \sigma(X(t), t) = \sigma$ in Eq. (1), where $\eta, r$ and $\sigma > 0$ are constants.

The ansatz circuit is identical for both models and shown in Fig. \[4\]. As the amplitudes of the quantum state must be real, the ansatz contains only CNOT and RY gates. This depth-$k$ circuit repeats the entangle blocks composed of CNOTs and RY gates $k$ times. The parameters of geometric Brownian motion were $r = 0.1, \sigma = 0.2, \Delta x = 1$, and $t \in [0, 4]$ and those of the Ornstein–Uhlenbeck process were $r = 0.5, \sigma = 0.01, \Delta x = 1$, and $t \in [0, 4]$. We simulate the quantum circuits without noise using numpy \[44\] and jax \[45\].
apply to the Fokker-Plank equation, which also gives the
time-evolution of the probability distributions of SDE solu-
tions.

Because the embedding methods of the probability dis-
tribution differ in the proposed method and the con-
ventional quantum algorithm, we proposed another method
for computing the expectation value. We approximated
the functions to calculate expectation values by piece-
wise polynomials and constructed operators correspond-
ing to the polynomial in each interval. The operators
were constructed as sums of unitary operators, which are
composed of easily-implementable gates. The expecta-
tion value was then computed using the sum of unitary
operators. Our algorithm was validated in classical simu-
lations of geometric Brownian motion and the Ornstein-
Uhlenbeck process. Both processes were well simulated
by the algorithm. Our algorithm is expected to efficiently
simulate other stochastic processes, provided that
\( L(t) \) can be written as a polynomial linear combination of uni-
tary matrices.

Let us summarize the computational cost of our
method presented in this work. Our method consists of
two parts; one is to perform VQS to simulate the SDE,
and the other is to calculate the expectation value of
the SDE solution. In the part of running VQS, we decompose
matrix \( L(t) \) in Eq. (1) into a sum of \( O(n^{m_{\text{max}}}) \) differ-
ent unitaries composed of \( O(n^2) \) few-qubit gates, where
\( m_{\text{max}} \) is the largest order of the polynomial expansion of
\( \mu, \sigma \) in Eq. (27). At each time step of VQS, the vector \( V_k \)
in Eq. (11) is evaluated as a sum of \( O(n^{m_{\text{max}}}) \) measure-
ment results of the circuits depicted in Fig. 2. As \( m_{\text{max}} \) is
typically finite and small (\( \sim 1, 2 \)) in most practical appli-
cations, the computational cost (i.e., the number of gates
in quantum circuits, the number of different circuits to
run) of the simulation of SDE is \( O(\text{Poly}(n)) \). In con-
trast, QLSA \( [12, 31, 32] \) requires much deeper and more
complex quantum circuits and a large number of ancilla
qubits because it uses the Hamiltonian simulation and
the quantum Fourier transform. This is an advantage
of our method leveraging the variational quantum algo-
rithm.

In the part of the expectation value evaluation of
the SDE solution, we evaluate it by running different
\( O(d^2n^{2L+2}) \) quantum circuits, where \( d \) and \( L \) are
the number of intervals and the order of the piecewise poly-
nomial approximation of the function \( f \) in Eq. (25),
respectively. Each circuit is constructed to compute an
expectation value \( \langle \psi | U | \psi \rangle \) of a unitary \( U \) that contains
\( O(n^4) \) quantum gates. When we adopt the Hadamard
test (Fig. 3) as such a quantum circuit, the number of
measurements to suppress statistical error of the expecta-
tion value below \( \epsilon \) is \( O(1/\gamma \epsilon^2) \), where \( \gamma \) is a factor de-
defined in Appendix A. This \( O(1/\epsilon^2) \) scaling is the same
as the classical Monte Carlo method to compute the ex-
pectation values from the probability distribution of the
SDE solution. When we choose the QPE-type circuit to
evaluate \( \langle \psi | U | \psi \rangle \), the number of measurements becomes
\( O(\log(1/\gamma \epsilon)) \) while the depth of the circuit in terms of

![Diagram](https://via.placeholder.com/150)

**FIG. 4.** In a depth-\( k \) circuit, CNOT and RY gates (enclosed by dashed lines) are repeated \( k \)-times. The circuit has \( 4(k+1) \) parameters.

set the number of qubits \( n = 4 \) and the number of repet-
tions of entangle blocks \( k = 2, 3 \).

