npj | Quantum Information www.nature.com/npjqi

ARTICLE OPEN

Quantum-enhanced analysis of discrete stochastic processes

Carsten Blank[1], Daniel K. Park[2] and Francesco Petruccione[3,4]✉

Discrete stochastic processes (DSP) are instrumental for modeling the dynamics of probabilistic systems and have a wide spectrum of applications in science and engineering. DSPs are usually analyzed via Monte-Carlo methods since the number of realizations increases exponentially with the number of time steps, and importance sampling is often required to reduce the variance. We propose a quantum algorithm for calculating the characteristic function of a DSP, which completely defines its probability distribution, using the number of quantum circuit elements that grows only linearly with the number of time steps. The quantum algorithm reduces the Monte-Carlo sampling to a Bernoulli trial while taking all stochastic trajectories into account. This approach guarantees the optimal variance without the need for importance sampling. The algorithm can be further furnished with the quantum amplitude estimation algorithm to provide quadratic speed-up in sampling. The Fourier approximation can be used to estimate an expectation value of any integrable function of the random variable. Applications in finance and correlated random walks are presented. Proof-of-principle experiments are performed using the IBM quantum cloud platform.

npj Quantum Information (2021)7:126 ; https://doi.org/10.1038/s41534-021-00459-2

INTRODUCTION

Simulation of physical processes on quantum computers[1,2] has many facets, with recent developments improving the usage of this technology[3–14]. While one obvious task for a quantum computer is to simulate quantum mechanical behaviors of nature[1], quantum simulation of probability distributions and stochastic processes has gained attention recently[15–20]. Since quantum mechanics can be viewed as a mathematical generalization of probability theory, where nonnegative real-valued probabilities are replaced by complex-valued probability amplitudes, quantum computing appears to be a natural tool for simulating classical probabilistic processes. Intuitively, some quantum advantage is expected since the probability amplitudes can interfere, unlike in classical probabilistic computing. Indeed, a quadratic quantum speed-up has been reported for solving financial problems when compared to Monte-Carlo simulations[16,21].

Classically, Monte-Carlo methods are essential for estimating expected values of random variables in DSPs, since the number of realizations increases exponentially with the time steps. When the Monte Carlo sampling is repeated N times, the expectation value to be found converges with $\mathcal{O}(1/\sqrt{N})$ regardless of the number of realizations. The central limit theorem ensures this, but it is important to note that this convergence is only attained in the limit of $N \rightarrow \infty$. When this limit is not nearly attained, it is often crucial to have sampling strategies to reduce the variance of the estimate. The reason for this is that less likely events can, on the other hand, be of high impact. This can create a bias in the computation as the estimation can be dominated by more probable, but less important values. To correctly sample with a given N that is not near the limit of the large number, it is beneficial to modify the probability of the process in a way that balances the importance of the events. This is called importance sampling[22–28]. This makes apparent two problems with Monte-Carlo algorithms. First, it may be difficult to calculate the probability of a particular realization of a random variable. Second, the importance sampling strategy relies on a sophisticated understanding of the probability distribution.

In this manuscript, we show that the characteristic function of a DSP can be efficiently calculated on a quantum computer, and in so doing we introduce an effect called quantum brute force. The random variables of the DSP of interest do not need to be identically and independently distributed, and non-Markovian processes can also be studied. The quantum-state space that grows exponentially with the number of qubits is used to move along all paths of the discrete stochastic process simultaneously in a quantum superposition, and hence the term quantum brute force is adequate.

The method proposed in this work maps the expectation value of interest into a probability of measuring a particular computational basis state of a qubit. This probability can essentially be thought of as the success parameter of a Bernoulli trial, i.e., a coin toss, whose statistics are well-known. As a consequence, the variance is optimal in the sense that it behaves according to the central limit theorem for all sample sizes, not only for large numbers. This leads to a crucial result that no sampling strategies are necessary.

Moreover, we connect quantum amplitude estimation (AE)[29], which can be utilized to overcome the classical limit on the sampling convergence of Monte-Carlo methods[15,16,30,31], to our method. In brief, the single-qubit measurement scheme can be replaced by an application of AE to the full system. With this modification, our algorithm uses resources that scale linearly instead of quadratically, and thus achieves a quantum speed-up. The methods developed here point therefore to an exciting and promising use of quantum computers: less variance and faster convergence for Monte-Carlo sampling. Equivalently, it is possible to reduce the sample size without losing precision.

Applications of our method span extensively across many fields in science and engineering; any physical behavior that can be mathematically modeled as a DSP can be studied in principle. In particular, we show how the found simulation procedure can be applied to option pricing theory and to correlated random walks

[1] Data Cybernetics, Landsberg, Germany. [2] Sungkyunkwan University Advanced Institute of Nanotechnology, Suwon, Republic of Korea. [3] Quantum Research Group, School of Chemistry and Physics, University of KwaZulu-Natal, Durban, KwaZulu-Natal, South Africa. [4] National Institute for Theoretical and Computational Sciences (NITheCS), KwaZulu-Natal, South Africa. ✉email: petruccione@ukzn.ac.za

which leads to various applications in biology, ecology, and finance. For each example, we experimentally demonstrate the proof-of-principle using the IBM quantum cloud platform.

Although the main focus of this work is to achieve quantum sampling advantage for discrete stochastic processes, we note that quantum memory advantages have been shown for stochastic processes with causal structures[18–20,32] and in rare-events sampling[33]. An interesting connection concerning our result to this line of research comes from the work by Binder et al.[32] which presented the construction of a unitary quantum simulator for generating all possible realizations of a hidden Markov model (HMM) that requires less memory than any classical counterparts throughout the simulation. The focus of this body of work has been sampling from a distribution, but with respect to the results discussed here, it can also be used to estimate expectation values, in particular characteristic functions.

RESULTS

A discrete stochastic process can be described with n discrete random variables $X_l : \Omega_l \rightarrow \mathbb{R}$, $l = 1, \ldots, n$ for some $n \in \mathbb{N}_+$, each having at most k nonzero realizations, i.e., given a sample space Ω_l, there exist at most k elements $x_{l,0}, \ldots, x_{l,k-1} \in \mathbb{R}$ with $X_l(\Omega_l) = \{x_{l,0}, \ldots, x_{l,k-1}\}$. In plain words, a discrete stochastic process consists of n steps, and in each step, one of k possible events occurs at random with a certain probability. Hereinafter, we use the following notations. Each realization of the stochastic process is identified with an index vector $\mathbf{j} = (j_1, \ldots, j_n)^\top \in K^n$ with $K = \{0, \ldots, k-1\}$ and is denoted by $\mathbf{x}(\mathbf{j}) = (x_{1,j_1}, \ldots, x_{n,j_n})^\top$. Moreover, we define $\text{sum}\{\mathbf{x}(\mathbf{j})\} := \sum_{l=1}^{n} x_{l,j_l}$ and $\mathbf{x}^{(m)}(\mathbf{j}) = (x_{1,j_1}, \ldots, x_{m,j_m})^\top$ with $m \leq n$. The first and the second subscripts of the random variables and probabilities label the time step and the event, respectively. A quantity of interest for such processes is the expectation value of an integrable function $f : \mathbb{R} \rightarrow \mathbb{R}$ of the random variable $S_n = \sum_{l=1}^{n} X_l$

$$\mathbb{E}[f(S_n)] = \sum_{\mathbf{j} \in K^n} f(\text{sum}\{\mathbf{x}(\mathbf{j})\}) \mathbb{P}[\mathbf{X} = \mathbf{x}(\mathbf{j})]. \quad (1)$$

The summation is over all possible realizations, each indexed by $\mathbf{j}$, and $\mathbf{X}$ is the random variable representing the sequence of events that can occur in a particular realization. The number of terms in the summation (i.e., the number of realizations) is k^n.

