Variational quantum simulations of stochastic differential equations

Kenji Kubo,[1,2,*] Yuya O. Nakagawa,[3,†] Suguru Endo,[4,‡] and Shota Nagayama[1,§]
[1]*R4D, Mercari Inc., Roppongi Hills Mori Tower 18F, 6-10-1, Roppongi, Minato-ku, Tokyo 106-6118, Japan*
[2]*Graduate School of Engineering Science, Osaka University, 1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan*
[3]*QunaSys Inc., Aqua Hakusan Building 9F, 1-13-7 Hakusan, Bunkyo, Tokyo 113-0001, Japan*
[4]*NTT Secure Platform Laboratories, NTT Corporation, Musashino, Tokyo 180-8585, Japan*

(Received 1 December 2020; accepted 22 April 2021; published 21 May 2021)

Stochastic differential equations (SDEs), which model 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. 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 whereas the previous studies took 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 speedup 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.

DOI: 10.1103/PhysRevA.103.052425

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 a SDE solution as its argument. However, all of these methods require prerequisite knowledge of the SDE solution. In Ref. [22], the partial differential equation describing the time evolution of the expectation value was simulated by a variational quantum computation, which requires prederivation of the partial differential equation of the expectation value. In Refs. [23] and [24], the probability distribution of the SDE solution was embedded in the quantum state, and the expectation value

[*] kenjikun@mercari.com
[†] nakagawa@qunasys.com
[‡] suguru.endou.uc@hco.ntt.co.jp
[§] shota.nagayama@mercari.com

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 a SDE with quantum algorithms, 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) [26–30]. 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 trinomial tree-model approximation and the VQS before introducing our method. Our main theoretical results are contained in Secs. III and IV. Section III proposes a VQS-based method that simulates the dynamics of the probability distribution of the stochastic process in the trinomial tree model. The quantum circuits and operators that perform the VQS are also constructed in this section. Section IV calculates the expectation value of the random variable using the state obtained by simulating SDE with the VQS. Section V discusses the advantages of our method and compares them with previous studies. Section VI numerically evaluates our algorithm on two SDE prototypes: the geometric Brownian motion and the Ornstein-Uhlenbeck process. Conclusions are presented in Section VII. Appendix A analyses the complexity of calculating the expectation value, and Appendix B generalizes our result to a multiple-variable process. Appendix C evaluates the error of expectation values from piecewise polynomial approximation.

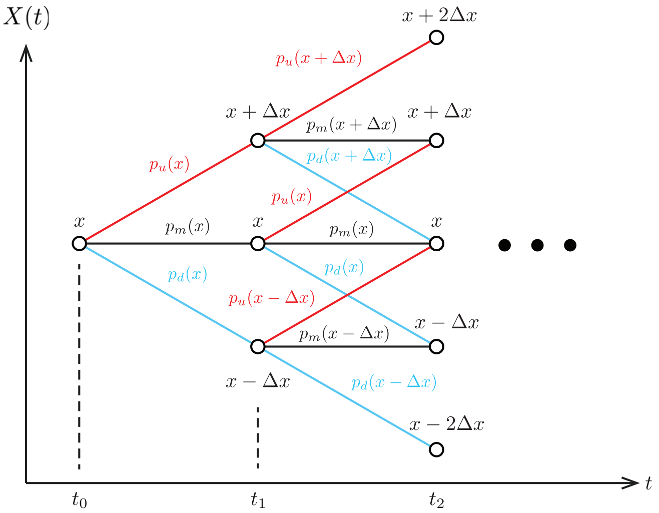


FIG. 1. Lattice of the trinomial tree model. Nodes (circles) at (t, x) represent the events in which $X(t)$ takes the value x. Edges represent the transition probabilities between the nodes. The stochastic process starts at node (t_0, x_0) and "hops" to the other nodes depending on the transition probabilities.

II. PRELIMINARIES

This section 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]

$$dX(t) = \mu(X(t), t)dt + \sigma(X(t), t)dW, \quad X(0) = x_{\text{ini}} \in I, \tag{1}$$

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 multivariable cases are described in Appendix B).

The tree model numerically simulates the time evolution of a SDE. Let us consider a SDE simulation of the process with event space $[0, x_{\max}]$ from $t = 0$ to $t = T$. We discretize the time as $t_i \equiv i\Delta t$ $(i = 0, 1, \dots, N_t)$ and the event space as $x_i \equiv i\Delta x$ $(i = 0, 1, \dots, N_x)$, where $N_t\Delta t = T$ and $N_x\Delta x = x_{\max}$. 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:

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

These probabilities were chosen to reproduce the first and second moment (mean and variance, respectively) of the random variable $X(t)$ in Eq. (1). Following the Euler-Maruyama method [7], 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

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

The corresponding moments on the trinomial tree model are

$$\begin{aligned} &E[X(t_{j+1}) - X(t_j)|X(t_j) = x_i] \\ &= [p_u(x_i, t_j) - p_d(x_i, t_j)]\Delta x, \\ &\mathrm{Var}[X(t_{j+1}) - X(t_j)|X(t_j) = x_i] \\ &= [p_u(x_i, t_j) + p_d(x_j, t_j)]\Delta x^2. \end{aligned}$$

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

$$p_u(x_i, t_j) = \frac{1}{2}\left(\frac{\sigma^2(x_i, t_j)}{\Delta x^2} + \frac{\mu(x_i, t_j)}{\Delta x}\right)\Delta t, \quad (3)$$

$$p_d(x_i, t_j) = \frac{1}{2}\left(\frac{\sigma^2(x_i, t_j)}{\Delta x^2} - \frac{\mu(x_i, t_j)}{\Delta x}\right)\Delta t, \quad (4)$$

$$p_m(x_i, t_j) = 1 - \frac{\sigma^2(x_i, t_j)}{\Delta x^2}\Delta t. \quad (5)$$

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 the second moments of the process.

The trinomial tree model simulates the SDE as follows. First, the closest value to x_{ini} in $\{x_i\}_{i\in[0,N_x]}$ is set to x_{i_0}, and the probabilities are set as $\mathrm{Prob}[X(t_0) = x_{i_0}] = 1$, $\mathrm{Prob}[X(t_0) = x_{i\neq i_0}] = 0$. Next, the probability distribution of $X(t_1 = \Delta t)$ is calculated using the transition probabilities given by Eqs. (3)–(5). Repeating this step for $X(t_j)$ $(j = 2, 3, \dots, N_t - 1)$ yields all probabilities $\mathrm{Prob}[X(t_j) = x_i]$ 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

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

We are interested in the nonunitary time evolution of an unnormalized quantum state $|\tilde{\psi}(t)\rangle$ 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 $|\tilde{v}(\boldsymbol{\theta}(t))\rangle$:

$$|\tilde{v}(\boldsymbol{\theta}(t))\rangle \equiv \alpha(t)R\{\boldsymbol{\theta}_1(t)\}|0\rangle, \quad (7)$$

where $\alpha(t)$ is a real number, $\boldsymbol{\theta}(t) \equiv (\alpha(t), \boldsymbol{\theta}_1(t)) \equiv (\alpha(t), \theta_1(t), \dots, \theta_M(t))$ are variational parameters of the ansatz, $|0\rangle$ is some reference state, and $R(\boldsymbol{\theta}_1) \equiv R_1(\theta_1)R_2(\theta_2)\cdots R_M(\theta_M)$ is a product of M parametric circuits (unitaries) composed of one parametric rotation gates $e^{i\theta_k G_k}(G_k^\dagger = G_k)$. The gates depend on their parameters and on other nonparametric gates. In particular, G_k is assumed as a multiqubit 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 $\boldsymbol{\theta}(t)$ of the ansatz. The mapping is performed by McLachlan's variational principle [33]

$$\min_{\boldsymbol{\theta}(t)} \left\| \frac{d}{dt}|\tilde{v}\{\boldsymbol{\theta}(t)\}\rangle - L(t)|\tilde{v}\{\boldsymbol{\theta}(t)\}\rangle \right\|, \quad (8)$$

where $\||\varphi\rangle\| \equiv \sqrt{\langle\varphi|\varphi\rangle}$. This equation reduces to an Euler-Lagrange equation,

$$\sum_{j=0}^{M} M_{k,j}\dot{\theta}_j(t) = V_k, \quad (9)$$

for $k = 0, \dots, M$, where

$$M_{k,j} \equiv \mathrm{Re}\left(\frac{\partial\langle\tilde{v}\{\boldsymbol{\theta}(t)\}|}{\partial\theta_k}\frac{\partial|\tilde{v}\{\boldsymbol{\theta}(t)\}\rangle}{\partial\theta_j}\right), \quad (10)$$

$$V_k \equiv \mathrm{Re}\left(\frac{\partial\langle\tilde{v}\{\boldsymbol{\theta}(t)\}|}{\partial\theta_k}L(t)|\tilde{v}\{\boldsymbol{\theta}(t)\}\rangle\right). \quad (11)$$