**B. Results**

Panels (a) and (b) of Fig. 5 present the numerical sim-
ulations of geometric Brownian motion and the Ornstein-
Uhlenbeck process, respectively. In comparison, we also
provide a probability density function (PDF) for the solu-
tion of the SDE equation obtained by solving the Fokker-
Planck equation [46] analytically. We can see that our
algorithm is expected to efficiently simulate other stochastic processes, provided that
\( L(t) \) can be written as a polynomial linear combination of uni-
tary matrices.

Let us summarize the computational cost of our
method presented in this work. Our method consists of
two parts; one is to perform VQS to simulate the SDE,
and the other is to calculate the expectation value of
the SDE solution. In the part of running VQS, we decompose
matrix \( L(t) \) in Eq. (1) into a sum of \( O(n^{m_{\text{max}}}) \) differ-
ent unitaries composed of \( O(n^2) \) few-qubit gates, where
\( m_{\text{max}} \) is the largest order of the polynomial expansion of
\( \mu, \sigma \) in Eq. (27). At each time step of VQS, the vector \( V_k \)
in Eq. (11) is evaluated as a sum of \( O(n^{m_{\text{max}}}) \) measure-
ment results of the circuits depicted in Fig. 2. As \( m_{\text{max}} \) is
typically finite and small (\( \sim 1, 2 \)) in most practical appli-
cations, the computational cost (i.e., the number of gates
in quantum circuits, the number of different circuits to
run) of the simulation of SDE is \( O(\text{Poly}(n)) \). In con-
trast, QLSA \( [12, 31, 32] \) requires much deeper and more
complex quantum circuits and a large number of ancilla
qubits because it uses the Hamiltonian simulation and
the quantum Fourier transform. This is an advantage
of our method leveraging the variational quantum algo-
rithm.

In the part of the expectation value evaluation of
the SDE solution, we evaluate it by running different
\( O(d^2n^{2L+2}) \) quantum circuits, where \( d \) and \( L \) are
the number of intervals and the order of the piecewise poly-
nomial approximation of the function \( f \) in Eq. (25),
respectively. Each circuit is constructed to compute an
expectation value \( \langle \psi | U | \psi \rangle \) of a unitary \( U \) that contains
\( O(n^4) \) quantum gates. When we adopt the Hadamard
test (Fig. 3) as such a quantum circuit, the number of
measurements to suppress statistical error of the expecta-
tion value below \( \epsilon \) is \( O(1/\gamma \epsilon^2) \), where \( \gamma \) is a factor de-
defined in Appendix A. This \( O(1/\epsilon^2) \) scaling is the same
as the classical Monte Carlo method to compute the ex-
pectation values from the probability distribution of the
SDE solution. When we choose the QPE-type circuit to
evaluate \( \langle \psi | U | \psi \rangle \), the number of measurements becomes
\( O(\log(1/\gamma \epsilon)) \) while the depth of the circuit in terms of
FIG. 5. (a), (b): Exact solutions of the SDE (solid lines) and numerical simulation of our algorithm (circles); (c), (d): time dependence of the means and (e), (f) variances of the encoded probability distributions. Dashed lines show the numerical solutions of VQS with a $k = 2, 3$ depth ansatz. Dotted lines show the numerical Runge-Kutta solutions of the linear differential equation (Eq. (16)). Solid lines show the exact solutions of the SDE.

(a) Dynamics of geometric Brownian process
(b) Dynamics of Ornstein-Uhlenbeck process
(c) Time dependence of mean in geometric Brownian motion
(d) Time dependence of mean in Ornstein-Uhlenbeck process
(e) Time dependence of variance in geometric Brownian motion
(f) Time dependence of variance in Ornstein-Uhlenbeck process

Geometric Brownian motion with parameters $r = 0.1, \sigma = 0.2$. Ornstein-Uhlenbeck Process with parameters $r = 7, \sigma = 0.5, \eta = 0.01$. 
$U$ is $O(1/\gamma\epsilon)$. This situation can provide a quantum advantage for computing the expectation value of the SDE solution. The error from the piecewise polynomial approximation of $f$ can be made small by increasing $d$ or $L$, which is detailed in Appendix C.