Now, we explain how to encode the described stochastic process in a quantum state, and evaluate Eq. (1) by making a measurement on the quantum state. The quantum state consists of an index system and a data system defined in a Hilbert space $\mathcal{H}_\mathcal{I} \otimes \mathcal{H}_\mathcal{D} = \mathbb{C}^{k^n} \otimes \mathbb{C}^2$, where $\mathcal{I}$ ($\mathcal{D}$) indicates the index (data) system. Each realization of the stochastic process is represented as a unitary operator $U(\cdot) : \mathbb{R}^n \longrightarrow B(\mathcal{H}_\mathcal{I} \otimes \mathcal{H}_\mathcal{D})$ parametrized by some n-dimensional vector ($B(\mathcal{H})$ is the space of linear operators on a Hilbert space $\mathcal{H}$). Then a DSP can be represented as

$$|\Psi_f\rangle = \sum_{\mathbf{j} \in K^n} p(\mathbf{j}) |\mathbf{j}\rangle \otimes U(\mathbf{j}) |\psi\rangle. \quad (2)$$

The factor of each part of the sum is denoted by $p(\mathbf{j})$ with $\sum_\mathbf{j} p^2(\mathbf{j}) = 1$ and the state of the index system is denoted by $|\mathbf{j}\rangle = |j_n \cdots j_1\rangle = |j_n\rangle \otimes \cdots \otimes |j_1\rangle$, where $|j_l\rangle$ for $j_l = 0, \ldots, k-1$ is the orthogonal basis. Measuring an expectation value of an observable M on the data system of the final state in Eq. (2) yields the convex sum of independent expectation values measured from all k^n trajectories[34] as

$$\begin{aligned} \langle M \rangle &= \langle \Psi_f | I_\mathcal{I} \otimes M | \Psi_f \rangle \\ &= \sum_{\mathbf{j} \in K^n} p^2(\mathbf{j}) \langle \psi | U^\dagger(\mathbf{j}) M U(\mathbf{j}) | \psi \rangle \\ &= \sum_{\mathbf{j} \in K^n} p^2(\mathbf{j}) \langle M(\mathbf{j}) \rangle_\psi, \end{aligned} \quad (3)$$

where $M(\mathbf{j}) = U^\dagger(\mathbf{j}) M U(\mathbf{j})$. The coefficient $p^2(\mathbf{j})$ can be identified with the joint probability and the expectation value with the evaluation of f, i.e.,

$$p^2(\mathbf{j}) \equiv \mathbb{P}[X_1 = x_{1,j_1}, \ldots, X_n = x_{n,j_n}] \quad (4)$$

$$\langle M(\mathbf{j}) \rangle \equiv f(x_{1,j_1} + \cdots + x_{n,j_n}) \quad (5)$$

for a function $f : \mathbb{R} \rightarrow \mathbb{R}$ that we will specify below.

In the worst case, evaluation of Eq. (3) requires two expensive procedures as follows. First, k^n probabilities need to be encoded as the amplitudes of k^n computational basis given by n qudits of dimension k. This can be done with various quantum state preparation techniques with a substantial amount of computational overhead[35,36]. Next, k^n unitary operators, conditioned on all possible index states, need to be applied to an input state $|\psi\rangle$. Such operators can be expressed as

$$\text{c-}U(\mathbf{j}) = |\mathbf{j}\rangle\langle\mathbf{j}| \otimes \left(V(x_{n,j_n}) \cdots V(x_{1,j_1})\right) + |\mathbf{j}\rangle\langle\mathbf{j}|_\perp \otimes I_\mathcal{D}$$

with $V : \mathbb{R} \rightarrow \mathcal{H}_\mathcal{D}$. On the other hand, for many interesting DSPs, the number of necessary unitary operators can be reduced to $\mathcal{O}(nk)$.

Before explaining such exponential reduction in detail, we introduce two results in the next two propositions (with proofs provided in Supplementary Information) to establish the grounds for the measurement scheme.

Proposition 1 (Pauli X and Y Measurement). *Let* $V(x) = R_z(2x) = |0\rangle\langle 0| + e^{ix}|1\rangle\langle 1|$ *with* $x \in \mathbb{R}$, *then by setting* $M = \sigma_x$ or $M = \sigma_y$ *and* $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, *we find*

$$\langle I_\mathcal{I} \otimes \sigma_x \rangle_{\Psi_f} = \mathbb{E}[\cos(S_n)], \quad (6)$$

$$\langle I_\mathcal{I} \otimes \sigma_y \rangle_{\Psi_f} = \mathbb{E}[\sin(S_n)]. \quad (7)$$

As the true value of $\mathbb{E}[\cos(S_n)]$ and $\mathbb{E}[\sin(S_n)]$ must be estimated, in general, the convergence behaves according to the central limit theorem, which guarantees that the measurement statistics approaches to a normal distribution around a mean that corresponds to $\mathbb{E}[f(S_n)]$ as the number of experiments $N_S \rightarrow \infty$. The speed of convergence is moreover given by $\mathcal{O}(1/\sqrt{N_S})$. Taking into account that the Pauli measurements have two eigenvalues, the task is essentially estimating a probability of a Bernoulli trial, which specifies how the central limit theorem is realized. Given confidence of $1-\alpha$, the number of experiments to be within a margin of error $\epsilon > 0$ is $N_S = \lceil z_\alpha^2/(4\epsilon^2) \rceil$ with $z_\alpha = z(1-\alpha/2)$ being the quantile function. In contrast, with classical Monte-Carlo sampling, the convergence rate by the central limit theorem is achieved with the caveat that the Monte-Carlo simulation samples from a usually unknown stochastic process and concise estimates on the number of experiments given a margin of error are in general not easily accessible. As we see, the property that Eq. (2) encompasses all possible paths with the correct probability with a quantum measurement leads to the seemingly small advantage of knowing the convergence beforehand, irrespective of the distribution of the underlying DSP. Contemplation on this fact reveals that this is no small feat: the quantum advantage lies in the fact that no sampling strategies are necessary.

The quantum amplitude estimation algorithm[29] can provide further speed-up compared to the convergence rate of the classical Monte-Carlo method given by the central limit theorem[15,16,30,31]. The algorithm uses m ancilla qubits in addition to data and index qubits and $\mathcal{O}(\text{poly}(m))$ number of Grover-like iterators followed by $\mathcal{O}(m^2)$ Hadamard and controlled phase-shift gates for quantum Fourier transform (QFT)[37] to estimate an amplitude with an error of $\mathcal{O}(1/2^m)$. The number of repeated measurements needed to reach confidence that the estimation succeeded is independent of m. The quantum AE algorithm can be adapted to our method to construct an even more powerful strategy by formulating an AE problem as follows. Given the final state in Eq. (2) written as a linear combination $|\Psi_f\rangle = |\Psi_0\rangle + |\Psi_1\rangle$ by separating the full Hilbert space into two orthogonal subspaces, we estimate the amplitude defined as $a = \langle\Psi_1|\Psi_1\rangle$. Then, the following proposition connects AE with the DSP simulation.

Proposition 2 (Amplitude Estimation). *Let $V(x) = R_y(x) = \cos(x/2)I - i\sin(x/2)\sigma_y$, $x \in \mathbb{R}$, $|\psi\rangle = |0\rangle$, then the final state is*

$$|\Psi_f\rangle = |\Psi_0\rangle + |\Psi_1\rangle \tag{8}$$

with

$$|\Psi_0\rangle = \sum_{\mathbf{j}\in K^n} p(\mathbf{j}) \cos\left(\frac{1}{2}\mathrm{sum}\{\mathbf{x}(\mathbf{j})\}\right)|\mathbf{j}\rangle|0\rangle \tag{9}$$

$$|\Psi_1\rangle = \sum_{\mathbf{j}\in K^n} p(\mathbf{j}) \sin\left(\frac{1}{2}\mathrm{sum}\{\mathbf{x}(\mathbf{j})\}\right)|\mathbf{j}\rangle|1\rangle. \tag{10}$$

With AE the value $\tilde{a}$ will be estimated, hence $\mathbb{E}[\cos(S_n)] = 1 - 2\tilde{a}$. Conversely, if $V(x) = R_y(-x)$, $|\psi\rangle = R_y(\pi/2)|0\rangle$, then there exists a similar decomposition $|\Psi'_f\rangle = |\Psi'_0\rangle + |\Psi'_1\rangle$ so that with $a' = \langle\Psi'_1|\Psi'_1\rangle$ we therefore find $\mathbb{E}[\sin(S_n)] = 1 - 2\tilde{a}'$.

The above result opens up an exciting avenue towards a fast Monte-Carlo alternative without the need of sampling strategies. These propositions in conjunction with Theorem 1 which is stated in the following section show that a quantum computer can simulate the quantities $\mathbb{E}[\cos(S_n)]$ and $\mathbb{E}[\sin(S_n)]$ efficiently. As a result, one can calculate a random variable's characteristic function $\varphi_X(v) = \mathbb{E}[e^{ivX}] = \mathbb{E}[\cos(vX)] + i\mathbb{E}[\sin(vX)]$ with two sets of experiments per $v \in \mathbb{R}$.