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

The terms $M_{k,j}$ and V_k can be evaluated by the quantum circuits depicted in Fig. 2 [29]. The normalized state $|v(\boldsymbol{\theta}(t))\rangle \equiv R(\boldsymbol{\theta}_1(t))|0\rangle$ is actually prepared on quantum computers and is multiplied by the normalization constant $\alpha(t)$ in postprocesses of the result of the circuit measurements. Decomposing the operator $L(t)$ as $L(t) = \sum_{k=1}^{k_{\mathrm{term}}(t)} \lambda_k U_k$, where U_k is an easily implementable unitary operator (e.g., a multiqubit Pauli operator) and λ_k is a complex coefficient, we must evaluate $O(M^2) + O(Mk_{\mathrm{term}}(t))$ 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, $k_{\mathrm{term}}(t)$, and the depth of the unitaries U_k must be $O(\mathrm{Poly}(n))$.

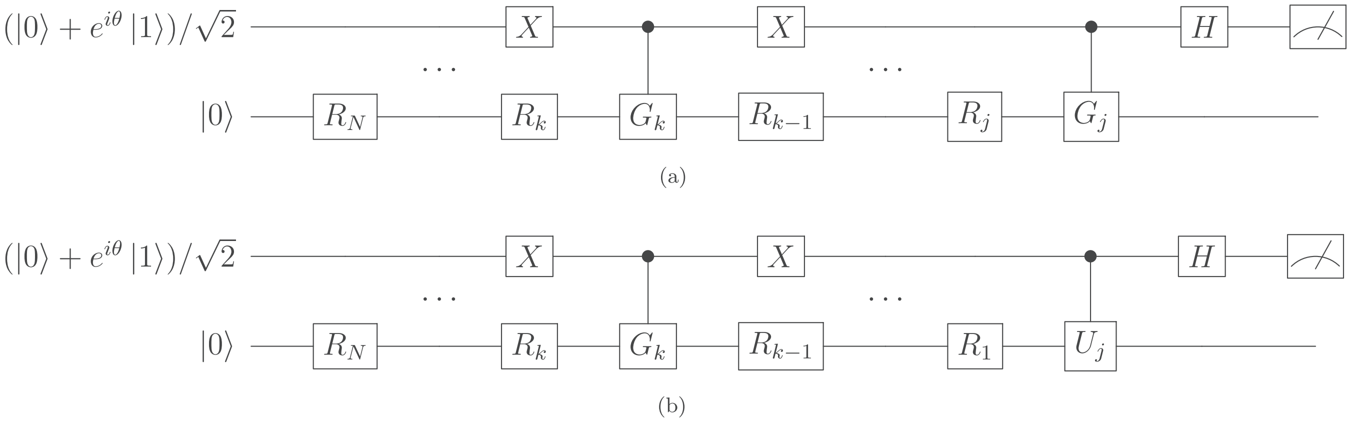


FIG. 2. Quantum circuits for evaluating (a) $M_{k,j}$ and (b) V_k.

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 nonunitary dynamics of a quantum state $|\tilde{\psi}(t)\rangle$ 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 $\{|i\rangle\}_{i=0}^{N_x}$ 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, $E[f(X(t_j))] \equiv \sum_i f(x_i)\text{Prob}[X(t_j) = x_i]$ 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 α in Eq. (7) 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

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

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

$$\begin{aligned}&\frac{P(x_i, t_{j+1}) - P(x_i, t_j)}{\Delta t}\\ &= \frac{1}{2}\left(\frac{\sigma^2(x_{i-1}, t_j)}{\Delta x^2} + \frac{\mu(x_{i-1}, t_j)}{\Delta x}\right)P(x_{i-1}, t_j)\\ &\quad+\frac{1}{2}\left(\frac{\sigma^2(x_{i+1}, t_j)}{\Delta x^2} - \frac{\mu(x_{i+1}, t_j)}{\Delta x}\right)P(x_{i+1}, t_j)\\ &\quad-\frac{\sigma^2(x_i, t_j)}{\Delta x^2}P(x_i, t_j).\end{aligned} \tag{15}$$

In the limit $\Delta t \to 0$, one obtains

$$\frac{d\vec{P}(t)}{dt} = L(t)\vec{P}(t), \tag{16}$$

$$(L(t))_{i,k} = \begin{cases} \frac{1}{2}\left(\frac{\sigma^2(x_k,t)}{\Delta x^2} + \frac{\mu(x_k,t)}{\Delta x}\right) & (i = k+1)\\ \frac{1}{2}\left(\frac{\sigma^2(x_k,t)}{\Delta x^2} - \frac{\mu(x_k,t)}{\Delta x}\right) & (i = k-1)\\ -\frac{\sigma^2(x_k,t)}{\Delta x^2} & (i = k)\\ 0 & \text{otherwise,}\end{cases} \tag{17}$$

where $\vec{P}(t) \equiv (P(x_0, t), P(x_1, t), \dots, P(x_{2^n-1}, t))^T$.

As shown in Eq. (16), the time evolution of the state $|\tilde{\psi}(t)\rangle$, or

$$\frac{d}{dt}|\tilde{\psi}(t)\rangle = \hat{L}(t)|\tilde{\psi}(t)\rangle, \tag{18}$$

where

$$\hat{L}(t) \equiv \sum_{i,k=0}^{2^n-1} \{L(t)\}_{i,k} |i\rangle\langle k|, \tag{19}$$

corresponds to the time evolution of the probability distribution $\{\mathrm{Prob}[X(t) = x_i]\}_{i=0}^{2^n-1}$. Equation (18) is the essence of our proposal to simulate VQS-based SDE simulation: specifically, the VQS algorithm applied to Eq. (18) 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. (16).

C. Construction of $L(t)$

As explained in the previous section, in the VQS, we evaluate Eqs. (10) and (11) and decompose $L(t)$ into a sum of easily implementable unitaries (composed of single-qubit, two-qubit, and few-qubit gates). These evaluations are important for a feasible VQS. This section discusses the explicit decomposition of $L(t)$ given by Eq. (19).

