



Extracting a function encoded in amplitudes of a quantum state by tensor network and orthogonal function expansion

Koichi Miyamoto¹ · Hiroshi Ueda^{1,2,3}

Received: 12 December 2022 / Accepted: 5 April 2023 / Published online: 7 June 2023
© The Author(s) 2023

Abstract

There are quantum algorithms for finding a function f satisfying a set of conditions, such as solving partial differential equations, and these achieve exponential quantum speedup compared to existing classical methods, especially when the number d of the variables of f is large. In general, however, these algorithms output the quantum state which encodes f in the amplitudes, and reading out the values of f as classical data from such a state can be so time-consuming that the quantum speedup is ruined. In this study, we propose a general method for this function readout task. Based on the function approximation by a combination of tensor network and orthogonal function expansion, we present a quantum circuit and its optimization procedure to obtain an approximating function of f that has a polynomial number of degrees of freedom with respect to d and is efficiently evaluable on a classical computer. We also conducted a numerical experiment to approximate a finance-motivated function to demonstrate that our method works.

Keywords Quantum algorithm · Tensor network · Orthogonal function expansion

1 Introduction

Quantum computing is an emerging technology that is expected to provide speedup for various classically time-consuming problems. Following recent advances, researchers

✉ Koichi Miyamoto
miyamoto.kouichi.qiqb@osaka-u.ac.jp

Hiroshi Ueda
ueda.hiroshi.qiqb@osaka-u.ac.jp

¹ Center for Quantum Information and Quantum Biology, Osaka University, Toyonaka, Osaka 560-0043, Japan

² JST, PRESTO, Kawaguchi, Saitama 332-0012, Japan

³ Computational Materials Science Research Team, RIKEN Center for Computational Science (R-CCS), Kobe 650-0047, Japan

are now investigating its practical applications. Some fundamental quantum algorithms have evolved into solvers for more concrete numerical problems. For example, the Harrow–Hassidim–Lloyd algorithm [1] to solve linear equation systems and its extensions [2–10] led to quantum solvers for ordinary differential equations (ODEs) [11–14] and partial differential equations (PDEs) [3, 15–19], including even nonlinear systems [20–25]. The practical use of these algorithms has been proposed in various fields such as epidemiology [20, 26], fluid dynamics [20], and financial derivative pricing [27]. Compared with existing classical methods, these quantum algorithms achieve exponential speedup with respect to the size of the problem. For an ODE, the problem size is the number of equations in the system; for a PDE, it is the dimension d of the domain of the solution function f , that is, the number of the variables of f . Thus, they are among the most promising applications of quantum computing. In addition to the above algorithms that run on fault-tolerant quantum computers (FTQCs), some algorithms for noisy intermediate-scale quantum (NISQ) devices also solve ODEs and PDEs [28–42].

However, when we consider these and other quantum solvers and the speedup they provide, we should pay attention to the meaning of the words “solve” and “speedup.” That is, we must resolve the issue of *how to extract the function values from the quantum state*, which might ruin the quantum speedup.

The detail is as follows. As pointed out in [43], many quantum algorithms output the solution not as a figure that we eventually want, but as the quantum state in which the solution is encoded as the amplitudes. For example, quantum algorithms for solving a PDE output the quantum state in the form of

$$|f\rangle := \frac{1}{C} \sum_{i=0}^{N_{\text{gr}}-1} f(\vec{x}_i) |i\rangle, \quad (1)$$

where f is the solution function, $\vec{x}_0, \vec{x}_1, \dots, \vec{x}_{N_{\text{gr}}-1}$ are the N_{gr} grid points in the domain of f , $|0\rangle, |1\rangle, \dots, |N_{\text{gr}}-1\rangle$ are the computational basis states, and C is the normalization factor. Reading out the values of the function as classical data from such a quantum state can often be a bottleneck. For the example of solving a PDE, the number of grid points N_{gr} is exponentially large with respect to the dimension d : naively, taking n_{gr} grid points in each dimension leads to $N_{\text{gr}} = n_{\text{gr}}^d$ points in total. This makes the amplitude $\frac{1}{C} f(\vec{x}_i)$ of each basis state in $|f\rangle$ exponentially small when the amplitude is not localized on certain grids. This means that when we try to retrieve the amplitude and then the function value $f(\vec{x}_i)$ at the point x_i using methods such as quantum amplitude estimation [44, 45], an exponentially large time overhead is added. Therefore, even if a quantum algorithm exists that generates the state $|f\rangle$ exponentially faster than a classical algorithm outputs the value of $f(\vec{x}_i)$, obtaining $f(\vec{x}_i)$ through the quantum algorithm might not be faster than the classical one.