In order to extend the above ideas to estimate expectation values of a range of integrable functions $f : \mathbb{R} \to \mathbb{R}$, a Fourier series is used. If f is P-periodic, then

$$f_L(x) = \sum_{l=-L}^{L} c_l e^{i\frac{2\pi l}{P}x} \tag{11}$$

is the Fourier approximation of order L for $f(x)$. By linearity of the expectation value, this approximation carries over to

$$\mathbb{E}[f_L(S_n)] = \sum_{l=-L}^{L} c_l \varphi_{S_n}\left(\frac{2\pi l}{P}\right). \tag{12}$$

As a consequence, it is possible to approximate any such expectation value in $\mathcal{O}(LN)$ experiments, where N is the number of shots per experiment. Convergence on Fourier series is a rich and mature field[38–40], which establishes basic results about convergence and the rate of convergence of each coefficient for given properties of the function f.

The underlying idea of simulating DSPs on a quantum computer has been laid out. Now we show that the simulation can be performed efficiently with a quantum circuit.

Independent increments

To deliver the underlying idea with a simple example, we start by presenting the case for DSPs with independent increments. To this end, the strategy taken is as follows. We first construct the state $\sum_{\mathbf{j}\in K^n} p(\mathbf{j})|\mathbf{j}\rangle \in \mathcal{H}_\mathcal{I}$, and systematically entangle the data system $\mathcal{D}$ with the index system. When each of the increments are independent of each other, i.e., $\mathbb{P}[X_i = x, X_j = y] = \mathbb{P}[X_i = x]\mathbb{P}[X_j = y]$ for pairwise different $i \neq j$, Eq. (1) can be written as

$$\mathbb{E}[f(S_n)] = \sum_{\mathbf{j}\in K^n} f(\mathrm{sum}\{\mathbf{x}(\mathbf{j})\}) \prod_{l=1}^{n} \mathbb{P}[X_l = x_{l,j_l}], \tag{13}$$

and $p^2_{l,j_l} = \mathbb{P}[X_l = x_{l,j_l}]$. This structure allows to partition $\mathcal{I}$ into subsystems, so-called level-index (sub)systems: $\mathcal{H}_\mathcal{I} = \bigotimes_{l=1}^{n}\mathcal{H}_{\mathcal{I}_l}$, where each $\mathcal{H}_{\mathcal{I}_l} = \mathbb{C}^k$ represents a qudit Hilbert space. Let $|a(n)\rangle \in \mathcal{H}_\mathcal{I}$ be the index-state. Then it can be described as a product state

$$|a(n)\rangle = \sum_{\mathbf{j}} p(\mathbf{j})|\mathbf{j}\rangle = \bigotimes_{l=1}^{n}\left(\sum_{j_l=0}^{k-1} p_{l,j_l}|j_l\rangle\right). \tag{14}$$

Each evolution from $|a(l)\rangle \to |a(l+1)\rangle$ is done by an unitary operator $A_{l+1} \in B(\mathcal{H}_{\mathcal{I}_l})$ as given by $A_{l+1}|0\rangle = \sum_{j=0}^{k-1} p_{l+1,j}|j\rangle$. After n levels, i.e., the application of $A = A_n \cdots A_1$, the final state $A|0\rangle_n = |a(n)\rangle$ shown in Eq. (14) is created. Since each of the operators A_l only acts on a separate subspace $\mathcal{H}_{\mathcal{I}_l}$, they commute and can be applied in parallel. This is a consequence of the independence of the increments X_l. For example, an n-step DSP with $k = 2$ possible paths at each time step can be realized with n index qubits each prepared by a single-qubit unitary operation $A_l|0\rangle = R_y(\theta_l)|0\rangle = \cos(\theta_l/2)|0\rangle + \sin(\theta_l/2)|1\rangle$, where θ_l is chosen to satisfy $\cos(\theta_l/2) = p_{l,0}$ and $\sin(\theta_l/2) = p_{l,1}$.

Now, for each step of the DSP, k unitary operators applied to the data system controlled by an index qudit split the data space into k spaces, each attached to an orthogonal subspace of the index system. In other words, in each time step, the data system undergoes k independent trajectories. Thus n steps of k controlled unitary operations allow for the encoding of k^n independent realizations of a DSP to the index-data quantum state. To this end, we identify to each realization x_{l,j_l} of the random variable X_l to the application of the operator $V(x_{l,j_l}) \in B(\mathcal{H}_\mathcal{D})$ to the data system. In fact, the operator will be defined in such a way that the lth index-level state's branches $|j_l\rangle_l$ with the amplitudes p_{l,j_l} (for $j_l \in K$) control the operator, thereby identifying the probability $p^2_{l,j_l} = \mathbb{P}[X_l = x_{l,j_l}]$ with the occurrence of $V(x_{l,j_l})$. We denote a projection to the lth index subsystem $\mathcal{I}_l$ as

$$\Pi_{lj} = |j\rangle\langle j|_l \mathrel{\widehat{=}} I_k^{\otimes l-1} \otimes |j\rangle\langle j| \otimes I_k^{\otimes n-l}, \tag{15}$$

where I_k is the k-by-k unit matrix and its perpendicular pendant as $\Pi_{lj}^{\perp}$. Then the following theorem establishes the desired result (see Supplementary Information for the proof).

Theorem 1 *Let $x \in \mathbb{R}$* and *$j \in K$. Define operators*

$$\text{c-}V_{lj_l}(x) := \Pi_{lj_l} \otimes V(x) + \Pi_{lj_l}^{\perp} \otimes I \tag{16}$$

with $\text{c-}V_{lj_l}(x) \in B(\mathcal{H}_{\mathcal{I}_l} \otimes \mathcal{H}_\mathcal{D})$. Furthermore, for $l = 1, \ldots, n$ and $\mathbf{x} \in \mathbb{R}^k$, define the operator

$$U_l(\mathbf{x}) = \prod_{j=0}^{k-1} \text{c-}V_{lj_l}(x_j) \in B(\mathcal{H}_\mathcal{I} \otimes \mathcal{H}_\mathcal{D}). \tag{17}$$

Then $U(\mathbf{j}) = U_n(\mathbf{x}_n) \cdots U_1(\mathbf{x}_1)$ for $\mathbf{x}_l \in \mathbb{R}^k$, $l = 1, \ldots, n$. The application of nk controlled operations $\text{c-}V_{lj_l}(x_{l,j_l})$ is thus necessary. See Fig. 1 for a depiction of Eqs. (16) *and* (17).

Propositions 1, 2 and Theorem 1 establish that for $V(x) = R_z(2x)$ or $V(x) = R_y(x)$, $x \in \mathbb{R}$, we can compute given expectation values of Eqs. (6) and (7), respectively, with $\mathcal{O}(nk)$ controlled gates. The above algorithm can be visualized quite intuitively. Consider a set of unitary operators $\mathcal{V} = \{V_{lj_l} = V(x_{l,j_l}) : l = 1, \ldots, n,\ j_l \in K\}$ as above. These operators span an ordered tree of height n and a maximal degree of k. The label of the nodes themselves are of secondary importance, but each edge is labeled by one of the operators of the family $\mathcal{V}$ in such a way that, given the level l ($l = 1$ is the root), each node of the lth level has k edges, each of them are labeled by V_{lj_l}, $j_l \in K$. A path between any two nodes, i.e., $(V_{lj_l}, \ldots, V_{(l+i)j_{l+i}})$, can be interpreted as a concatenation of operators, i.e., $V(j_l, \ldots, j_{l+i}) = V_{(l+i)j_{l+i}} \circ \cdots \circ V_{lj_l}$. After l level operations, paths from the root to every nodes at $(l+1)$th level in the operator tree are traveled simultaneously. Hence the data system is evolved by the operator $U(\mathbf{j}) = V(j_1, \ldots, j_l) = V_{lj_l} \circ \cdots \circ V_{1j_1}$ for all possible $\mathbf{j}$. An example of the operator tree with $n = 3$ and $k = 2$ is depicted in Fig. 2.