To express the operator $L(t)$ in Eq. (19), we define operators

$$V_+(n) \equiv \sum_{i=0}^{2^n-2} |i+1\rangle\langle i|, \quad V_-(n) \equiv \sum_{i=1}^{2^n-1} |i-1\rangle\langle i|. \tag{20}$$

These operators can be constructed from the n-qubit cyclic increment or decrement operator

$$\mathrm{CycInc}(n) \equiv \sum_{i=0}^{2^n-1} |i+1\rangle\langle i|, \quad \mathrm{CycDec}(n) \equiv \sum_{i=0}^{2^n-1} |i-1\rangle\langle i|, \tag{21}$$

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, controlled-NOT (CNOT), and X gates with $O(n)$ ancilla qubits [39]. $V_+(n)$ $[V_-(n)]$ is constructed from $\mathrm{CycInc}(n)$ $[\mathrm{CycDec}(n)]$ and an n-qubit-control Z gate $C^nZ \equiv \sum_{i=0}^{2^n-2} |i\rangle\langle i| - |2^n-1\rangle\langle 2^n-1|$, which can be implemented [13] as a product of $O(n^2)$ Toffoli, CNOT, and single-qubit gates. Using $\frac{1}{2}(C^nZ + I^{\otimes n}) = \sum_{i=0}^{2^n-2} |i\rangle\langle i|$, we can show that

$$V_+(n) = \mathrm{CycInc}(n)\tfrac{1}{2}(C^nZ + I^{\otimes n}), \tag{22}$$

$$V_-(n) = \tfrac{1}{2}(C^nZ + I^{\otimes n})\mathrm{CycDec}(n), \tag{23}$$

meaning that $V_\pm(n)$ 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

$$D(n) = \sum_{i=0}^{2^n-1} i|i\rangle\langle i| = \frac{2^n-1}{2} I^{\otimes n} - \sum_{i=1}^{n} 2^{n-i-1} Z_i, \tag{24}$$

where Z_i is a Z gate acting on the ith qubit. Therefore, $D(n)$ is a sum of $O(n)$ unitaries composed of a single-qubit gate. It follows that

$$V_+(n)[D(n)]^m = \sum_{i=0}^{2^n-2} i^m |i+1\rangle\langle i|, \tag{25}$$

$$V_-(n)[D(n)]^m = \sum_{i=1}^{2^n-1} i^m |i-1\rangle\langle i|. \tag{26}$$

Let us recall

$$\begin{aligned} L(t) = & \sum_{i=0}^{2^n-2} \frac{1}{2}\left(\frac{\sigma^2(x_i,t)}{\Delta x^2} + \frac{\mu(x_i,t)}{\Delta x}\right)|i+1\rangle\langle i| \\ & + \sum_{i=1}^{2^n-1} \frac{1}{2}\left(\frac{\sigma^2(x_i,t)}{\Delta x^2} - \frac{\mu(x_i,t)}{\Delta x}\right)|i-1\rangle\langle i| \\ & - \sum_{i=0}^{2^n-1} \frac{\sigma^2(x_i,t)}{\Delta x^2}|i\rangle\langle i|. \end{aligned}$$

Expanding $\sigma^2(x_i,t)$ and $\mu(x_i,t)$ as

$$\sigma^2(x_i,t) = \sum_{m=0}^{m_\sigma} a_{\sigma,m}(t)x_i^m, \ \mu(x_i,t) = \sum_{m=0}^{m_\mu} a_{\mu,m}(t)x_i^m, \tag{27}$$

we can decompose $L(t)$ as follows:

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

$V_+(n)[D(n)]^m$, $V_-(n)[D(n)]^m$, 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_σ, m_μ 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. (18)].

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 (1) at time $t = T$. The expectation value can be explicitly written as

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

Here, we assume that $f(x)$ in the interval $[a_k, a_{k+1}] \in \{[0, a_1], [a_1, a_2], \ldots, [a_{d-1}, x_{\max}]\}$ $(k = 0, \ldots, d-1)$ is well approximated by Lth-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. Given that 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., $E[f(X(T))] \geqslant 0$. In most situations (such as the pricing of European call options, as we see in Sec. IV C) the number of intervals d does not scale with the number of qubits n.

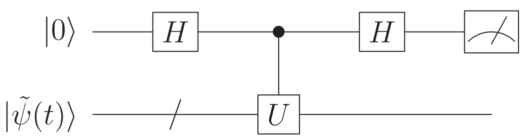


FIG. 3. Quantum circuit for evaluating the real part of an expectation value $\mathrm{Re}\langle\tilde{\psi}(t)|U|\tilde{\psi}(t)\rangle$ of a unitary operator $U = Q_iQ_{i'}^\dagger, Q_iC^nZ\cdot X^{\otimes n}Q_{i'}^\dagger$. The imaginary part of the expectation value $\mathrm{Im}\langle\tilde{\psi}(t)|U|\tilde{\psi}(t)\rangle$ is evaluated by the circuit with an $S^\dagger$ gate inserted to the left of the second H gate.

B. General formula for calculating expectation values

We now on compute the expectation value (28) using the quantum state $|\tilde{\psi}(t)\rangle$ [Eq. (12)]. First, we consider a nonunitary 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 ξ_i. It follows that

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

As $|0\rangle\langle 0| = \frac{1}{2}(I - C^nZ\cdot Z^{\otimes n})$ is also a sum of easily implementable unitaries, as explained in the previous section, the Hermitian observable $S_f|0\rangle\langle 0|S_f^\dagger$ is decomposed as

$$S_f|0\rangle\langle 0|S_f^\dagger = \sum_{i,i'} \xi_i\xi_{i'}^*\big(Q_iQ_{i'}^\dagger - Q_i(C^nZ\cdot Z^{\otimes n})Q_{i'}^\dagger\big), \tag{31}$$

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

There are two options to evaluate the quantities $\langle\tilde{\psi}(t)|Q_iQ_{i'}^\dagger|\tilde{\psi}(t)\rangle$ and $\langle\tilde{\psi}(t)|Q_i(C^nZ\cdot Z^{\otimes n})Q_{i'}^\dagger|\tilde{\psi}(t)\rangle$. The first one is to use the Hadamard test depicted in Fig. 3. The second one is to use quantum phase estimation [40,41]. The former 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. V 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 $S_{\chi[0,a]}$ hence the decomposition of S_f. As $a\in[0,x_{\max}]$, there exists $k_a\in\mathbb{N}$ such that $\Delta x2^{k_a-1}\leqslant a<\Delta x2^{k_a}$, $0<k_a\leqslant n$. The binary expansion of $a/\Delta x$ is given by $a/\Delta x = \sum_{j=0}^{k_a-1}s_j2^j$, $s_j\in\{0,1\}$. We define the list of l as $l_1, l_2, \dots, l_B$ $(=k_a-1)$ satisfying $s_l = 1$ in ascending order, and also define an interval

$$\chi_l^a = \left[2^{l_B} + \sum_{j=0}^{l-1}s_j2^j + 1,\ 2^{l_B} + \sum_{j=0}^{l}s_j2^j\right] \tag{33}$$

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

$$[0, a/\Delta x] = [0, 2^{l_B}]\cup\chi_{l_1}^a\cup\cdots\cup\chi_{l_B}^a. \tag{34}$$

The indicator operator $S_{\chi[0,a]}$ is obtained by summing the indicator operators on each interval. In the binary expansion, the k_ath and the lth 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 χ_l^a is defined as follows:

$$\begin{aligned}S_{\chi_l^a}|0\rangle &= |0\rangle^{\otimes k_a-1}\otimes|1\rangle\bigotimes_{j=0}^{n-k_a-l-1}|s_{n-k_a-j}\rangle\otimes\left(\sum_{j=0}^{l}|j\rangle\right)\\ &= 2^{l/2}I^{\otimes k_a-1}\otimes X\bigotimes_{j=0}^{n-k_a-l-1}\mathbb{X}_{s_{n-k_a-j}}\otimes H^{\otimes l}|0\rangle,\end{aligned} \tag{35}$$

where

$$\mathbb{X}_s = \begin{cases} X & (s=1)\\ I & (s=0).\end{cases} \tag{36}$$

In addition, we define

$$S_{\chi_{[0,2^{k_a-1}]}}|0\rangle = 2^{(k_a-1)/2}I^{\otimes n-k_a+1}H^{\otimes k_a-1}|0\rangle. \tag{37}$$

We can construct $S_{\chi[0,a]}$ by summing Eqs. (35) and (37) on each interval. $S_{\chi_{\alpha_k}}$ on the interval $\alpha_k\equiv[a_k, a_{k+1}]$ is

$$S_{\chi_{\alpha_k}} = S_{\chi_{[0,a_{k+1}]}} - S_{\chi_{[0,a_k]}}, \tag{38}$$

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

$$S_f|0\rangle = \sum_{k=0}^{d-1}\sum_{x_i\in\alpha_k} f(x_i)S_{\chi_{\alpha_k}}|0\rangle, \tag{39}$$

and S_f is constructed as

$$S_f = \sum_{k=0}^{d-1}\sum_{m=0}^{L} a_m^{(k)}[D(n)]^m S_{\chi_{\alpha_k}}. \tag{40}$$

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 $[a_k, a_{k+1}]\in\{[0,a_1], [a_1,a_2], \dots, [a_{d-1}, x_{\max}]\}$.