If we want to obtain the function values at many points, for example, to plot the values as a graph, the situation worsens. To obtain the function values at M points, the quantum algorithm must be repeated M times.

Motivated by this background, this study focuses on how to efficiently extract the function encoded as the amplitudes in the quantum state $|f\rangle$, given an oracle

O_f to generate it. Although we can resolve this issue using the specific nature of the problem in some cases such as derivative pricing considered in [27], where the martingale property of the derivative price is used to calculate the function value at a point, nevertheless, it is desirable to devise methods that are generally applicable.

The method proposed in this study is twofold. First, we use an *orthogonal function* system. Orthogonal functions such as trigonometric functions, Legendre polynomials, Chebyshev polynomials, and so on are the sequences of functions orthogonal to each other with respect to some inner product, which is defined as the weighted integral of the product of two functions over the domain or the sum of the values of the product at the specific nodes. Orthogonal functions are often used for function approximation. Any function f satisfying some conditions of smoothness is approximated as $f \approx \sum_l a_l P_l$, the series of orthogonal functions $\{P_l\}_l$, with the coefficient a_l given by the inner product of f and P_l . We expect that orthogonal function expansion may also be used in the quantum setting, because, as explained later, the coefficient a_l is given by $\langle P_l|f \rangle$ multiplied by a known factor, with $|P_l\rangle$ being the quantum state in which P_l is encoded like $|f\rangle$. Thus, by estimating $\langle P_l|f \rangle$ for every l up to a sufficiently high order, we obtain the orthogonal function expansion \tilde{f} of f , and then the approximate values of f at arbitrary points by evaluating \tilde{f} . This approach seems promising because we expect that $\langle P_l|f \rangle$ is not exponentially small, unlike the amplitudes of the computational basis states in $|f\rangle$ (see Sect. 2.3).

However, in the high-dimensional case, the above approach still suffers from high complexity. If we use the D orthogonal functions $\{P_l\}_{l \in [D]_0}$ ¹ for an accurate approximation in the one-dimensional case, the naive way to achieve similar accuracy in the d -dimensional case is to use the tensorized functions $\{P_{\vec{l}}\}_{\vec{l} \in [D]_0^d}$, where $P_{\vec{l}}(x_1, \dots, x_d) = \prod_{i=1}^d P_{l_i}(x_i)$ for $\vec{l} = (l_1, \dots, l_d)$. In this way, because the total number of $P_{\vec{l}}$'s is D^d , obtaining the coefficients $a_{\vec{l}}$ for all $P_{\vec{l}}$'s and then the orthogonal function expansion of f exhibits exponential complexity, and so does evaluating the resultant expansion. Although this is less serious than reading out the amplitudes in $|f\rangle$ because we take $D < n_{gr}$, the exponential dependence of the complexity on d still exists.

Then, we make use of the second building block of our method: *tensor network*, especially the type called *matrix product state (MPS)*, which is simple and widely used (for reviews, see [46, 47]). Tensor network is an approximation scheme for high-order tensors as a contraction of lower-order tensors. In some situations, it approximates the original tensor well, reducing the degrees of freedom (DOF) and data volume. It was originally invented in quantum many-body physics to approximate wave functions in intractably high-dimensional Hilbert spaces; however, it is currently used to reduce complexity in various fields including function approximation [48–54]. A recent study [54] showed that the complexity of the tensor network approximation of a d -dimensional function does not scale exponentially on d under certain conditions, which indicates the powerful approximation ability of tensor network.