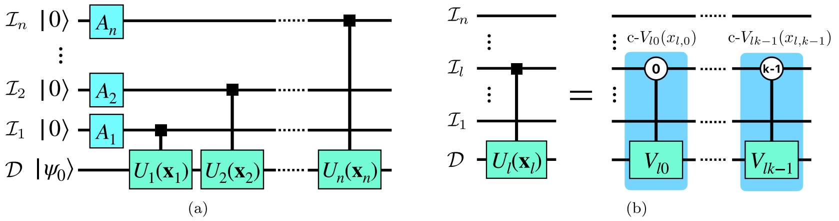

Fig. 1 Circuit design of DSPs with independent increments. a A quantum circuit including preparation of the index system (operators in blue denoted by A_l) and the realization of $c-U(\mathbf{j})$ (operators in green denoted by $U_l(\mathbf{x})$ with a verticle line connected to black squares). A black square is placed on a control register, and the green box denoted by $U_l(\mathbf{x})$ is placed on a target register. Unlike the conventional symbol for a controlled-NOT gate between qubits, the black square indicates that one of the k different unitary transformations is performed conditioned on the state of the control register of dimension k. **b** Schematic circuit representation of the applications of $c-U_l \in B(\mathcal{H}_{\mathcal{I}_l} \otimes \mathcal{H}_{\mathcal{D}})$ for $l = 1, \ldots, n$ as used in (**a**). We use the notation $V_{lj} = V(x_{l,j})$. The open circle with a number j means that the unitary operator V_lj is applied if the control register state is $|j\rangle$.

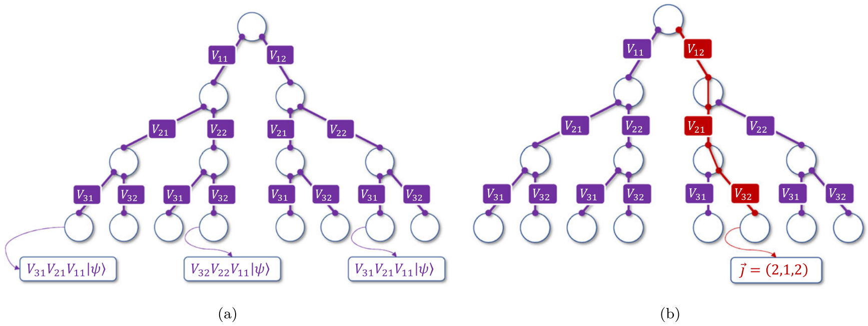
V_{11}
V_{12}
V_{21}
V_{22}
V_{31}
V_{32}
$V_{31}V_{21}V_{11}|\psi\rangle$
$V_{32}V_{22}V_{11}|\psi\rangle$
$V_{31}V_{21}V_{11}|\psi\rangle$
(a)
$\vec{j}$ = (2,1,2)
(b)

Fig. 2 Operator tree. a The operator tree spanned by a binary example ($k = 2$) of $\mathcal{V}$ for three time steps with the edges named. Each edge represents a state transformation from a parent node to a child node, and each node represents a quantum state. Three states at the final leaves starting from an initial state $|\psi\rangle$ at the root node are explicitly shown as an example. **b** The path $\mathbf{j} = (2, 1, 2)^{\top} \in K^3$ and the operator $U(\mathbf{j}) = V_{32}V_{21}V_{12}$ are shown as an example.

Path-dependent increments

For a DSP with path-dependent increments, the probability of a particular realization can be expressed as

$$\mathbb{P}[\mathbf{X}=\mathbf{x}(\mathbf{j})] = \prod_{i=2}^{n} \mathbb{P}\left[X_i = x_{i,j_i} | X_{i-1} = x_{i-1,j_{i-1}}\right] \times \mathbb{P}\left[X_1 = x_{1,j_1}\right]. \tag{18}$$

This is the first-order Markov chain that the probability to take a particular path in a given time step is determined by which path was taken in the previous step. Discussions thus far can easily be extended to such DSPs. The only difference is in the preparation of the index system, i.e., in the construction of the state $\sum_{\mathbf{j}\in K^n} p(\mathbf{j})|\mathbf{j}\rangle$. Unlike in the case of the independent increments, one needs to introduce entangling operations directly within the index system. Without loss of generality, we use the index system consisting of qubits, i.e., two paths at each level, to describe the procedure. First, the index qubit for the first level is prepared in $A_0|0\rangle = \cos(\theta_0/2)|0\rangle + \sin(\theta_0/2)|1\rangle$ as before, while the rest of the qubits are in $|0\rangle$. Then the second index qubit is prepared by using the controlled operation $|0\rangle\langle 0| \otimes R_y(\theta_2^{(0)}) + |1\rangle\langle 1| \otimes R_y(\theta_2^{(1)})$, controlled by the first index qubit, where the superscript indicates the path index of the previous step. Then the second index qubit is used as the control to prepare the third index qubit and so on. Thus, the index system state preparation can be done in n steps and the lth index qubit is prepared by applying a controlled operator $|0\rangle\langle 0| \otimes R_y(\theta_l^{(0)}) + |1\rangle\langle 1| \otimes R_y(\theta_l^{(1)})$. An example quantum circuit for preparing the index quantum state for a DSP of $n = 3$ time steps (levels), each branch to two possibilities, is shown in Fig. 3.

It is also straightforward to generalize the above procedure to implement path-dependence between events that are more than one time steps away. As an example, let us consider the $n = 3$, $k = 2$ cases depicted in Fig. 3. When the events at step 0 and step 1 have an impact on the event at step 2 (i.e., second-order Markov chain), then the single-qubit controlled rotations applied on the index qubit for step 2 should be replaced with the two-qubits controlled rotations controlled by both step 0 and step 1 index qubits. The number of three-qubit operations applied to the step 2 index qubit is four. Similarly, more complicated path-dependence, such as higher-order Markov chains, can be realized by designing multiqubit operations among index registers such that the rotation of the future state is controlled by the past states. The quantum circuit depth increases as $\mathcal{O}(k^{D_q})$ where D_q is the quantum topological memory[41–43], which satisfies $D_q \leq q$ for a qth-order Markov chain (i.e., the current state is affected by q past states). Equality is given if each past state generates a unique future state. Note that many interesting applications can be described with small q (e.g., the first- or second-order Markov chain), and in some cases, D_q can even stay constant for unbounded q[43].

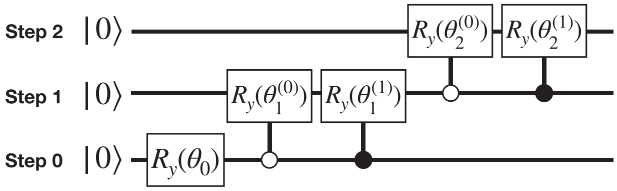

Fig. 3 Quantum circuit to prepare the index system for simulating a path-dependent DSP. The figure depicts an example of the first-order Markov chain of two steps each consisting of two realizations.

Besides the index-state preparation procedure, the analysis of the DSP follows exactly the same procedure as described in the previous section.

State-dependent increments

The increments in a given DSP can also vary in each step, determined by the state in the preceding step. In this case, the probability of a particular realization can be expressed as

$$\mathbb{P}[\mathbf{X}=\mathbf{x}(\mathbf{j})] = \prod_{i=2}^{n}\mathbb{P}\left[X_i = x_{i,j_i}|\sum_{l=1}^{i-1}X_l\right] \times \mathbb{P}\left[X_1 = x_{1,j_1}\right]. \tag{19}$$

Intuitively, in order to control the increment given a particular realization of the current step, one needs to apply controlled operation to the index register, controlled by the data register in between each successive step. We leave the explicit details on how to realize the above process with quantum circuits as future work.

State machine simulation

In the "Introduction", we mentioned that there is another way of simulating discrete stochastic processes by using a hidden Markov model (HMM). This is a state machine that has states $\Sigma=\{s_1,s_2,\ldots\}$, output alphabets $A=\{a_1,\ldots,a_k\}$, and a transition probability from state s_i to s_j while outputting the symbol a_k, i.e., $p^2(j,k|i)=\mathbb{P}\left[S_n=s_j,A_n=a_k|S_{n-1}=s_i\right]$. A quantum advantage in simulating such processes has been reported in several places[32,42,43], which requires provably less memory than its classical counterpart[18]. This uses the unifilarity property which states that the output symbol uniquely defines the next state. Consequently, the above probability notation can be shortened to $p^2(k|i)=\mathbb{P}[S_n=\lambda(k,i),A_n=a_k|S_{n-1}=s_i]$ with a function λ that maps uniquely the output symbol to a state. Models with this property are fairly common through a topological equivalence relation[41,44]; unifilar models exist for any stochastic process with finite topological memory, which even includes some instances with infinite Markov order[18,19,32,43].