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

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

As $S_{\chi_{[a_k,a_{k+1}]}}$, $[D(n)]^m$, $|0\rangle\langle 0|$ 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, $S_f|0\rangle\langle 0|S_f^\dagger$ is the sum of $O(d^2n^{2L+2})$ 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 ϵ, the number of unitaries becomes $O(x_{\max}^2 \epsilon^{-\frac{2}{L+1}} n^{2L+2})$ (the derivation is presented in Appendix C). Note that, as L is increased, $\epsilon^{-\frac{2}{L+1}}$ 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 the evaluation of the 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 $K \geqslant 0$, interest rate $r \geqslant 0$, and maturity $T \geqslant 0$ is defined by the conditional probability

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

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 (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 $\{\text{Prob}[X_T = x_i | X_0 = x_0]\}_{i=0}^{2^n-1}$, the expectation value is

$$\begin{aligned} &e^{-rT} E_Q[\max(X_T - K, 0) | X_0 = x_0] \\ &= e^{-rT} \sum_{i=0}^{2^n-1} \text{Prob}[X_T = x_i | X_0 = x_0] \max(x_i - K, 0). \end{aligned} \tag{42}$$

For simplicity, we assume $\Delta x{=}1$ and $K{=}2^k < 2^n{-}1, k \in \mathbb{N}$. We thus obtain

$$S_{\max(i-K,0)} = [D(n) - KI] S_{\chi_{[K,2^n-1]}}. \tag{43}$$

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. (30).

V. POSSIBLE ADVANTAGES OF OUR METHOD

In this section, we discuss the advantages of our method compared with 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 the Feynman-Kac formula [1]. In fact, the authors of Ref. [22] performed a variational quantum computation of a PDE of this function. We point out two advantages of our method compared with this strategy using the Feynman-Kac formula: First, the resulting PDE by the Feynman-Kac formula must be solved backwards in time from $t = T$ to $t = 0$, with the initial condition at $t = T$ being related to the functional form of $f(X)$. It is not trivial to prepare the initial state $|\psi(T)\rangle$ corresponding to the initial condition; the authors of Ref. [22] executed an additional VQE to prepare the initial state. Second, when using the Feynman-Kac formula, the initial condition of the PDE is different for each function f for which we want to calculate the expectation value $E[f(X(T))]$. If we want to calculate a different expectation value $E[f'(X(T))]$, we need to run the whole algorithm simulating the PDE with the different initial state corresponding to f'. On the other hand, in our method, once we perform VQS, we obtain the probability distribution of $X(T)$ as a quantum state and the corresponding variational parameters to reproduce it. We only need to redo the part of the expectation value calculation (Sec. IV) for different f'.

The authors of Ref. [23] embedded the probability distribution by quantum arithmetic. Their embedding, proposed in Ref. [42], requires $O(2^n)$ gates to embed the probability distribution into an n-qubit quantum state. To moderate the gate complexity, the authors of Ref. [24] 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 an lth-order piecewise polynomial, which can be embedded with $O(ln^2)$ 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 Refs. [23] and [24], by employing QAE, the expectation value [Eq. (28)] was calculated by using an oracle that is a complex quantum gate reflecting the functional form of f for $O(1/\epsilon)$ times, where ϵ is the precision for the expectation values. The classical Monte Carlo method requires $O(1/\epsilon^2)$ sampling for precision ϵ, 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 (Fig. 3) 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 ϵ is $O(1/\gamma\epsilon^2)$ for the Hadamard test and $O(\log(1/\gamma\epsilon))$ for QPE, where γ is some factor. We note that the depth of the circuit is $O(1/\gamma\epsilon)$ in QPE, which is of the same order as the QAE whereas our method requires not a complicated oracle but a relatively small unitary. Hence, when the factor γ is not too small, our method combined with QPE can also exhibit a quantum advantage for the evaluation of the expectation values. The factor γ depends on the parameters of the polynomial approximation $(a_k^{(m)}, d, L)$, the domain of the approximated function $x_{\max}$, and the probability distribution $\{\text{Prob}[X = x_i]\}_{i=0}^{2^n-1}$. The detailed evaluation of γ 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

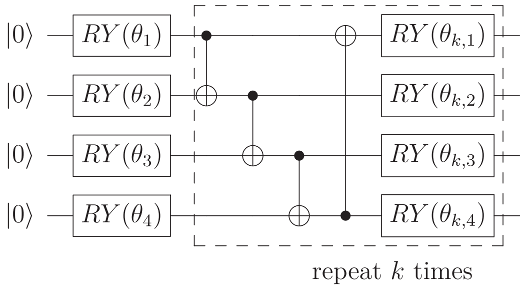


FIG. 4. In a depth-k circuit, CNOT and RY gates (enclosed by dashed lines) are repeated k times. The circuit has $4(k+1)$ parameters.

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 σ 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 η, 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 $r = 7$, $\sigma = 0.5$, $\eta = 0.01$, $\Delta x = 1$, and $t \in [0,4]$. We simulate the quantum circuits without noise using numpy [44] or jax [45]. We set the number of qubits $n = 4$ and the number of repetitions of entangle blocks $k = 2, 3$.

B. Results

Figures 5(a) and 5(b) present the numerical simulations of geometric Brownian motion and the Ornstein-Uhlenbeck process, respectively. For comparison, we also provide a probability density function (PDF) for the solution of the SDE equation obtained by solving the Fokker–Planck equation [46] analytically. We can see that our method well describes the time evolution of the probability distribution.

We calculated the means [Figs. 5(c) and 5(d)] and variances [Figs. 5(e) and 5(f)] of the resulting distributions. We also present the mean and variance obtained from the analytical solution and the solution of Eq. (16) using the Runge-Kutta method. Because of the approximation with the tree model, even the results of the Runge-Kutta method slightly differ from the analytical solution. In the case of VQS with $k = 2$, we see that the error from the analytical solution is larger than that of the $k = 3$ case. This is because the number of VQS parameters is less than the number of lattice points in the event space when $k = 2$, i.e., the degrees of freedom of the ansatz are less than the degrees of freedom of the system, and thus the errors due to the ansatz appear. In the case of $k = 3$, the number of parameters in the ansatz is sufficient, and thus the results are closer to the results of the Runge-Kutta method.