Another advantage of tensor network is its compatibility with quantum computing. That is, we can generate a quantum state in which a type of tensor network is amplitude-encoded using a simple quantum circuit [55]. Moreover, there is a general procedure

¹ For $n \in \mathbb{N}$, we define $[n]_0 := \{0, 1, \dots, n - 1\}$.

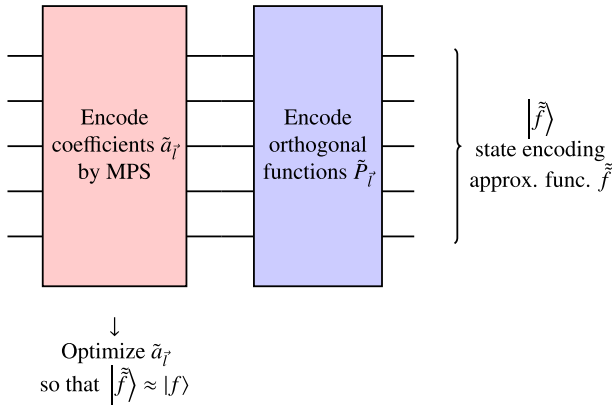


Fig. 1 Overview of the quantum circuit used in the proposed method

for optimizing such a tensor network circuit to maximize the fidelity between the resulting state and a given state [56]. Therefore, we reach the following idea: given O_f , we find a tensor network approximation of the coefficients a_f in the orthogonal function expansion of f and then an approximate function of f through quantum circuit optimization.

In the remainder of this paper, we show how this idea is concretely realized. We present the tensor-network-based quantum circuit and optimization procedure for making the generated state close to $|f\rangle$, based on [56]. The parameters in the tensor network are easily read out from the circuit. Note that this method can be used on both fault-tolerant quantum computers and noisy intermediate-scale quantum computers.

The remainder of this paper is organized as follows. Section 2 presents our method. Beginning with an explanation of the problem setting under consideration, we present the quantum circuit we use and the procedure to optimize it. Section 3 describes the numerical experiment we conducted to validate our method, which demonstrates that the method successfully approximates a finance-motivated multivariate function. Section 4 summarizes this paper.

2 Our method

2.1 Overview

Before providing a detailed explanation of the proposed method, we present a high-level overview. For the given function f , our goal is to obtain an approximation function efficiently computable on a classical computer. For this purpose, we use a quantum circuit, whose overview is shown is given in Fig. 1. This consists of two parts. In the first half, we encode the coefficients \tilde{a}_f of the orthogonal function series written in the form of an MPS into the amplitudes of the quantum state. In the latter half, we operate the quantum circuit that encodes the values of the orthogonal functions into the amplitudes. These two operations yield the quantum state $|\tilde{f}\rangle$, in which the

approximation function \tilde{f} based on the MPS and orthogonal function expansion is encoded in the amplitudes. Then, we optimize the first half of the circuit such that $|\tilde{f}\rangle$ approaches $|f\rangle$, the state encoding f . As a result, we obtain the coefficients in the form of an MPS optimized to approximate f , and then the approximation function \tilde{f} , which is efficiently computable by virtue of the MPS.

The approximation problem addressed in this study is explained in more detail in Sect. 2.2 and the setting of the oracles available is shown in Sect. 2.3. The MPS approximation of the coefficients is explained in Sect. 2.4, and encoding it into the quantum state, that is, the first half of the circuit, is described in Sect. 2.5. The entire circuit including the latter half and optimization of the MPS are explained in Sect. 2.6.

2.2 Problem description

As the approximation target, we consider a real-valued function f on the d -dimensional hyperrectangle $\Omega := [L_1, U_1] \times \dots \times [L_d, U_d]$, where, for $i \in [d]$,² L_i and U_i are real values such that $L_i < U_i$.