While the references above focus on sampling from a distribution rather than calculating an expectation value, we point out that our methods can be used to expand these ideas in the following way. First, it is necessary to map s_i to a quantum state $|\sigma_i\rangle\in\mathcal{H}_S\subseteq\mathbb{C}^s$ on a subspace that will be added to the total system. The unifilarity property allows the HMM states to be translated to non-orthogonal quantum states (i.e., $|\sigma_i\rangle\neq|s_i\rangle$ in general) which can reduce the memory requirement even further and more interestingly, $s<|\Sigma|$. Going forward, as described above, the index space $\mathcal{H}_\mathcal{I}$ is constructed by nk-dimensional qudits. A unitary operator is applied to $\mathcal{H}_S\otimes\mathcal{H}_{\mathcal{I}_l}$ on the lth step. To be precise, given an initial state $|\sigma_i\rangle$, the first step

$$U|\sigma_i\rangle|0\rangle = \sum_{k_1\in A}p(k_1|i)\left|\sigma_{\lambda(k_1,i)}\right\rangle|k_1\rangle \tag{20}$$

creates a superposition of all possible outputs and their current state. The subsequent step creates

$$U\sum_{k_1\in A}p(k_1|i)\left|\sigma_{\lambda(k_1)}\right\rangle|k_1\rangle|0\rangle = \sum_{k_1,k_2\in A}p(k_2|\lambda(k_1,i))p(k_1|i)\left|\sigma_{\lambda(k_2,\lambda(k_1,i))}\right\rangle|k_1\rangle|k_2\rangle. \tag{21}$$

By repeating the application of U with a fresh qudit initialized in $|0\rangle$ l times, a quantum state that encodes all possible paths of the l-step HMM can be created.

By introducing a mapping from the output alphabet to a numerical value $g:A\to\mathbb{R}$, we can create the process $X_n=g(A_n)$ and then apply propositions 1 or 2 by adding a data system $\mathcal{H}_\mathcal{D}$. Therefore the quantum memory advantage reported[18] and experimentally verified[19,20] can be carried over to our reported advantages to estimate the characteristic function efficiently. An interesting avenue for future work is also to apply these methods to rare-event sampling strategies[33].

The Delta for European call option

An exciting potential application of quantum computing is financial analysis[15,16,31]. In particular, the framework developed in this work can be employed to compute the Delta of an European call option, which is central to various hedging strategies and hence of significant importance in financial engineering[21]. In brief, options are financial contracts that give the holder the right to buy (call) or sell (put) an underlying financial asset (e.g., stock) at a prescribed price (strike) on or before a prescribed date (expiration), whose value derives from the underlying asset. For European options, the holder can exercise the option only at the time of expiration. In this context, Delta measures the rate of change of the theoretical option value with respect to changes in the price of the underlying asset, which is instrumental for risk analysis.

Let S_t be stochastic process of an underlying asset then the Delta of a European call option can be expressed as

$$f(S_t)=\Phi\left(\frac{\ln\frac{S_t}{K}+\left(r+\frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}\right) \tag{22}$$

where $\Phi:\mathbb{R}\to[0,1]$ is the cumulative distribution function (CDF) of the standard normal distribution, $K>0$ is the strike price, r is the risk-free interest rate, $T-t$ is the time to maturity and σ is the volatility of the underlying asset. We are interested in the expectation value of the Delta, $\mathbb{E}[f(S_t)]$ when the underlying asset is described by a geometric Brownian motion, i.e., $S_t=S_0\exp((\mu-\sigma^2/2)t+\sigma W_t)$ where $\mu\in\mathbb{R}$ and S_0 denote the drift and the starting value, respectively. Indeed, Eq. (22) considers a value reminiscent of the log-return, which is modeled by the Brownian motion and can be approximated as a random walk by Donsker's invariance principle[45,46]. The following result establishes the way to implement this on a quantum computer.

Proposition 3 *Given n independent and identically distributed random variables X_l with $\mathbb{P}[X_l=x_1]=\mathbb{P}[X_l=x_2]=1/2$ where*

$$x_1=\frac{\mu-\frac{\sigma^2}{2}}{n\sigma\sqrt{T-t}}-\frac{1}{\sqrt{n}\sqrt{T-t}} \tag{23}$$

$$x_2=\frac{\mu-\frac{\sigma^2}{2}}{n\sigma\sqrt{T-t}}+\frac{1}{\sqrt{n}\sqrt{T-t}} \tag{24}$$

and an initial starting point of

$$x_0=\frac{\ln S_0-\ln K+\left(r+\frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}} \tag{25}$$

with $\tilde{S}_n=x_0+\sum_{l=1}^{n}X_l$, we find that

$$\mathbb{E}[f(S_t)]=\frac{1}{2}-{\sum_{l=-\infty}^{\infty}}'\frac{i}{2\pi l}e^{-2\pi^2l^2/P^2}\varphi_{\tilde{S}_n}\left(\frac{2\pi l}{P}\right) \tag{26}$$

when Φ is limited on the interval $[-P/2,P/2]$. Note that we define the sum with prime as the sum over all summands except for $l=0$ (see Supplementary Information).

When using the Pauli measurement scheme as explained in Proposition 1, one needs to use $V(vx)=R_z(2vx)$ as operators with constants $v=2\pi l/P$ for $l=-L,\ldots,L$ to evaluate the characteristic function at those points for the Fourier-series approximation. This makes a total of $2L$ experiments (L measurements for each Pauli observable) with n Hadamard gates on n index qubits for encoding the probability information, n bit-flip gates on the index qubits for implementing the controlled operations that operate if the control qubit is 0, and $2n$ controlled-Rz gates, which of each can be further decomposed to two controlled-NOT and two R_z gates, applied on a data qubit controlled by the index qubit for implementing x_1 and x_2. In addition, the initial value x_0 can be implemented by one R_z gate (see Supplementary Fig. 2).

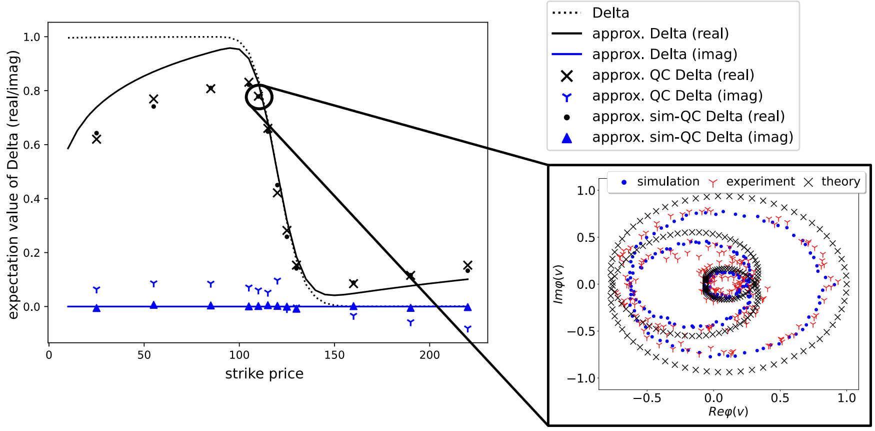

Fig. 4 Evaluation of the Delta for the European call option. Left: The Delta of an European call option is calculated for various strike prices. The underlying asset is defined with $\mu = 0, \sigma = 0.02, r = 0.02, S_0 = 100, t = 1$, and the time of maturity $T = 10$. The dotted black line is the true evaluation of the Delta. The black (blue) solid line is the real (imaginary) part of the theoretical Delta calculation obtained by a Fourier approximation with $P = 100$ and $L = 100$, which serves as the reference for the experimental validation. The black crosses (blue tri-downs) are the real (imaginary) part of the Delta calculated with the experiment on the IBM quantum computer with error mitigation applied. The black dots (blue triangles) are the simulation with noise model provided in `qiskit` with the same error mitigation applied. Right: This plot shows the characteristic function calculated in theory (×), by simulation with noise and error mitigation (dot) and the IBM quantum experiment with error mitigation (tri-down) for the example strike price of $K = 110$.