This study focused on computational finance because financial engineering is among the most popular applications of stochastic processes. Pricing of derivatives, and many other problems in financial engineering, satisfy the conditions of the proposed method. However, as the stochastic processes themselves are quite general, the proposed method is expected to contribute to solving problems in various fields.

ACKNOWLEDGMENTS

The authors thank Keisuke Fujii for valuable discussions on variational quantum algorithms.

[1] S. Shreve, *Stochastic Calculus for Finance II* (Springer, 2004).
[2] N. G. V. Kampen, *Stochastic Processes in Physics and Chemistry* (North Holland, 2011).
[3] D. J. Wilkinson, *Nat. Rev. Genet.* 10, 122 (2009).
[4] N. Metropolis and S. Ulam, *J. Am. Stat. Assoc.* 44, 335 (1949).
[5] F. Black and M. Scholes, *J. Polit. Econ.* 81, 637 (1973).
[6] F. A. Longstaff and E. S. Schwartz, *Rev. of Financ. Stud.* 14, 113 (2001).
[7] F. Black and M. Scholes, *J. Polit. Econ.* 81, 637 (1973).
[8] F. A. Longstaff and E. S. Schwartz, *Rev. of Financ. Stud.* 14, 113 (2001).
[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, T. Baker, J. Barends, R. Bialczak, R. Blakestad, S. Boixo, F. Brem, J. Chen, D. Chiaro, A. Chong, M. Christiansen-Franskin, C. Cobanler, M. Cordaro, A. Del compatibility, B. Dunlop, P. Frunzio, M. Gammon, A. Ganzhorn, M. Harris, C. Hong, Y. Huang, T. Jiang, D. Kechedzhi, K. Keich, M. Kelly, J. Kelly, M. Kuwen, A. Landgarten, M. Lenske, T. Lamb, I. Lim, A. Lobas, A. Lucchi, S. Mariantoni, J. Chow, J. Gambetta, B. Geraci, S. Gray, A. Hare, A. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. R. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. N. Korotkov, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, M. Y. Niu, E. Os-
The error in the linear combination of expectation values is determined as

\[ \sum_{i=1}^{N_U} \beta_i \tilde{u}_i - \langle \psi \mid \sum_{i=1}^{N_U} \beta_i U_i \mid \psi \rangle = \sum_{i=1}^{N_U} \beta_i (\tilde{u}_i - \langle \psi \mid U_i \mid \psi \rangle) \leq \sum_{i=1}^{N_U} |\beta_i| \langle (\tilde{u}_i - \langle \psi \mid U_i \mid \psi \rangle) \rangle \leq \epsilon' \sum_{i=1}^{N_U} |\beta_i|. \tag{A4} \]

Denoting the estimation of \( E \) as \( \tilde{E} \), we have

\[ |\tilde{E} - E| \leq \sum_{j=0}^{2^n-1} p_j^2 \epsilon' \sum_{i=1}^{N_U} |\beta_i| E + E \sim \sum_{j=0}^{2^n-1} p_j^2 \epsilon' \sum_{i=1}^{N_U} |\beta_i| \frac{2E}{2E}. \tag{A5} \]

To upper limit the error \( \epsilon \) in \( E \), the error \( \epsilon' \) must satisfy following condition:

\[ \epsilon' \leq \frac{2\epsilon E}{\sum_{j=0}^{2^n-1} p_j^2 \sum_{i=1}^{N_U} |\beta_i|} = \gamma \epsilon, \tag{A6} \]

where \( \gamma \equiv \left( \frac{2E}{\sum_{j=0}^{2^n-1} p_j^2 \sum_{i=1}^{N_U} |\beta_i|} \right). \)