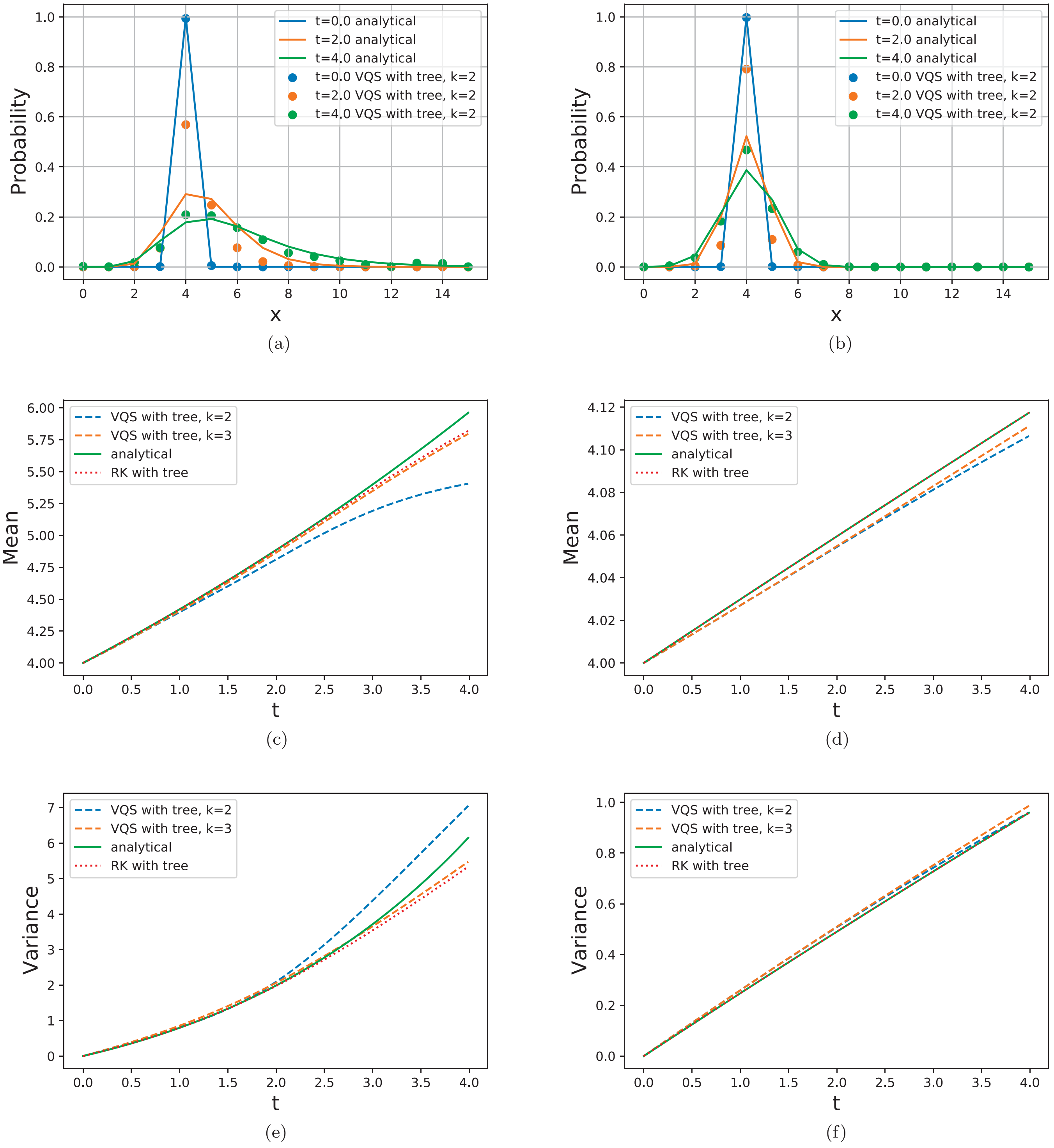
VII. CONCLUSION

This paper proposed a quantum-classical hybrid algorithm that simulates SDEs based on VQS. A continuous stochastic process was discretized in a trinomial tree model and was reformulated as a linear differential equation. The obtained differential equation was solved with VQS, obtaining quantum states representing the probability distribution of the stochastic processes. As our method can embed the probability distribution of the solution of a given SDE into the quantum state, it is applicable to general SDEs. We note that our methods can apply to the Fokker–Plank equation, which also gives the time evolution of the probability distributions of SDE solutions.

Because the embedding methods of the probability distribution of the proposed method differ from those in the conventional quantum algorithm, we proposed another method for computing the expectation value. We approximated the functions to calculate expectation values by piecewise polynomials and constructed operators corresponding to the polynomial in each interval. The operators were constructed as sums of unitary operators, which are composed of easily implementable gates. The expectation value was then computed using the sum of unitary operators. Our algorithm was validated in classical simulations 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 unitary 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 involving VQS, we decompose the matrix $L(t)$ in Eq. (16) into a sum of $O(n^{m_{\max}})$ different unitaries composed of $O(n^2)$ few-qubit gates, where $m_{\max}$ is the largest order of the polynomial expansion of μ, σ in Eq. (27). At each time step of VQS, the vector V_k in Eq. (11) is evaluated as a sum of $O(n^{m_{\max}})$ measurement results of the circuits depicted in Fig. 2. As $m_{\max}$ is typically finite and small ($\approx$1, 2) in most practical applications, 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 $O(\mathrm{Poly}(n))$. In contrast, 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 algorithm.

In the part of the expectation value evaluation of the SDE solution, we evaluate it by running different $O(d^2 n^{2L+2})$ quantum circuits, where d and L are the number of intervals and the order of the piecewise polynomial approximation of the function f in Eq. (28), respectively. Each circuit is constructed to compute an expectation value $\langle\psi|U|\psi\rangle$ of a



(a)

(b)

(c)

(d)

(e)

(f)

FIG. 5. (a) Dynamics of geometric Brownian motion and (b) dynamics of Ornstein-Uhlenbeck process. Solid lines shoe exact solutions of the SDE and circles show numerical simulations of our algorithm. (c) Time dependence of mean in geometric Brownian motion and (d) time dependence of mean in Ornstein-Uhlenbeck process. (e) Time dependence of variances of geometric Brownian motion and (f) time dependence of Ornstein-Uhlenbeck process. Dashed lines show the numerical solutions of VQS with a $k = 2$-depth and 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), (c), (e) Geometric Brownian motion with parameters $r = 0.1$, $\sigma = 0.2$. (b), (d), (f) Ornstein–Uhlenbeck Process with parameters $r = 7$, $\sigma = 0.5$, $\eta = 0.01$.

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 expectation value below ϵ is $O(1/\gamma\epsilon^2)$, where γ is a factor defined in Appendix A. This $O(1/\epsilon^2)$ scaling is the same as the classical Monte Carlo method to compute the expectation 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 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. Since it is difficult to accurately estimate the error caused by ansatz, it is also difficult to accurately estimate the overall error. Therefore, quantum speedup is not mathematically rigorous as with other variational quantum algorithms.

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, because the stochastic processes themselves are quite general, the proposed method is expected to contribute to solving problems in various fields.

ACKNOWLEDGMENT

The authors thank Keisuke Fujii for valuable discussions on variational quantum algorithms. This work is supported by Q-LEAP.

APPENDIX A: COMPLEXITY OF CALCULATING EXPECTATION VALUE

This Appendix derives the computational complexity of calculating the expectation by Eq. (30). To limit the error ϵ in the expectation value $E = (\langle\tilde{\psi}|S_f|0\rangle\langle 0|S_f^\dagger|\tilde{\psi}\rangle)^{1/2}$, we show the upper limit of error ϵ' of the expectation value for each term in Eq. (31) and find the number of measurements and gate complexity required to achieve this error. We assume that $S_f|0\rangle\langle 0|S_f^\dagger$ can be written as a linear combination of N_U unitary operators as follows:

$$S_f|0\rangle\langle 0|S_f^\dagger = \sum_{i=1}^{N_U}\beta_i U_i, \tag{A1}$$

where $\{U_i\}$ are unitary operators. We denote the error of expectation values $\langle\psi|U_i|\psi\rangle$, where $|\psi\rangle$ is the normalized state,

$$|\psi\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^n-1} p_j^2}}|\tilde{\psi}\rangle, \tag{A2}$$

where $p_j = \mathrm{Prob}[X(t) = x_i]$. We define the error as ϵ' of the expectation value of each term in a state $|\psi\rangle$. That is, the estimated expectation value of each term $\tilde{u}_i$ satisfies

$$|\tilde{u}_i - \langle\psi|U_i|\psi\rangle| \leqslant \epsilon'. \tag{A3}$$

The error in the linear combination of expectation values is determined as

$$\begin{aligned}\left|\sum_{i=1}^{N_U}\beta_i\tilde{u}_i - \langle\psi|\sum_{i=1}^{N_U}\beta_i U_i|\psi\rangle\right| &= \left|\sum_{i=1}^{N_U}\beta_i(\tilde{u}_i - \langle\psi|U_i|\psi\rangle)\right|\\ &\leqslant \sum_{i=1}^{N_U}|\beta_i||(\tilde{u}_i - \langle\psi|U_i|\psi\rangle)|\\ &\leqslant \epsilon'\sum_{i=1}^{N_U}|\beta_i|.\end{aligned} \tag{A4}$$

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

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

To upper limit the error ϵ in E, the error ϵ' must satisfy following condition:

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

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

The Hadamard test (Fig. 3) requires $O(1/\epsilon'^2)$ measurements to limit the error in the expectation value to ϵ', and the depth of the quantum circuit is $O(1)$ in terms of the unitary $U(= Q_i Q_{i'}^\dagger,\ Q_i C^n Z\cdot X^{\otimes n}Q_{i'}^\dagger)$ 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 $O(1/\epsilon')$ [40,41]. 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 an Lth-order piecewise polynomial approximation of the function f with d intervals (see Sec. IV B). Therefore, the total number of measurements required to calculate the expectation value is $O(d^2 n^{2L+2}/\gamma\epsilon^2)$ by the 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 $O(1/\gamma\epsilon)$.