For any $i \in [d]$, we also consider orthogonal functions $\{P_l^i\}_l$ on $[L_i, U_i]$ labeled by $l \in \mathbb{N}_0 := \{0\} \cup \mathbb{N}$. These functions are characterized by the orthogonal relation that, for any $l, l' \in \mathbb{N}_0$,

$$\int_{L_i}^{U_i} P_l^i(x)P_{l'}^i(x)w^i(x)dx = \delta_{l,l'}, \tag{2}$$

where the weight function w^i is defined on $[L_i, U_i]$ and takes a nonnegative value, and $\delta_{l,l'}$ is the Kronecker delta. However, we hereafter assume that $\{P_l^i\}_l$ satisfy the discrete orthogonal relation as follows: for any $D \in \mathbb{N}$, there exist n_{gr} points $x_{i,0} < x_{i,1} < \dots < x_{i,n_{\text{gr}}-1}$ in $[L_i, U_i]$, where $n_{\text{gr}} \geq D$, such that, for any $l, l' \in [D]_0$,

$$\sum_{j=0}^{n_{\text{gr}}-1} P_l^i(x_{i,j})P_{l'}^i(x_{i,j}) = c_l^i \delta_{l,l'} \tag{3}$$

holds for some $c_l^i > 0$. We can consider Eq. (3) as the discrete approximation of Eq. (2), with w^i absorbed into the spacing of the grid points $x_{i,j}$. For some orthogonal functions such as trigonometric functions and Chebyshev polynomials, the relationship in Eq. (3) holds strictly.

We define the tensorized orthogonal functions as follows: for any $\vec{l} = (l_1, \dots, l_d) \in \mathbb{N}_0^d$ and $\vec{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$,

$$P_{\vec{l}}(\vec{x}) := \prod_{i=1}^d P_{l_i}^i(x_i). \tag{4}$$

² For $n \in \mathbb{N}$, we define $[n] := \{1, \dots, n\}$.

It follows from Eq. (3) that, with $\vec{x}_{\vec{j}}$ defined as $\vec{x}_{\vec{j}} := (x_{1,j_1}, \dots, x_{d,j_d}) \in \mathbb{R}^d$ for any $\vec{j} = (j_1, \dots, j_d) \in [n_{\text{gr}}]_0^d$, they satisfy the orthogonal relation

$$\sum_{\vec{j} \in [n_{\text{gr}}]_0^d} P_{\vec{l}}(\vec{x}_{\vec{j}}) P_{\vec{l}'}(\vec{x}_{\vec{j}}) = c_{\vec{l}} \delta_{\vec{l}, \vec{l}'} \tag{5}$$

for any $\vec{l} = (l_1, \dots, l_d)$ and $\vec{l}' = (l'_1, \dots, l'_d)$ in $[D]_0^d$, where $\delta_{\vec{l}, \vec{l}'} := \prod_{i=1}^d \delta_{l_i, l'_i}$ and $c_{\vec{l}} := \prod_{i=1}^d c_{l_i}^i$.

We can use these $P_{\vec{l}}$ to approximate f by the orthogonal function series

$$\tilde{f}(\vec{x}) := \sum_{\vec{l} \in [D]_0^d} a_{\vec{l}} P_{\vec{l}}(\vec{x}) \tag{6}$$

with D such that $\max_{\vec{x}} |f(\vec{x}) - \tilde{f}(\vec{x})|$ is less than the desired threshold ϵ . Here, for any $\vec{l} \in [D]_0^d$, the coefficient $a_{\vec{l}}$ is given by

$$a_{\vec{l}} = \frac{1}{c_{\vec{l}}} \sum_{\vec{j} \in [n_{\text{gr}}]_0^d} f(\vec{x}_{\vec{j}}) P_{\vec{l}}(\vec{x}_{\vec{j}}), \tag{7}$$

which follows from Eq. (3). It is known that this kind of series converges to f in the large D limit for some orthogonal functions (see [57] for the trigonometric series and [58] for the Chebyshev series).