The Hadamard test (Fig. 3) requires \( O(1/\epsilon'^2) \) measurements to limit the error in the expectation value to \( \epsilon' \), and the depth of the quantum circuit is \( O(1) \) in terms of the unitary \( U = Q_1 Q_1^\dagger, Q_{C^n} Z \otimes Q_1 \) except for the circuit to prepare the quantum state. On the other hand, the number of measurements to achieve the same accuracy with QPE is \( O(\log(1/\epsilon')) \), but the depth of the quantum circuit in terms of the unitary \( U \) is \( O(1/\epsilon') \) \[ \left[ 40, 41 \right] \). The total number of measurements is equal to the number of measurements for each term multiplied by \( N_U \). Note that \( N_U = O(d^2 n^{2L+2}) \) with \( L \)th-order piecewise polynomial approximation of the function \( f \) with \( d \) intervals (see Sec. IVB). The total number of measurements required to calculate the expectation value is \( O(d^2 n^{2L+2}/\epsilon^2) \) by Hadamard test and \( O(d^2 n^{2L+2} \log(1/\gamma/\epsilon')) \) by QPE. QPE requires extra \( U \) gates, the number of which is \( O(1/\gamma) \).

Finally, we provide the estimation of the value of \( \gamma \) as follows. The factor \( \sum_{j=0}^{2^n-1} p_j^2 \) satisfies \( \sum_{j=0}^{2^n-1} p_j^2 \leq 1 \) and then \( \gamma \geq 2E/\sum_{i=1}^{N_U} |\beta_i| \). To evaluate \( \sum_{i=1}^{N_U} |\beta_i| \), we use Eq. (11) and obtain

\[ \sum_{i=1}^{N_U} |\beta_i| = 2 \sum_{l,t} |\xi_l \xi_t^*| = 2 \sum_{l,t} |\xi_l| |\xi_t|. \tag{A7} \]

We estimate the upper limit of sum of absolute values of coefficients in Eq. (11) to evaluate \( \sum_{i} |\xi_i| \). From Eq. (24), the absolute values of coefficients of \( D(n) \) is at most \( O(2^m) = O(x_{\text{max}}^m) \). As \( S_\chi \) is a linear combination of

---

[38] S. Lloyd, G. de Palma, C. Gokler, B. Kiani, Z. W. Liu, F. Tennie, T. Palmer, and M. Marvian, arXiv, 1 (2020).

[39] X. Li, G. Yang, C. M. Torres, D. Zheng, and K. L. Wang, Int. J. Mod. Phys. B 28, 1350191 (2014).

[40] E. Knill, G. Ortiz, and R. D. Somma, Phys. Rev. A 75, 012328 (2007).

[41] D. Wang, O. Higgott, and S. Brierley, Phys. Rev. Lett. 122, 140504 (2019) [arXiv:1802.00171].

[42] L. Grover and T. Rudolph, (2002).

[43] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, L. Grover and T. Rudolph, (2002).

[44] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, L. Grover and T. Rudolph, (2002).

[45] H. Risken, The Fokker-Planck Equation (Springer, Berlin, Heidelberg, 1996).
respectively given by adjacent values, respectively. The transition probability to nodes where two variables \((X_1(t), X_1(t+1))\) occur to nodes with identical variable values, to nodes where 1\((X_1(t), X_1(t+1))\) is larger than the ratio of \(\rho_{kl}\). The transition probabilities can be determined identically to those of the one-variable SDE. The SDE at \((X_1(t), \ldots, X_D(t))\) is discretized as

\[
X_d(t+1) - X_d(t) = \mu_d(X_d(t), t)\Delta t + \sigma_d(X_d(t), t)\sqrt{\Delta t}z_d,
\]

where \(\{z_d\}_{d=1}^D\) is sampled from the multi-variable Gaussian distribution, \(\mathbb{E}[z_d] = 0, \text{Var}[z_d] = 1, \text{Corr}[z_k, z_l] = \rho_{kl}\). The first and second first and second conditional moments satisfy

\[
\text{E}[X_d(t_{j+1}) - X_d(t_j) | X_d(t_j) = x] = \mu_d(x, t_j)\Delta t,
\]

\[
\text{Var}[X_d(t_{j+1}) - X_d(t_j) | X_d(t_j) = x] = \sigma_d^2(x, t_j)\Delta t
\]

for \(d = 1, \ldots, D\) and the covariance of the variables satisfies

\[
\text{Cov}[X_k(t_{j+1}) - X_k(t_j), X_l(t_{j+1}) - X_l(t_j)] = \sigma_k(x, t_j)\sigma_l(y, t_j)\rho_{kl}\Delta t
\]