Finally, we provide the estimation of the value of γ as follows: The factor $\sum_{j=0}^{2^n-1} p_j^2$ satisfies $\sum_{j=0}^{2^n-1} p_j^2 \leqslant 1$ and then $\gamma \geqslant 2E/\sum_{i=1}^{N_U}|\beta_i|$. To evaluate $\sum_{i=1}^{N_U}|\beta_i|$, we use Eq. (31) and obtain

$$\sum_{i=1}^{N_U}|\beta_i| = 2\sum_{l,l'}|\xi_l\xi_{l'}^*| = 2\sum_{l,l'}|\xi_l||\xi_{l'}|. \tag{A7}$$

We estimate the upper limit of sum of absolute values of coefficients in Eq. (40) to evaluate $\sum_l|\xi_l|$. From Eq. (24), the absolute values of the coefficients of $[D(n)]^m$ are at most $O(2^{nm}) = O(x_{\max}^m)$. As $S_{\chi_{\alpha_k}}$ is a linear combination of $S_{\chi_l^\alpha}$, the absolute values of coefficients of $S_{\chi_{\alpha_k}}$ is at most $O(2^{(k_a-1)/2})$ from Eq. (37). Since k_a satisfies $0 < k_a \leqslant n$ by definition, $O(2^{(k_a-1)/2}) = O(\sqrt{x_{\max}})$. The largest $|\xi_l|$ is as large as $(\max_k\max_m(|a_m^{(k)}|x_{\max}^m))O(\sqrt{x_{\max}})$ from Eq. (40). Thus, we obtain

$$\sum_{i=1}^{N_U}|\beta_i| \lesssim O\big(\big[\max_k\max_m|a_m^{(k)}|x_{\max}^m\big]^2 d^2 n^{2L+2}x_{\max}\big). \tag{A8}$$

Therefore, γ is larger than the ratio of E and the right-hand side of Eq. (A8).

APPENDIX B: MULTIVARIATE STOCHASTIC DIFFERENTIAL EQUATION

In this Appendix, we apply our proposed method to a SDE with multiple variables.

1. Definition and construction of the tree-model approximation

Let us consider a SDE with D variables,

$$dX_d(t) = \mu_d(X_d, t)dt + \sigma_d(X_d, t)dW_d, \tag{B1}$$

for $X_1(t), \ldots, X_D(t)$, where $\{W_d\}_{d=1}^{D}$ describes the Brownian motion with correlation $\mathrm{Corr}[W_k, W_l] = \rho_{kl}$. For simplicity, we assume an event space of each variable $X_d(t)$ as $[0, x_{\max}^{(d)}]$ and divide it into $N_x + 1$ points; that is, $x_i^{(d)} \equiv i\Delta x^{(d)}$, $\Delta x^{(d)} \equiv x_{\max}^{(d)}/N_x$ $(d = 1, \ldots, D)$. The time period of the simulation, $t \in [0, T]$, is divided into $N_t + 1$ points, $t_j \equiv j\Delta t$; that is $\Delta t \equiv T/N_t$.

We define a lattice of the tree model for Eq. (B1) with nodes $(i_1, \ldots, i_D; j)$ representing the random variables $(X_1(t_j), \ldots, X_D(t_j)) = (x_{i_1}^{(1)}, \ldots, x_{i_D}^{(D)})$, where $i_d = 0, \ldots, N_x$, $j = 0, \ldots, N_t$, $d = 1, \ldots, D$. The node transitions during time $t_j \to t_{j+1}$ are of three types:

(1) $(i_1, \ldots, i_D; j) \to (i_1, \ldots, i_D; j+1)$,

(2) $(i_1, \ldots, i_D; j) \to (i_1, \ldots, i_k \pm 1, \ldots, i_D; j+1)$,

(3) $(i_1, \ldots, i_D; j) \to (i_1, \ldots, i_k \pm 1, \ldots, i_l \pm 1, \ldots, i_D; j+1)$,

where $1 \leqslant k < l \leqslant D$. Type (1)–(3) transitions occur to nodes with identical variable values, to nodes where one-variable X_k hops to its adjacent values, and to nodes where two variables (X_k and X_l) hop to their adjacent values. The transition probabilities associated with type (1)–(3) transitions are respectively given by

$$p_m\big(x_{i_1}^{(1)}, \ldots, x_{i_D}^{(D)}, t\big),$$
$$p_{u,d}^{(k)}\big(x_{i_1}^{(1)}, \ldots, x_{i_D}^{(D)}, t\big),$$
$$p_{uu,ud,du,dd}^{(k,l)}\big(x_{i_1}^{(1)}, \ldots, x_{i_D}^{(D)}, t\big),$$

where the subscript u (d) corresponds to the sign $+$ $(-)$.

The transition probabilities can be determined identically to those of the one-variable SDE. The SDE (B1) at $(X_1(t_j), \ldots, X_D(t_j)) = (x_{i_1}^{(1)}, \ldots, x_{i_D}^{(D)})$ is discretized as

$$X_d(t_{j+1}) - X_d(t_j) = \mu_d(X_d(t_j), t)\Delta t + \sigma_d(X_d(t_j), t)\sqrt{\Delta t}z_d, \tag{B2}$$

where $\{z_d\}_{d=1}^{D}$ is sampled from the multivariable Gaussian distribution, $\mathrm{E}[z_d] = 0$, $\mathrm{Var}[z_d] = 1$, $\mathrm{Corr}[z_k, z_k] = \rho_{kl}$. The first and second conditional moments satisfy

$$E[X_d(t_{j+1}) - X_d(t_j)|X_d(t_j) = x] = \mu_d(x, t_j)\Delta t, \tag{B3}$$

$$\mathrm{Var}[X_d(t_{j+1}) - X_d(t_j)|X_d(t_j) = x] = \sigma_d^2(x, t_j)\Delta t \tag{B4}$$

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

$$\mathrm{Cov}[X_k(t_{j+1}) - X_k(t_j),\ X_l(t_{j+1}) - X_l(t_j) \\ |X_k(t_j) = x,\ X_l(t_j) = y] = \sigma_k(x, t_j)\sigma_l(y, t_j)\rho_{kl}\Delta t \tag{B5}$$

for $1 \leqslant k < l \leqslant D$. The corresponding quantities in the tree model are

$$\begin{aligned} &E[X_d(t_{j+1}) - X_d(t_j)|X_d(t_j) = x] \\ &= \Bigg[p_u^{(d)} - p_d^{(d)} + \sum_{k=1}^{d-1}\big(p_{uu}^{(k,d)} - p_{ud}^{(k,d)} + p_{du}^{(k,d)} - p_{dd}^{(k,d)}\big) \\ &\quad + \sum_{l=d+1}^{D}\big(p_{uu}^{(d,l)} + p_{ud}^{(d,l)} - p_{du}^{(d,l)} - p_{dd}^{(d,l)}\big)\Bigg]\Delta x^{(d)}, \end{aligned} \tag{B6}$$

$$\begin{aligned} &\mathrm{Var}[X_d(t_{j+1}) - X_d(t_j)|X_d(t_j) = x] \\ &= \Bigg[p_u^{(d)} + p_d^{(d)} + \sum_{k=1}^{d-1}\big(p_{uu}^{(k,d)} + p_{ud}^{(k,d)} + p_{du}^{(k,d)} + p_{dd}^{(k,d)}\big) \\ &\quad + \sum_{l=d+1}^{D}\big(p_{uu}^{(d,l)} + p_{ud}^{(d,l)} + p_{du}^{(d,l)} + p_{dd}^{(d,l)}\big)\Bigg](\Delta x^{(d)})^2 \end{aligned} \tag{B7}$$

for $d = 1, \ldots, D$ and

$$\begin{aligned} &\mathrm{Cov}[X_k(t_{j+1}) - X_k(t_j),\ X_l(t_{j+1}) - X_l(t_j) \\ &\quad |X_k(t_j) = x,\ X_l(t_j) = y] \\ &= \big(p_{uu}^{(k,l)} - p_{ud}^{(k,l)} - p_{du}^{(k,l)} + p_{dd}^{(k,l)}\big)\Delta x^{(k)}\Delta x^{(l)}. \end{aligned} \tag{B8}$$