To demonstrate the proof-of-principle, we performed classical simulations and experiments of the quantum algorithm for calculating the Delta of an European call option on a IBM quantum device named `ibmqx2`. As an example, the underlying asset is given with $\mu = 0, \sigma = 0.02, r = 0.02, S_0 = 100, t = 1$, and the time of maturity $T = 10$. The Fourier-series approximation is performed by choosing $L = 100$ and $P = 100$. The standard error mitigation protocol available in `qiskit`[47] is applied to reduce experimental errors. Despite imperfections, the experiment predicts the expectation value with high accuracy for the most important region, which is where it changes drastically, i.e., where the strike price varies between 100 and 150. We also performed a classical simulation of the quantum algorithm with the standard noise model provided in `qiskit`. These results are compared with the theoretical values in Fig. 4. The comparison shows that the noise model provided in `qiskit` explains the experimental error reasonably well (see "Methods" for a brief comment on the remaining difference between the two results).

Correlated random walks

The quantum advantage against classical methods manifests when the DSP strictly requires the ability to simulate path-dependent increments. We demonstrate the simulation of correlated random walks as one such example in this section.

Correlated random walk (CRW) is a mathematical model that describes discrete random processes with correlations between successive steps[48]. It has been a useful tool to study biological processes[49,50], and can also be used to approximate fractional Brownian motion[51,52], which has broad applications for example in mathematical finance[53–55] and data network[56–59]. The correlation, often referred to as persistence, results in a local directional bias as the walker moves. More precisely, the CRW denoted by $S_n = \sum_{l=0}^{n} X_l$ with n discrete random variables X_l, $l = 1, \ldots, n$ and persistence parameters $p_l \in [0, 1]$ and $q_l \in [0, 1]$ has the following properties: (1) $X_0 = x_0$, (2) $\mathbb{P}[X_1 = x_1] = \mathbb{P}[X_1 = x_2] = 1/2$, and (3) $\mathbb{P}[X_{l+1} = x_1 | X_l = x_1] = p_l$ and $\mathbb{P}[X_{l+1} = x_2 | X_l = x_2] = q_l \; \forall \, l \geq 1$. The first two properties can be incorporated in quantum simulations easily by following the same procedure used in the previous example. Given p_l and q_l, the third property can be implemented as follows. First, all index qubits except the first one that encodes the probability distribution of X_1 are initialized in $|0\rangle$. The first index qubit is prepared in $(|0\rangle + |1\rangle)/\sqrt{2}$ in accordance with the second property. Then a controlled rotation gate is applied from each index qubit to an index qubit of the successive step. The controlled operation can be expressed as

$$|0\rangle\langle 0| \otimes R_y(\theta_{p_l}) + |1\rangle\langle 1| \otimes R_y(\theta_{q_l}), \tag{27}$$

where $\theta_{p_l} = 2\cos^{-1}(\sqrt{p_l})$ and $\theta_{q_l} = 2\cos^{-1}(\sqrt{q_l})$. For example, after one application of the above-controlled operation, the index qubits representing the probability distribution of the first two steps of the CRW is given as

$$\frac{|0\rangle(\sqrt{p_1}|0\rangle + \sqrt{1-p_1}|1\rangle) + |1\rangle(\sqrt{1-q_1}|0\rangle + \sqrt{q_1}|1\rangle)}{\sqrt{2}}. \tag{28}$$

The above state shows that $\mathbb{P}[X_2 = x_1 | X_1 = x_1] = p_1$ and $\mathbb{P}[X_2 = x_2 | X_1 = x_2] = q_1$ as required by the property (3) of the CRW.

We demonstrate the proof-of-principle with an example designed as follows. The correlated random walk is given by increments $x_{l1} = 1$ and $x_{l2} = -1$ with an initial value $x_0 = 0$. The persistence parameters are $p_l = (1/2, 2/3, 5/6, 1)$ and $q_l = (1/2, 1/3, 1/6, 0)$ with $l = 1, 2, 3, 4$. Experiments were performed to calculate the characteristic function for $v_l = 2\pi l/P$, $l = -L, \ldots, L$ with $L = 100$ and $P = 100$ on `ibmqx2`. The standard error mitigation protocol available in `qiskit` is applied to reduce experimental errors. We also performed classical simulation of the quantum algorithm with the standard noise model provided by `qiskit`. These results are compared with the theoretical values in Fig. 5.

DISCUSSION

We presented a quantum-classical hybrid framework for estimating an expectation value of a random variable based on a discrete stochastic process under the action of function that admits a Fourier-series approximation. As the main ingredient of the framework, we developed a quantum algorithm for efficiently calculating point evaluations of the characteristic function of a random variable. This may also lead to other interesting applications since the probability distribution of a random variable can be completely defined by its characteristic function. More specifically, in the quantum part, the framework proposes a succinct representation of a classical DSP as a quantum state in the form of Eq. (2). The joint probability and the value of each realization are encoded in an entangled state $|\Psi_f\rangle$, and therefore

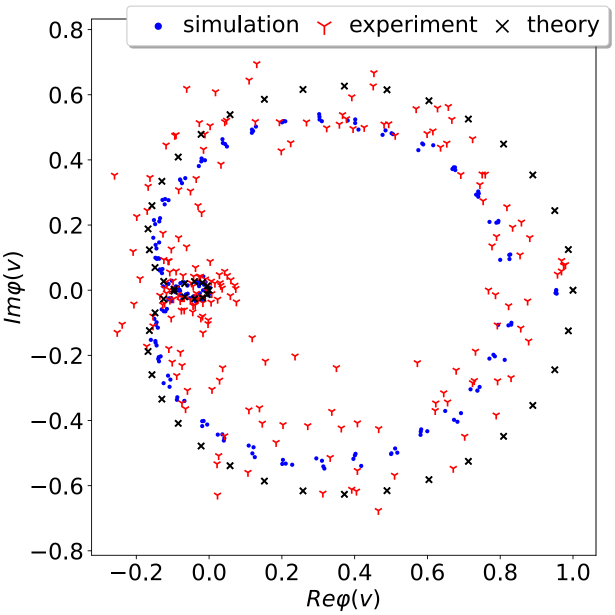

Fig. 5 Characteristic functions of a correlated random walk. Characteristic functions of a random walk is calculated theoretically (×), by simulation (dot), and by experiment (tri-down). The simulation includes the standard noise model provided in qiskit. The experiment employs the standard error mitigation technique provided in qiskit. The parameters that define the correlated random walk is described in the main text.

all necessary information about a DSP is present. We also detailed the construction of a quantum circuit for preparing the quantum state $|\Psi_f\rangle$ using the number of circuit elements that only grow linearly with the total number of time steps. For a DSP of n total steps each consisting of k possibilities, there are k^n paths. Such a process can be encoded in a quantum state using nk-dimensional index qudits (or $\lceil \log_2(k) \rceil$ qubits) and nk controlled gates. There is no need for sampling strategies since all realizations exist in quantum superposition.

Two different measurement schemes are proposed for the estimation of expectation values $\mathbb{E}[\cos(S_n)]$ and $\mathbb{E}[\sin(S_n)]$. The first scheme is to measure an expectation value of σ_x and σ_y directly on the data system of $|\Psi_f\rangle$, resulting in a convergence error of $\mathcal{O}(1/\sqrt{N})$ for N repeated experiments for any $N > 0$. This shows that the quantum brute-force approach acquires the optimal importance distribution. Furthermore, the sampling convergence can be improved by utilizing the quantum amplitude estimation technique, which promises to reduce the approximation error by a factor of $\mathcal{O}(1/2^m)$ using m additional qubits. However, the resource overhead of the latter approach is m ancilla qubits, $\mathcal{O}(\mathrm{poly}(m))$ Grover-like controlled operations, and $\mathcal{O}(m^2)$ one- and two-qubit gates for implementing QFT. Therefore, with the noisy intermediate-scale quantum (NISQ) devices, it may be desirable to use the former approach.

The advantage of this algorithm lies in two facts. First, it is a quantum-classical hybrid computation and thus is a viable candidate to be solved with near-term quantum devices. One can envisage multiple near-term quantum devices running in parallel to calculate all 2L terms independently at the same time. Similar parallelization is also suitable for multiple partitions of a large quantum device where the qubit connectivity is high within each partition but low among partitions. Second, given one type of process S_n, we can pre-compute a number of evaluations of the characteristic functions $\varphi_{S_n}(\pm v_i)$ for $v_1 < \cdots < v_L \in \mathbb{R}$ a priori. With those evaluations at hand, it is possible to assemble the Fourier-series on-demand when given the Fourier-coefficients of a function f. This approach makes it possible to invest resources to maximize the precision of the characteristic function evaluation for regularly used DSPs. Furthermore, our framework promotes the idea for a co-design quantum computer[60–63], which focuses on optimizations and design decisions at the hardware level to particularly support the DSP simulations at hand.