for \(1 \leq k < l \leq D\). The corresponding quantities in the tree model are

\[
\text{E}[X_d(t_{j+1}) - X_d(t_j) | X_d(t_j) = x] = \left(p_{u}^{(d)} - p_{d}^{(d)} + \sum_{k=1}^{l} \left(p_{u}^{(k,d)} - p_{u}^{(k,d)} + p_{u}^{(k,d)} - p_{d}^{(k,d)}\right)\Delta x^{(d)} + \sum_{l=1}^{l} \left(p_{u}^{(d,l)} - p_{u}^{(d,l)} + p_{u}^{(d,l)} - p_{d}^{(d,l)}\right)\Delta x^{(l)}\right)^2
\]

for \(d = 1, \ldots, D\) and

\[
\text{Cov}[X_k(t_{j+1}) - X_k(t_j), X_l(t_{j+1}) - X_l(t_j)] = \left(p_{u}^{(k,l)} - p_{u}^{(k,l)} + p_{u}^{(k,l)} - p_{u}^{(k,l)}\right)\Delta x^{(k)}\Delta x^{(l)}
\]

As is the same for the case of a single variable we set the transition amplitudes by equating Eqs. (B3), (B4), (B5) with (A6), (A7), (A8). If the solutions of \(p_{u}^{(k)}, p_{d}^{(k)}\) are proportional to \(\Delta t\), the linear differential equation can be derived by taking the limit of \(\Delta t \to 0\) (as in the one-dimensional case Eq. (10)).
When $D > 1$, one should note the numbers of variables and conditional expressions. As the numbers of $p_m$, $p_{u,d}^{(k,l)}$, $p_{uu,ud,du,dd}^{(k,l)}$ are $1, 2D, 2D(D - 1)$, respectively, the number of independent variables is $2D^2$ under the normalized probability conditions. On the other hand, the number of equations of the mean, variance, and covariance are $D, D, D(D - 1)/2$, respectively, so the total number of equations is $D(D + 3)/2$. When $D > 1$, the number of variables exceeds the number of conditions, so an infinite number of transition probabilities satisfy the condition.

Here, we show there is indeed a solution of the transition amplitudes which admit taking limit $\Delta t \to 0$ and obtain the linear differential equation of the probability distributions of the SDE. Fixing $p_{u,d}^{(k,l)} = p_{d}^{(k,l)} = p_{uu}^{(k)} = 0$, the number of variables becomes $D(D + 3)/2$, which is slightly asymmetric (because we consider only $p_{uu}^{(k)}$ to be nonzero), but agrees with the number of conditional expressions. In this case, the transition probabilities are

\[
p_{uu}^{(k,l)} = \frac{\sigma_k \sigma_l \rho_{kl}}{\Delta x^{(k)} \Delta x^{(l)}} \Delta t,
\]

\[
p_{d}^{(k,l)} = \frac{1}{2} \left( \frac{\sigma_k^2}{\Delta x^{(k)}} + \frac{\mu_d}{\Delta x^{(k)}} \right) - \sum_{k \neq d} \frac{\sigma_k \sigma_l \rho_{kl}}{\Delta x^{(k)} \Delta x^{(l)}},
\]

\[
p_{d}^{(k,l)} = \frac{1}{2} \left( \frac{\sigma_k^2}{\Delta x^{(k)}} - \frac{\mu_d}{\Delta x^{(k)}} \right),
\]

\[
p_m = 1 - \sum_{d=1}^{D} \left( \frac{\sigma_d^2}{\Delta x^{(d)}} \right) - \sum_{k \neq d} \frac{\sigma_k \rho_{kl}}{\Delta x^{(k)} \Delta x^{(l)}} \Delta t
\]

\[= 1 - \sum_{d=1}^{D} \frac{\sigma_d^2}{\Delta x^{(d)}} \Delta t.
\]

Here, we omit the arguments of $\mu_d$ and $\sigma_d$ to simplify the notation.