As is the same for the case of a single variable we set the transition amplitudes by equating Eqs. (B3)–(B5) with (B6)–(B8). If the solutions of $p_{u,d}^{(k)}$, $p_{uu,ud,du,dd}^{(k,l)}$ are proportional to Δt, the linear differential equation can be derived by taking the limit of $\Delta t \to 0$ [as in the one-dimensional case of Eq. (16)].

When $D > 1$, one should note the numbers of variables and conditional expressions. As the numbers of p_m, $p_{u,d}^{(k)}$, $p_{uu,ud,du,dd}^{(k,l)}$ 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 equitation of the probability distributions of the SDE. Fixing $p_{dd}^{(k)} = p_{ud}^{(k)} = p_{du}^{(k)} = 0$, 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

$$p_{uu}^{(k,l)} = \frac{\sigma_k\sigma_l\rho_{kl}}{\Delta x^{(k)}\Delta x^{(l)}}\Delta t, \tag{B9}$$

$$p_u^{(d)} = \frac{1}{2}\left(\frac{\sigma_d^2}{(\Delta x^{(d)})^2} + \frac{\mu_d}{\Delta x^{(d)}}\right) - \sum_{k \neq d}\frac{\sigma_k\sigma_d\rho_{kd}}{\Delta x^{(k)}\Delta x^{(d)}}, \tag{B10}$$

$$p_d^{(d)} = \frac{1}{2}\left(\frac{\sigma_d^2}{(\Delta x^{(d)})^2} - \frac{\mu_d}{\Delta x^{(d)}}\right), \tag{B11}$$

$$p_m = 1 - \left[\sum_{d=1}^{D}\left(\frac{\sigma_d^2}{(\Delta x^{(d)})^2} - \sum_{k\neq d}\frac{\sigma_k\sigma_d\rho_{kd}}{\Delta x^{(k)}\Delta x^{(d)}}\right) - \sum_{k\neq l}\frac{\sigma_k\sigma_l\rho_{kl}}{\Delta x^{(k)}\Delta x^{(l)}}\right]\Delta t = 1 - \sum_{d=1}^{D}\frac{\sigma_d^2}{(\Delta x^{(d)})^2}\Delta t. \tag{B12}$$

Here, we omit the arguments of μ_d and σ_d to simplify the notation.

2. Mapping to variational quantum simulation and construction of $L(t)$

In the multivariate case, we can construct $L(t)$ as described in Sec. III. For notational simplicity, we denote $|i_1,\dots,i_D\rangle = |\boldsymbol{i}\rangle$, $|i_1,\dots,i_{d-1},i_d\pm 1,i_{d+1},\dots,i_D\rangle = |\boldsymbol{i}\pm\boldsymbol{e}_d\rangle$, $|i_1,\dots,i_k+1,\dots,i_l+1,\dots i_D\rangle = |\boldsymbol{i}+\boldsymbol{e}_k+\boldsymbol{e}_l\rangle$. Using Eqs. (B9)–(B11) and (B12), we obtain

$$L(t) = \frac{1}{2}\sum_{d=1}^{D}\sum_{i_d=0}^{2^n-2}\sum_{\boldsymbol{i}_{-d}}\left[\frac{\sigma_d^2}{(\Delta x^{(d)})^2} + \frac{\mu_d}{\Delta x^{(d)}} - \sum_{k\neq d}\frac{\sigma_k\sigma_d\rho_{kd}}{\Delta x^{(k)}\Delta x^{(d)}}\right]|\boldsymbol{i}+\boldsymbol{e}_d\rangle\langle\boldsymbol{i}| + \frac{1}{2}\sum_{d=1}^{D}\sum_{i_d=1}^{2^n-1}\sum_{\boldsymbol{i}_{-d}}\left(\frac{\sigma_d^2}{(\Delta x^{(d)})^2} - \frac{\mu_d}{\Delta x^{(d)}}\right)$$
$$\times|\boldsymbol{i}-\boldsymbol{e}_d\rangle\langle\boldsymbol{i}| + \sum_{k\neq l}\sum_{i_{k,l}=0}^{2^n-2}\sum_{\boldsymbol{i}_{-k},\boldsymbol{i}_{-l}}\frac{\sigma_k\sigma_l\rho_{kl}}{\Delta x^{(k)}\Delta x^{(l)}}|\boldsymbol{i}+\boldsymbol{e}_k+\boldsymbol{e}_l\rangle\langle\boldsymbol{i}| - \sum_{k=1}^{D}\sum_{\boldsymbol{i}}\frac{\sigma_d^2}{(\Delta x^{(d)})^2}|\boldsymbol{i}\rangle\langle\boldsymbol{i}|, \tag{B13}$$

where $\sum_{\boldsymbol{i}}$ denotes the sum of $i_m\in\{0,\dots,2^n-1\}$ for all $m\in\{1,\dots,D\}$, $\sum_{\boldsymbol{i}_{-d}}$ is the sum of $i_m\in\{0,\dots,2^n-1\}$ for all $m\neq d$, and $\sum_{\boldsymbol{i}_{-k,-l}}$ is the sum for $i_m\in\{0,\dots,2^n-1\}$ for all $m\neq k,l$.

Here, we expand $\sigma_k(x^{(k)},t)$, $\mu_k(x^{(k)},t)$ as

$$\sigma_k(x^{(k)},t) = \sum_{m=0}^{m_{\sigma_k}} a_{\sigma,m}^{(k)}(t)(x^{(k)})^m, \tag{B14}$$

$$\mu_k(x^{(k)},t) = \sum_{m=0}^{m_{\mu_k}} a_{\mu,m}^{(k)}(t)(x^{(k)})^m. \tag{B15}$$

We also define the operators

$$V_+^{(k)}(n) = I^{\otimes k-1}\otimes V_+(n)\otimes I^{\otimes D-k}, \tag{B16}$$

$$V_-^{(k)}(n) = I^{\otimes k-1}\otimes V_-(n)\otimes I^{\otimes D-k}, \tag{B17}$$

$$D^{(k)}(n) = I^{\otimes k-1}\otimes D(n)\otimes I^{\otimes D-k}. \tag{B18}$$

These operators satisfy the following equations:

$$V_+^{(k)}(n)[D^{(k)}(n)]^m = \sum_{i_k=0}^{2^n-2}\sum_{\boldsymbol{i}_{-k}} i_k^m|\boldsymbol{i}+\boldsymbol{e}_k\rangle\langle\boldsymbol{i}|, \tag{B19}$$

$$V_-^{(k)}(n)[D^{(k)}(n)]^m = \sum_{i_k=1}^{2^n-1}\sum_{\boldsymbol{i}_{-k}} i_k^m|\boldsymbol{i}-\boldsymbol{e}_k\rangle\langle\boldsymbol{i}|, \tag{B20}$$

$$V_+^{(k)}(n)[D^{(k)}(n)]^{m_k}V_+^{(l)}(n)[D^{(l)}(n)]^{m_l} = \sum_{i_k=1}^{2^n-2}\sum_{i_l=1}^{2^n-2}\sum_{\boldsymbol{i}_{-k,-l}} i_k^{m_k}i_l^{m_l}|\boldsymbol{i}+\boldsymbol{e}_k+\boldsymbol{e}_l\rangle\langle\boldsymbol{i}|. \tag{B21}$$