2.3 Oracles to generate the function-encoding states

In this study, we assume the availability of the oracle O_f mentioned in the introduction. We now define this formally. Hereafter, we assume that the grid number n_{gr} satisfies $n_{\text{gr}} = 2^{m_{\text{gr}}}$ with some integer m_{gr} . Then, we consider the system S consisting of $d m_{\text{gr}}$ -qubit registers and the unitary operator O_f that acts on S as

$$O_f |0\rangle^{\otimes d} = |f\rangle := \frac{1}{C} \sum_{\vec{j}=(j_1, \dots, j_d) \in [n_{\text{gr}}]_0^d} f(\vec{x}_{\vec{j}}) |j_1\rangle \cdots |j_d\rangle. \tag{8}$$

For any $n \in \mathbb{N}_0$, we denote by $|n\rangle$ the computational basis states on the quantum register with a sufficient number of qubits, in which the bit string on the register corresponds to the binary representation of n . $|j\rangle$ with $j \in [n_{\text{gr}}]_0$ in Eq. (8) is the computational basis state on an m_{gr} -qubit register. Furthermore, C in Eq. (8) is defined as

$$C := \sqrt{\sum_{\vec{j} \in [n_{\text{gr}}]_0^d} (f(\vec{x}_{\vec{j}}))^2}. \tag{9}$$

We also assume that the availability of the oracles $V_{\text{OF}}^1, \dots, V_{\text{OF}}^d$, each of which acts on an m_{gr} -qubit register as

$$V_{\text{OF}}^i |l\rangle = |P_l^i\rangle := \frac{1}{\sqrt{c_l^i}} \sum_{j=0}^{n_{\text{gr}}-1} P_l^i(x_{i,j}) |j\rangle \tag{10}$$

for any $l \in [D]_0, V_{\text{OF}}^i |D\rangle, \dots, V_{\text{OF}}^i |n_{\text{gr}} - 1\rangle$ may be any states as far as V_{OF}^i is unitary. In principle, these oracles are constructible because of the orthogonal relation (3). We discuss their implementation in ‘‘Appendix A.’’

Now, let us elaborate on the reason why reading out $f(\vec{x}_{\vec{j}})$ for $\vec{j} \in [n_{\text{gr}}]_0^d$ from $|f\rangle$ is difficult but obtaining it through orthogonal function expansion is more promising, as mentioned briefly in the introduction. We rewrite the amplitude in $|f\rangle$ as

$$\frac{f(\vec{x}_{\vec{j}})}{C} = \frac{1}{\sqrt{N_{\text{gr}}}} \frac{f(\vec{x}_{\vec{j}})}{\bar{f}}, \tag{11}$$

where $\bar{f} := \sqrt{\frac{1}{N_{\text{gr}}} \sum_{\vec{j} \in [n_{\text{gr}}]_0^d} (f(\vec{x}_{\vec{j}}))^2}$ is the root mean square of f over the grid points. The amplitude is suppressed by the factor $1/\sqrt{N_{\text{gr}}}$, which is exponential with respect to d , and thus exponentially small unless f is extremely localized at $\vec{x}_{\vec{j}}$ such that $f(\vec{x}_{\vec{j}})/\bar{f}$ is comparable to $\sqrt{N_{\text{gr}}}$. However, we note that, for any $\vec{l} \in [D]_0^d$, we have

$$\langle P_{\vec{l}}^i | f \rangle = \frac{1}{C \sqrt{c_{\vec{l}}^i}} \sum_{\vec{j} \in [n_{\text{gr}}]_0^d} f(\vec{x}_{\vec{j}}) P_{\vec{l}}^i(\vec{x}_{\vec{j}}) = \frac{\sqrt{c_{\vec{l}}^i}}{C} a_{\vec{l}}, \tag{12}$$

where $|P_{\vec{l}}^i\rangle := |P_{l_1}^1\rangle \cdots |P_{l_d}^d\rangle$. Because both C and $\sqrt{c_{\vec{l}}^i}$ are the root mean squares of some functions over the grid points, we expect that their ratio is $O(1)$ and that we can efficiently obtain the expansion coefficient $a_{\vec{l}}$ and then the approximation \tilde{f} of f in Eq. (6) by estimating $\langle P_{\vec{l}}^i | f \rangle$, without suffering from an exponential suppression factor such as $1/\sqrt{N_{\text{gr}}}$. However, we do not consider this direction in this study, because finding all the expansion coefficients suffers from an exponential increase in the number of coefficients in the high-dimensional case, as explained in the introduction.