2. Mapping to VQS and construction of $L(t)$

In the multivariate case, we can construct $L(t)$ as described in Sec. 111. For notational simplicity, we denote $|i_1, \ldots, i_D \rangle = |i_1^{\text{even}}, i_1^{\text{odd}}, \ldots, i_{D-1}^{\text{even}}, i_{D-1}^{\text{odd}}, i_D \rangle = |i^{\text{even}}, i_{D-1}^{\text{odd}} \rangle$, $|i^{\text{odd}}, i_{D-1}^{\text{even}} \rangle = |i^{\text{odd}}, i_{D-1}^{\text{even}} \rangle$, $|i^{\text{even}}, i_{D-1}^{\text{odd}} \rangle = |i^{\text{even}}, i_{D-1}^{\text{odd}} \rangle$. Using Eqs. (B9), (B10), (B11) and (B12), we obtain

\[
L(t) = \frac{1}{2} \sum_{d=1}^{D} \sum_{i_d=0}^{2n-2} \left[ \frac{\sigma_d^2}{\Delta x^{(d)}} \right] + \frac{\mu_d}{\Delta x^{(d)}}
\]

\[- \sum_{k \neq d} \frac{\sigma_k \sigma_l \rho_{kl}}{\Delta x^{(k)} \Delta x^{(l)}} \langle i | e_d \rangle \langle \tilde{i} | \]

\[+ \frac{1}{2} \sum_{d=1}^{D} \sum_{i_d=0}^{2n-1} \left( \frac{\sigma_d^2}{\Delta x^{(d)}} - \frac{\mu_d}{\Delta x^{(d)}} \right) \times \langle \tilde{i} | i - e_d \rangle \langle \tilde{i} | i | \]

\[+ \sum_{k \neq l} \sum_{i_k, i_l=0}^{2n-2} \frac{\sigma_k \sigma_l \rho_{kl}}{\Delta x^{(k)} \Delta x^{(l)}} \times \langle \tilde{i} | i + e_k + e_l \rangle \langle \tilde{i} | i \rangle
\]

\[= \sum_{i_k=1}^{D} \sum_{i_l=1}^{D} \sum_{i_{k-1}=1}^{D} \frac{m_k}{m_l} | i + e_k + e_l \rangle \langle i | .
\]
Thus, we can construct $S_f$ as

$$L(t) = \sum_{d=1}^{D} \sum_{m=0}^{m_d} \sum_{k=0}^{m_d} a_{\sigma,m,k} a_{\sigma,m,k} \left( \Delta x^{(d)} \right)^{m_k+m_l-2} \times \left( \frac{V^{(k)}(t) + V^{(k)}(0)}{2} - I \right) \left( D^{(d)}(n) \right)^{m_k+m_l}$$

$$+ \sum_{d=1}^{D} \sum_{m=0}^{m_d} \sum_{k=0}^{m_d} a_{\mu,m,k} a_{\mu,m,k} \left( \Delta x^{(d)} \right)^{m_k+m_l-1} \times \left( \frac{V^{(k)}(t) - V^{(k)}(0)}{2} \right) \left( D^{(d)}(n) \right)^{m_k+m_l}$$

$$+ \sum_{k,l,d=m=0}^{m_d} \sum_{k=0}^{m_d} a_{\sigma,m,k} a^{(l)}_{\sigma,m,k} \left( \Delta x^{(k)} \Delta x^{(l)} \right)^{-1} \times V^{(k)}(t) \left( D^{(d)}(n) \right)^{m_k} V^{(l)}(t) \left( D^{(d)}(n) \right)^{m_l}.$$  \hspace{1cm} (B22)

As $V^{(k)}(t) \left( D^{(d)}(n) \right)^{m_k} V^{(l)}(t) \left( D^{(d)}(n) \right)^{m_l}$ are the sums of $O(n^m)$ unitaries composed of $O(n^2)$ few-qubit gates, Eq. (B22) is feasible decomposition of $L(t)$.