We underscore our findings with two interesting examples. First, we showed an application to finance, the calculation of the Delta of a European call option. The key idea was to model the stochastic behavior of an underlying asset as a Brownian motion, which is then approximated as a random walk by Donsker's invariance principle. Next, an application to DSPs with path-dependent increments is demonstrated by an example of correlated random walks. We presented the results of proof-of-principle experiments for each example to demonstrate the validity and the feasibility of our method.

The framework can be extended to multivariate functions $f: \mathbb{R}^d \to \mathbb{R}$. Given $P_1, \ldots, P_d \in \mathbb{R}$, the period of the respective argument of f and $L_1, \ldots, L_d \in \mathbb{N}_+$, a multi-dimensional Fourier-series approximation

$$f(x_1, \ldots, x_d) = \sum_{l_1=1}^{L_1} \cdots \sum_{l_d=1}^{L_d} c(\omega_1, \ldots, \omega_d) e^{i \sum_j \frac{2\pi l_j}{P_j} x_j}$$

can be applied like Eq. (11) and consequently Eq. (12). The primary harmonics to calculate are thus $\varphi_{S_n}(\mathbf{v}) = \mathbb{E}\left[e^{i\mathbf{v}\cdot\mathbf{S}_n}\right]$, which can be achieved by increasing the dimension of the index system and applying a new set of a unitary family $\mathcal{V}_r$ $(r = 1, \ldots, d)$ to the same data system $\mathcal{D}$.

The ability to simulate multivariate stochastic dynamics with the optimal variance and quadratic quantum speed-up without requiring any sampling strategies is highly beneficial for various analyses in the quantitative and computational science. The path-dependent and state-dependent DSP simulations are excellent fits for studying random walks with internal states[64], and discrete processes that converge to Ornstein–Uhlenbeck[65] and fractional Brownian motion[51]. Due to the broad applicability of these mathematical models, the framework developed in this work opens tremendous and extensive opportunities in physics, biology, epidemiology, hydrology, engineering, and finance.

In future work, we plan to provide explicit quantum circuit design for simulating state-dependent DSPs. Interesting applications to accompany this model include the analysis of stochastic epidemic models[66,67] and of hot streak[68].

METHODS

All experiments are performed using one of the publicly available IBM quantum devices consisting of five superconducting qubits, named as `ibmqx2`. In order to fully utilize the IBM quantum cloud platform, we used the IBM quantum information science kit (`qiskit`) framework[47]. The versions—as defined by PyPi version numbers—used for this work were 0.20.0.

Superconducting quantum computing devices that are currently available via the cloud service, such as those used in this work, have limited coupling between qubits. Resolving coupling constraints as well as optimizations are done in `qiskit` with a preset of so-called pass managers. The optimization level ranges from 0 to 3. As we chose the `ibmqx2` for our experiments the mapping of data register and index register follow the connectivity. For the family of devices that contains `ibmq_ourense`, one swap operation must be used to exchange the physical qubit 3 and 4. For both layouts see Supplementary Fig. 1a, b. During the compilation step of the experiments, we first use a level 0 pass and then a subsequent level 3 pass to optimize and resolve connectivity constraints. To reduce experimental errors, we used the standard error mitigation functionality of `qiskit`. This requires an extra set of experiments to be executed prior to the main experiment.

In order to understand the source of experimental discrepancy, the experimental results were compared to simulation results that take a realistic noise model into account. During the execution of an experiment, the current device parameters were gathered and stored. Upon completion, a simulation was executed with the standard `qiskit` noise model, also applying error mitigation on the result. The remaining discrepancy between experimental and simulation results can be attributed to errors that are not included in the basic error model, such as various cross-talk effects, drift, and non-Markovian noise. The standard noise model is described in the supplementary information of ref. [69] in detail.

The example for the Delta applied the strike price K from 10 to 220 in increments of 5 for the theoretical calculation, while the experiments were performed for $K = 25, 55, 85, 105, 110, 115, 120, 125, 130, 160, 190, 220$. The experiment for each K is executed for characteristic function evaluations at $v_l = 2\pi l/P$ with $l = -L, \ldots, L$, $L = 100$ and $P = 100$, each with 8129 shots. For the correlated random walk experiment, we used 32786 shots for each of the evaluations of $v_l = 2\pi l/P$ with $l = -L, \ldots, L$ and $P = L = 100$. Note that this example did not use the Fourier approximation, and P has been chosen to be the same as L so that v_l goes through one period on each side, resulting in a symmetric picture. As the IBM quantum cloud platform allows for 8192 shots per execution, we created multiple identical experiments and manually added the results.

DATA AVAILABILITY

The data that support the findings of this study are available from C.B. upon reasonable request.

Received: 4 October 2020; Accepted: 14 July 2021;
Published online: 09 August 2021