2.4 Matrix product state

Thus, we consider using a tensor network. Among the various types of tensor networks, we use MPS, also known as the *tensor train*, which is simple but powerful and therefore widely used in various fields; a basic introduction of the MPS is pre-

sented in ‘‘Appendix B.’’ In MPS scheme, the order- d tensor $a_{\vec{l}} \in \mathbb{R}^{\overbrace{D \times \cdots \times D}^d}$ is

approximated by

$$\tilde{a}_{\vec{l}} := \sum_{k_1=1}^r \cdots \sum_{k_{d-2}=1}^r U_{l_1, k_1}^1 U_{k_1, l_2, k_2}^2 \cdots U_{k_{d-3}, l_{d-2}, k_{d-2}}^{d-2} U_{k_{d-2}, l_{d-1}, l_d}^{d-1}, \tag{13}$$

where $r \in \mathbb{N}$ is called the bond dimension, $U^1 \in \mathbb{R}^{D \times r}$, $U^i \in \mathbb{R}^{r \times D \times r}$ for $i \in \{2, \dots, d-2\}$, and $U^{d-1} \in \mathbb{R}^{r \times D \times D}$.³ We also impose the following conditions: for any $i \in \{2, \dots, d-2\}$ and $k_{i-1}, k'_i \in [r]$,

$$\sum_{l_i=0}^{D-1} \sum_{k_i=1}^r U_{k_{i-1}, l_i, k_i}^i U_{k'_{i-1}, l_i, k_i}^i = \delta_{k_{i-1}, k'_i} \tag{15}$$

holds, and, for any $k_{d-2}, k'_{d-2} \in [r]$,

$$\sum_{l_{d-1}=0}^{D-1} \sum_{l_d=0}^{D-1} U_{k_{d-2}, l_{d-1}, l_d}^{d-1} U_{k'_{d-2}, l_{d-1}, l_d}^{d-1} = \delta_{k_{d-2}, k'_{d-2}} \tag{16}$$

holds. We call this form of the MPS the right canonical form. This can always be imposed by the procedure explained in ‘‘Appendix B.’’

The representation in Eq. (13) actually reduces the DOF compared with the original $a_{\vec{l}}$ as a d -dimensional tensor. The total number of components in U^1, \dots, U^{d-1} is

$$rD + (d-3)r^2D + rD^2, \tag{17}$$

which is smaller than that in $a_{\vec{l}}$, D^d , unless $r = O(D^{O(d)})$. Furthermore, with the coefficients $a_{\vec{l}}$ in Eq. (6) represented as Eq. (13), the computation of the approximation of the function f becomes efficient. We now approximate f by

$$\begin{aligned} \tilde{f}(\vec{x}) &= \sum_{\vec{l} \in [D]_0^d} \tilde{a}_{\vec{l}} P_{\vec{l}}(\vec{x}) = \sum_{l_1=0}^{D-1} \cdots \sum_{l_d=0}^{D-1} \sum_{k_1=1}^r \cdots \sum_{k_{d-2}=1}^r U_{l_1, k_1}^1 U_{k_1, l_2, k_2}^2 \\ &\quad \cdots U_{k_{d-3}, l_{d-2}, k_{d-2}}^{d-2} U_{k_{d-2}, l_{d-1}, l_d}^{d-1} P_{l_1}^1(x_1) \cdots P_{l_d}^d(x_d). \end{aligned} \tag{18}$$

³ Usually, the MPS representation of a d -dimensional tensor takes the form

$$\tilde{a}_{\vec{l}} := \sum_{k_1=1}^r \cdots \sum_{k_{d-1}=1}^r U_{l_1, k_1}^1 U_{k_1, l_2, k_2}^2 \cdots U_{k_{d-2}, l_{d-1}, k_{d-1}}^{d-1} U_{k_{d-1}, l_d}^d, \tag{14}$$

where, in comparison to Eq. (13), $(U_{k_{d-2}, l_{d-1}, k_{d-1}}^{d-1})$ is in $\mathbb{R}^{r \times D \times r}$ rather than $\mathbb{R}^{r \times D \times D}$, and $(U_{k_{d-1}, l_d}^d) \in \mathbb{R}^{r \times D}$ is added. We can consider that U^{d-1} and U^d in Eq. (14) are contracted to U^{d-1} in Eq. (13). The reason for the form in Eq. (13) is that it corresponds to the quantum circuit considered in Sect. 2.5. In addition, although we can set the bond dimension r separately for each pair of U^i and U^{i+1} , for simplicity we set it to the same value.