3. Evaluating the expectation value

To perform computation of the expectation value, we construct a multivariate indicator operator. In the $D$ dimensional case, the domain of the function is $\prod_{i=1}^{D}[0,x^{(i)}_{\max}]$. In each dimension, we divide $[0,x^{(i)}_{\max}]$ into $d$ intervals $\{a^{(i)}_0,a^{(i)}_1,\ldots,a^{(i)}_{d-1},x^{(i)}_{\max}\}$ and obtain $d^D$ regions $I(k_i) = \prod_{i=1}^{D}[a^{(i)}_{k_i},a^{(i)}_{k_i+1}]$. The indicator operator on $I(k_i)$ is represented by the tensor product of the one-dimensional indicator operator Eq. (24), i.e.,

$$S_{\chi_I(I)} = \bigotimes_{i=1}^{D} S_{\chi_{[a^{(i)}_{k_i},a^{(i)}_{k_i+1}]}}.$$  \hspace{1cm} (B23)

Thus, we can construct

$$S_f = \sum_{(k_i)} \sum_{m=0}^{m_d} a(k_i) (D(n))^{m} S_{\chi_I(I)}.$$  \hspace{1cm} (B24)

Note that $S_f |0\rangle \langle 0|$ is the sum of $O(n^{2D(m+1)})$ unitaries and each $Q_k$ in Eq. (24) is composed of $O(n^{d})$ gates. In general, the number of sums grows exponentially with the dimensions. However, even if the correlations between multivariate stochastic processes are important, exponential growth is inconsequential if the function depends on a small number of random variables. These issues are not unique to our algorithm. When calculating expectations using QAE, if the arguments of the function are multidimensional, exponentially greater resources are required to build a multidimensional oracle.

Thus, when the number of sums required to construct $S_f$ is independent of the dimension $D$ of the random variable, our algorithm may be particularly effective.

Appendix C: Error from Piecewise Polynomial Approximation

In this section, we evaluate the error of the expectation value $E[f(X(T))]$ from the polynomial approximation of the function $f$.

As in the main text, we divide $[0,x^{\max}]$ into $d$ intervals $\{[0,a_1],[a_1,a_2],\ldots,[a_{d-1},x^{\max}]\}$. For simplicity, we assume the equally-spaced intervals, so the width of the intervals is $h = x^{\max}/d$. We ignore the errors in the probability density function $p(x)$ that come from the tree model approximation of the SDE and the incompleteness of the ansatz of VQS because we focus on the error derived from the piecewise polynomial approximation of $f$.

We define the Lth order residual term of the Taylor expansion of $f$ around $a_k = kh$ as

$$R^L_k(x) = \frac{1}{(L+1)!} f^{(L+1)}(x-kh)^{L+1},$$  \hspace{1cm} (C1)

where $x \in [a_k,a_{k+1}] = [kh,(k+1)h]$ and $c \in [x,(k+1)h]$. As $x-kh \leq h$, $R^L_k(x)$ is $O(h^{L+1})$. When we approximate $f$ on $[a_k,a_{k+1}]$ by the Lth order Taylor expansion $g^L(x)$, the error of expectation value $E_f = \sum_{i=0}^{n-1} f(x_i)p(x_i)$ is

$$|E_f - E_g| = \left| \sum_{k=1}^{d-1} \int_{kh}^{(k+1)h} f(x)p(x)dx - \sum_{k=1}^{d-1} \int_{kh}^{(k+1)h} g^L(x)p(x)dx \right|$$

$$= \left| \sum_{k=1}^{d-1} \int_{kh}^{(k+1)h} R^L_k(x)p(x)dx \right|$$

$$\leq \max_{k} \left[ \max_{k \leq x \leq (k+1)h} \left| R^L_k(x) \right| \right] \cdot \sum_{k'=1}^{d-1} \int_{kh}^{(k'+1)h} p(x)dx$$

$$= \max_{k} \left[ \max_{k \leq x \leq (k+1)h} \left| R^L_k(x) \right| \right] \cdot \sum_{k'=1}^{d-1} \int_{kh}^{(k'+1)h} p(x)dx$$

$$= O(h^{L+1})$$

To suppress the error below $\epsilon$, it is necessary to set $d > x^{\max} \epsilon^{-1}$. From the discussion in Sec. IV, $S_f |0\rangle \langle 0|$ is the sum of $O(d^2n^{L+2})$ unitaries. Thus, we can see that $S_f |0\rangle \langle 0|$ is the sum of $O(x^{2\max} \epsilon^{-1} n^{2L+2})$ unitaries.