REFERENCES

1. Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982).
2. Lloyd, S. Universal quantum simulators. *Science* **273**, 1073–1078 (1996).
3. Kassal, I., Whitfield, J. D., Perdomo-Ortiz, A., Yung, M.-H. & Aspuru-Guzik, A. Simulating chemistry using quantum computers. *Annu. Rev. Phys. Chem.* **62**, 185–207 (2011).
4. Zalka, C. Simulating quantum systems on a quantum computer. *Proc. Math. Phys. Eng. Sci.* **454**, 313–322 (1998).
5. Blatt, R. & Roos, C. F. Quantum simulations with trapped ions. *Nat. Phys.* **8**, 277–284 (2012).
6. Barreiro, J. T. et al. An open-system quantum simulator with trapped ions. *Nature* **470**, 486–491 (2011).
7. Gerritsma, R. et al. Quantum simulation of the dirac equation. *Nature* **463**, 68–71 (2010).
8. Aspuru-Guzik, A. & Walther, P. Photonic quantum simulators. *Nat. Phys.* **8**, 285–291 (2012).
9. Friedenauer, A., Schmitz, H., Glueckert, J. T., Porras, D. & Schätz, T. Simulating a quantum magnet with trapped ions. *Nat. Phys.* **4**, 757–761 (2008).
10. Weimer, H., Müller, M., Lesanovsky, I., Zoller, P. & Büchler, H. P. A rydberg quantum simulator. *Nat. Phys.* **6**, 382–388 (2010).
11. Aspuru-Guzik, A., Dutoi, A. D., Love, P. J. & Head-Gordon, M. Simulated quantum computation of molecular energies. *Science* **309**, 1704–1707 (2005).
12. Lanyon, B. P. et al. Towards quantum chemistry on a quantum computer. *Nat. Chem.* **2**, 106–111 (2010).
13. Abrams, D. S. & Lloyd, S. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.* **83**, 5162 (1999).
14. Berry, D. W., Ahokas, G., Cleve, R. & Sanders, B. C. Efficient quantum algorithms for simulating sparse hamiltonians. *Commun. Math. Phys.* **270**, 359–371 (2007).
15. Rebentrost, P., Gupt, B. & Bromley, T. R. Quantum computational finance: Monte Carlo pricing of financial derivatives. *Phys. Rev. A* **98**, 022321 (2018).
16. Woerner, S. & Egger, D. J. Quantum risk analysis. *npj Quant. Inf.* **5**, 15 (2019).
17. Heinrich, S. From monte carlo to quantum computation. *Math. Comput. Simul.* **62**, 219–230 (2003).
18. Gu, M., Wiesner, K., Rieper, E. & Vedral, V. Quantum mechanics can reduce the complexity of classical models. *Nat. Commun.* **3**, 1–5 (2012).
19. Ghafari, F. et al. Dimensional quantum memory advantage in the simulation of stochastic processes. *Phys. Rev. X* **9**, 041013 (2019).
20. Ghafari, F. et al. Interfering trajectories in experimental quantum-enhanced stochastic simulation. *Nat. Commun.* **10**, 1–8 (2019).
21. Stamatopoulos, N. et al. Option pricing using quantum computers. *Quantum* **4**, 291 (2020).
22. Blanchet, J. H. & Liu, J. State-dependent importance sampling for regularly varying random walks. *Adv. Appl. Probab.* **40**, 1104–1128 (2008).
23. Hastings, W. K. *Monte Carlo Sampling Methods Using Markov Chains and Their Applications* (Oxford University Press, 1970).
24. Rubinstein, R. Y. & Kroese, D. P. *Simulation and the Monte Carlo Method*, Vol. 10 (John Wiley & Sons, 2016).
25. Owen, A. & Zhou, Y. Safe and effective importance sampling. *J. Am. Stat. Assoc.* **95**, 135–143 (1998).
26. Cappé, O., Douc, R., Guillin, A., Marin, J.-M. & Robert, C. P. Adaptive importance sampling in general mixture classes. *Stat. Comput.* **18**, 447–459 (2008).
27. Srinivasan, R. *Importance Sampling: Applications in Communications and Detection* (Springer Science & Business Media, 2013).
28. Martino, L., Elvira, V., Luengo, D. & Corander, J. Layered adaptive importance sampling. *Stat. Comput.* **27**, 599–623 (2017).
29. Brassard, G., Hoyer, P., Mosca, M. & Tapp, A. Quantum amplitude amplification and estimation. *Contemp. Math.* **305**, 53–74 (2002).
30. Montanaro, A. Quantum speedup of monte carlo methods. *Proc. Math. Phys. Eng. Sci.* **471**, 20150301 (2015).
31. Rebentrost, P. & Lloyd, S. Quantum computational finance: quantum algorithm for portfolio optimization. Preprint at https://arxiv.org/abs/1811.03975 (2018).
32. Binder, F. C., Thompson, J. & Gu, M. Practical unitary simulator for non-markovian complex processes. *Phys. Rev. Lett.* **120**, 240502 (2018).
33. Aghamohammadi, C., Loomis, S. P., Mahoney, J. R. & Crutchfield, J. P. Extreme quantum memory advantage for rare-event sampling. *Phys. Rev. X* **8**, 011025 (2018).
34. Park, D. K., Sinayskiy, I., Fingerhuth, M., Petruccione, F. & Rhee, J.-K. K. Parallel quantum trajectories via forking for sampling without redundancy. *N. J. Phys.* **21**, 083024 (2019).
35. Möttönen, M., Vartiainen, J. J., Bergholm, V. & Salomaa, M. M. Transformation of quantum states using uniformly controlled rotations. *Quantum Info Comput.* **5**, 467–473 (2005).
36. Park, D. K., Petruccione, F. & Rhee, J.-K. K. Circuit-based quantum random access memory for classical data. *Sci. Rep.* **9**, 3949 (2019).
37. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. (Cambridge University Press, 2011).
38. Zygmund, A. *Trigonometric Series*, Vol. 1 (Cambridge University Press, 2002).
39. Bary, N. K. *A Treatise on Trigonometric Series*, Vol. 1 (Elsevier, 2014).
40. Katznelson, Y. *An Introduction to Harmonic Analysis* (Cambridge University Press, 2004).
41. Crutchfield, J. P. & Young, K. Inferring statistical complexity. *Phys. Rev. Lett.* **63**, 105–108 (1989).
42. Liu, Q., Elliott, T. J., Binder, F. C., Di Franco, C. & Gu, M. Optimal stochastic modeling with unitary quantum dynamics. *Phys. Rev. A* **99**, 062110 (2019).
43. Elliott, T. J. et al. Extreme dimensionality reduction with quantum modeling. *Phys. Rev. Lett.* **125**, 260501 (2020).
44. Crutchfield, J. P. The calculi of emergence: computation, dynamics and induction. *Phys. D: Nonlinear Phenom.* **75**, 11–54 (1994).
45. Donsker, M. D. An invariance principle for certain probability limit theorems. *Mem. Am. Math. Soc.* **6**, 12 (1951).
46. Fristedt, B. E. & Gray, L. F. *A Modern Approach to Probability Theory* (Springer Science & Business Media, 2013).
47. ANIS, M. S. et al. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.5096364 (2021).
48. Gillis, J. Correlated random walk. *Math. Proc. Camb. Philos. Soc.* **51**, 639–651 (1955).
49. Bovet, P. & Benhamou, S. Spatial analysis of animals' movements using a correlated random walk model. *J. Theor. Biol.* **131**, 419 – 433 (1988).
50. Codling, E. A., Plank, M. J. & Benhamou, S. Random walk models in biology. *J. R. Soc. Interface* **5**, 813–834 (2008).
51. Mandelbrot, B. B. & Van Ness, J. W. Fractional Brownian motions, fractional noises and applications. *SIAM Rev.* **10**, 422–437 (1968).
52. Enriquez, N. A simple construction of the fractional Brownian motion. *Stoch. Process. Their Appl.* **109**, 203 – 223 (2004).
53. Elliott, R. J. & van der Hoek, J. Fractional brownian motion and financial modelling. In Kohlmann M., Tang S. (eds) *Mathematical Finance. Trends in Mathematics*, 140–151, https://doi.org/10.1007/978-3-0348-8291-0_13 (Birkhäuser Basel, 2001).
54. Cheridito, P. Arbitrage in fractional Brownian motion models. *Financ. Stoch.* **7**, 533–553 (2003).
55. Rostek, S. & Schöbel, R. A note on the use of fractional Brownian motion for financial modeling. *Econ. Model.* **30**, 30 – 35 (2013).
56. Norros, I. On the use of fractional Brownian motion in the theory of connectionless networks. *IEEE J. Sel. Areas Commun.* **13**, 953–962 (1995).
57. Konstantopoulos, T. & Lin, S.-J. In *Stochastic Networks* (eds Glasserman, P., Sigman, K. & Yao, D. D.) 257–273 (Springer New York, 1996).
58. Belly, S. & Decreusefond, L. In *Fractals in Engineering* (eds Lévy Véhel, J., Lutton, E. & Tricot, C.) 170–184 (Springer London, 1997).
59. Filatova, D. Mixed fractional Brownian motion: some related questions for computer network traffic modeling. In *2008 International Conference on Signals and Electronic Systems*, 393–396 (IEEE, 2008).

60. Brown, K. R., Kim, J. & Monroe, C. Co-designing a scalable quantum computer with trapped atomic ions. *npj Quant. Inf.* **2**, 1–10 (2016).
61. Langford, N. et al. Experimentally simulating the dynamics of quantum light and matter at deep-strong coupling. *Nat. Commun.* **8**, 1–10 (2017).
62. Parra-Rodriguez, A., Lougovski, P., Lamata, L., Solano, E. & Sanz, M. Digital-analog quantum computation. *Phys. Rev. A* **101**, 022305 (2020).
63. Lamata, L., Parra-Rodriguez, A., Sanz, M. & Solano, E. Digital-analog quantum simulations with superconducting circuits. *Adv. Phys.* **3**, 1457981 (2018).
64. Hughes, B. *Random Walks and Random Environments: Random Walks* (Clarendon Press, 1995).
65. Uhlenbeck, G. E. & Ornstein, L. S. On the theory of the brownian motion. *Phys. Rev.* **36**, 823–841 (1930).
66. Tuckwell, H. C. & Williams, R. J. Some properties of a simple stochastic epidemic model of sir type. *Math. Biosci.* **208**, 76 – 97 (2007).
67. Allen, L. J. A primer on stochastic epidemic models: formulation, numerical simulation, and analysis. *Infect. Dis. Model.* **2**, 128 – 142 (2017).
68. Liu, L. et al. Hot streaks in artistic, cultural, and scientific careers. *Nature* **559**, 396–399 (2018).
69. Blank, C., Park, D. K., Rhee, J.-K. K. & Petruccione, F. Quantum classifier with tailored quantum kernel. *npj Quant. Inf.* **6**, 1–7 (2020).

ACKNOWLEDGEMENTS

We acknowledge the use of IBM Q for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team. We thank Philipp Leser, Mile Gu, and Jayne Thompson for fruitful discussions, and James Crutchfield for sharing references. This research is supported by the National Research Foundation of Korea (No. 2019R1I1A1A01050161 and 2021M3H3A1038085), Quantum Computing Development Program (No. 2019M3E4A1080227), and the South African Research Chair Initiative of the Department of Science and Technology and the National Research Foundation (UID: 64812).

AUTHOR CONTRIBUTIONS

C.B. and D.K.P. contributed equally to this work. C.B. and D.K.P. designed and analyzed the model. C.B. conducted the simulations and the experiments on the IBM Q. All authors reviewed and discussed the analyses and results, and contributed toward writing the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-021-00459-2.

Correspondence and requests for materials should be addressed to F.P.

Reprints and permission information is available at http://www.nature.com/reprints

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