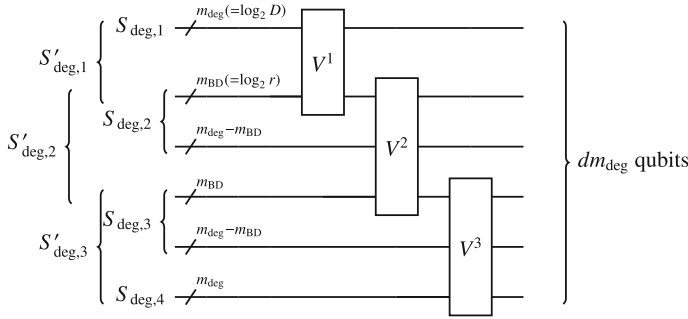


Fig. 2 Diagram of the quantum circuit V_{MPS} that generates the MPS-encoded state (21) in the case of $d = 4$. All the dm_{deg} qubits are initialized to $|0\rangle$. We describe the subsystems consisting of a portion of the qubits at the left end: $S_{deg,i}$ has m_{deg} qubits, the $((i - 1)m_{deg} + 1)$ -th to im_{deg} -th ones, and $S'_{deg,i}$ has $m_{deg} + m_{BD}$ qubits, the $((i - 1)m_{deg} + 1)$ -th to $(im_{deg} + m_{BD})$ -th ones, except for $S'_{deg,d-1}$ having the last $2m_{deg}$ qubits

This can be computed as

$$\begin{aligned}
 \tilde{f}(\vec{x}) &= \sum_{k_1=1}^r \cdots \sum_{k_{d-2}=1}^r \left(\sum_{l_1=0}^{D-1} U_{l_1,k_1}^1 P_{l_1}^1(x_1) \right) \times \left(\sum_{l_2=0}^{D-1} U_{k_1,l_2,k_2}^2 P_{l_2}^2(x_2) \right) \\
 &\times \cdots \times \left(\sum_{l_{d-2}=0}^{D-1} U_{k_{d-3},l_{d-2},k_{d-2}}^{d-2} P_{l_{d-2}}^{d-2}(x_{d-2}) \right) \\
 &\times \left(\sum_{l_{d-1},l_d=0}^{D-1} U_{k_{d-2},l_{d-1},l_d}^{d-1} P_{l_{d-1}}^{d-1}(x_{d-1}) P_{l_d}^d(x_d) \right), \tag{19}
 \end{aligned}$$

that is, we first contract U_{k_{i-1},l_i,k_i}^i and $P_{l_i}^i$ with respect to l_i for each $i \in [d]$, and then take contractions with respect to k_1, \dots, k_{d-2} . In this procedure, the number of arithmetic operations is

$$O(dr^2D + rD^2), \tag{20}$$

which is much smaller than $O(D^d)$ for computing (6) with general a_i not having any specific structure.

2.5 Quantum circuit to generate the tensor network state

The quantum circuit to generate an MPS-encoded quantum state is shown in Fig. 2. First, we prepare dm_{deg} qubits initialized to $|0\rangle$, where we assume that $D = 2^{m_{deg}}$ holds with some $m_{deg} \in \mathbb{N}$. Labeling them with the integers $1, \dots, dm_{deg}$, for each $i \in [d]$, we denote the system of the $((i - 1)m_{deg} + 1)$ -th to im_{deg} -th qubits by $S_{deg,i}$. In addition, assuming that $r = 2^{m_{BD}}$ also holds with some $m_{BD} \in \mathbb{N}$, for each

$i \in [d - 2]$, we denote the system of $S_{\text{deg},i}$ and the first m_{BD} qubits in $S_{\text{deg},i+1}$ by $S'_{\text{deg},i}$, and the system of the last $2m_{\text{deg}}$ qubits by $S'_{\text{deg},d-1}$. Then, we put the quantum gates V_1, \dots, V_{d-1} on $S'_{\text{deg},1}, \dots, S'_{\text{deg},d-1}$, respectively, in this order.