Using these operators, we can rewrite $L(t)$ as

$$L(t) = \sum_{d=1}^{D}\sum_{m_k=0}^{m_{\sigma_d}}\sum_{m_l=0}^{m_{\sigma_d}} a_{\sigma,m_k}^{(d)}a_{\sigma,m_l}^{(d)}(\Delta x^{(d)})^{m_k+m_l-2}\left(\frac{V_+^{(k)}+V_-^{(k)}}{2} - I\right)[D^{(d)}(n)]^{m_k+m_l} + \sum_{d=1}^{D}\sum_{m_k=0}^{m_{\mu_d}}\sum_{m_l=0}^{m_{\mu_d}} a_{\mu,m_k}^{(d)}a_{\mu,m_l}^{(d)}(\Delta x^{(d)})^{m_k+m_l-1}$$
$$\times\left(\frac{V_+^{(k)}-V_-^{(k)}}{2}\right)[D^{(d)}(n)]^{m_k+m_l} + \sum_{k\neq d}\sum_{m_k=0}^{m_{\sigma_k}}\sum_{m_l=0}^{m_{\sigma_l}} a_{\sigma,m_k}^{(k)}a_{\sigma,m_l}^{(l)}(\Delta x^{(k)}\Delta x^{(l)})^{-1}V_+^{(k)}(n)[D^{(k)}(n)]^{m_k}V_+^{(l)}(n)[D^{(l)}(n)]^{m_l}. \tag{B22}$$

Because $V_+^{(k)}(n)[D^{(k)}(n)]^m$, $V_-^{(k)}(n)[D^{(k)}(n)]^m$ are the sums of $O(n^m)$ unitaries composed of $O(n^2)$ few-qubit gates, Eq. (B22) is a 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 $\prod_{i=1}^{D}[0, x_{\max}^{(i)}]$. In each dimension, we divide $[0, x_{\max}^{(i)}]$ into d intervals $\{[a_0^{(i)}, a_1^{(i)}], \ldots, [a_{d-1}^{(i)}, x_{\max}^{(i)}]\}$ and obtain d^D regions $I(\{k_i\}) = \prod_{i=1}^{D}[a_{k_i}^{(i)}, a_{k_i+1}^{(i)}]$. The indicator operator on $I(\{k_i\})$ is represented by the tensor product of the one-dimensional indicator operator Eq. (29), i.e.,

$$S_{\chi_{I(\{k_i\})}} = \bigotimes_{i=1}^{D} S_{\chi_{[a_{k_i}^{(i)}, a_{k_i+1}^{(i)}]}}. \tag{B23}$$

Thus, we can construct

$$S_f = \sum_{\{k_i\}} \sum_{m=0}^{m_{k_i}} a_m^{(k_i)} [D(n)]^m S_{\chi_{I(\{k_i\})}}. \tag{B24}$$

Note that $S_f|0\rangle\langle 0|S_f^\dagger$ is the sum of $O(n^{2D(m+1)})$ unitaries and each Q_k in Eq. (31) is composed of $O(n^4)$ 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

$$R_k^L(x) = \frac{1}{(L+1)!} f^{(n)}(c)(x-kh)^{L+1}, \tag{C1}$$

where $x \in [a_k, a_{k+1}] = [kh, (k+1)h]$ and $c \in [x, (k+1)h]$. Because $x - kh \leqslant h$, $R_k^L(x)$ 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 the expectation value $E_f = \sum_{i=0}^{2^n-1} f(x_i)p(x_i)$ is

$$\begin{aligned}
|E_f - E_g| &= \left| \sum_{k=1}^{d-1} \int_{kh}^{(k+1)h} f(x)p(x)dx - \sum_{k=1}^{d-1} \int_{kh}^{(k+1)h} g^L(x)p(x)dx \right| \\
&= \left| \sum_{k=1}^{d-1} \int_{kh}^{(k+1)h} R_k^L(x)p(x)dx \right| \\
&\leqslant \max_k \left[\max_{kh \leqslant x \leqslant (k+1)h} \left\{ \left| R_k^L(x) \right| \right\} \right] \sum_{k'=1}^{d-1} \int_{k'h}^{(k'+1)h} p(x)dx \\
&= \max_k \left[\max_{kh \leqslant x \leqslant (k+1)h} \left\{ \left| R_k^L(x) \right| \right\} \right] \\
&= O(h^{L+1}).
\end{aligned}$$

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

[1] S. Shreve, *Stochastic Calculus for Finance II* (Springer-Verlag, New York, 2004).

[2] N. G. V. Kampen, *Stochastic Processes in Physics and Chemistry* (North Holland, Amsterdam, 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. Finan. Stud. **14**, 113 (2001).

[7] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations* (Springer-Verlag, Berlin, Heidelberg, 1992).

[8] J. Preskill, Quantum **2**, 79 (2018).
[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler *et al.*, Nature (London) **574**, 505 (2019).
[10] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96* (ACM Press, New York, 1996), pp. 212–219.
[11] P. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society Press, Washington, DC, 2002), pp. 124–134.
[12] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).
[13] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2010).
[14] A. Peruzzo, J. McClean, P. Shadbolt, M. H. Yung, X. Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, Nat. Commun. **5**, 4213 (2014).
[15] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature (London) **549**, 242 (2017).
[16] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Rev. Mod. Phys. **92**, 015003 (2020).
[17] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya *et al.*, Chem. Rev. **119**, 10856 (2019).
[18] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Phys. Rev. A **98**, 032309 (2018).
[19] M. Schuld and N. Killoran, Phys. Rev. Lett. **122**, 040504 (2019).
[20] E. Farhi and H. Neven, arXiv:1802.06002.
[21] I. Cong, S. Choi, and M. D. Lukin, Nat. Phys. **15**, 1273 (2019).
[22] F. Fontanela, A. Jacquier, and M. Oumgari, arXiv:1912.02753.
[23] P. Rebentrost, B. Gupt, and T. R. Bromley, Phys. Rev. A **98**, 022321 (2018).
[24] C. Zoufal, A. Lucchi, and S. Woerner, npj Quantum Inf. **5**, 103 (2019).
[25] P. P. Boyle, Int. Opt. J. **3**, 7 (1986).
[26] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017).
[27] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, npj Quantum Inf. **5**, 75 (2019).
[28] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Quantum **3**, 191 (2019).
[29] S. Endo, J. Sun, Y. Li, S. C. Benjamin, and X. Yuan, Phys. Rev. Lett. **125**, 010501 (2020).
[30] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, J. Phys. Soc. Jpn. **90**, 032001 (2021).
[31] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC'14* (ACM Press, New York, NY, 2014), pp. 283–292.
[32] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Commun. Math. Phys. **356**, 1057 (2017).
[33] A. McLachlan, Mol. Phys. **8**, 39 (1964).
[34] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, in *Quantum Amplitude Amplification and Estimation*, Contemporary Mathematics (American Mathematical Society, Providence, RI, 2002), pp. 53–74.
[35] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, Phys. Rev. A **101**, 010301(R) (2020).
[36] O. Kyriienko, A. E. Paine, and V. E. Elfving, arXiv:2011.10395.
[37] J. P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, arXiv:2011.03185.
[38] S. Lloyd, G. de Palma, C. Gokler, B. Kiani, Z. W. Liu, F. Tennie, T. Palmer, and M. Marvian, arXiv:2011.06571.
[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).
[42] L. Grover and T. Rudolph, arXiv:quant-ph/0208112.
[43] T. Häner, M. Roetteler, and K. M. Svore, arXiv:1805.12445.
[44] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant *et al.*, Nature (London) **585**, 357 (2020).
[45] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne, JAX: Composable transformations of Python+NumPy programs (2018).
[46] H. Risken, *The Fokker-Planck Equation* (Springer-Verlag, Berlin, Heidelberg, 1996).