Let us denote by V_{MPS} the unitary that corresponds to the whole of the above quantum circuit, which is depicted in Fig. 2. Then, V_{MPS} generates the MPS-encoded state

$$V_{\text{MPS}}|0\rangle^{\otimes d} = |\tilde{a}\rangle := \sum_{l_1=0}^{D-1} \cdots \sum_{l_d=0}^{D-1} \sum_{k_1=1}^r \cdots \sum_{k_{d-2}=1}^r U_{l_1,k_1}^1 U_{k_1,l_2,k_2}^2 \cdots U_{k_{d-3},l_{d-2},k_{d-2}}^{d-2} U_{k_{d-2},l_{d-1},l_d}^{d-1} |l_1\rangle \cdots |l_d\rangle, \tag{21}$$

which is close to the state

$$|a\rangle := \sqrt{\frac{1}{\sum_{\tilde{l} \in [D]_0^d} |a_{\tilde{l}}|^2}} \sum_{\tilde{l} \in [D]_0^d} a_{\tilde{l}} |l_1\rangle \cdots |l_d\rangle \tag{22}$$

that encodes the true coefficients $a_{\tilde{l}}$ if $\tilde{a}_{\tilde{l}}$ in Eq. (13) approximates $a_{\tilde{l}}$. Here, $|l\rangle$ with $l \in [D]_0$ denotes the computational basis states on $S_{\text{deg},1}, \dots, S_{\text{deg},d}$. In addition, we associate the entries in U^1, \dots, U^{d-1} with those in V^1, \dots, V^{d-1} as unitaries, respectively, as follows:

$$U_{l_1,k_1}^1 = \langle l_1 + Dk_1 | V^1 | 0 \rangle \tag{23}$$

for any $l_1 \in [D]_0$ and $k_1 \in [r]$,

$$U_{k_{i-1},l_i,k_i}^i = \langle l_i + Dk_i | V^i | k_{i-1} \rangle \tag{24}$$

for any $i \in \{2, \dots, d - 2\}$, $l_i \in [D]_0$ and $k_{i-1}, k_i \in [r]$, and

$$U_{k_{d-2},l_{d-1},l_d}^{d-1} = \langle l_{d-1} + Dl_d | V^{d-1} | k_{d-2} \rangle \tag{25}$$

for any $l_{d-1}, l_d \in [D]_0$ and $k_{d-2} \in [r]$, where $|n\rangle$ with $n \in \mathbb{N}_0$ denotes the computational basis state on either $S'_{\text{deg},1}, \dots, S'_{\text{deg},d-1}$. Note that each U^i , which is not a unitary matrix, is realized as a block in V^i , a component unitary in the circuit V_{MPS} in Fig. 2. Although V^1, \dots, V^{d-1} have additional components that do not appear in Eqs. (23) to (25), those do not affect the state $|\tilde{a}\rangle$ because of the initialization of all qubits to $|0\rangle$.

Note that U^2, \dots, U^{d-1} in Eqs. (24) and (25) automatically satisfy the conditions (15) and (16) because of the unitarity of V^2, \dots, V^{d-1} . However, the unitarity of V^1 imposes the constraint

$$\sum_{l_1=0}^{D-1} \sum_{k_1=1}^r |U_{l_1,k_1}^1|^2 = 1 \tag{26}$$

on U^1 in Eq. (23). This implies that, with such U^1 , the MPS-based approximation \tilde{f} in Eq. (18) does not have the DOF of the overall factor, and therefore, although we can express the functional form of f by \tilde{f} , we cannot adjust the magnitude of \tilde{f} so that it fits f . Conversely, if we have some estimate C for the ratio of f to \tilde{f} , we can approximate f by $C\tilde{f}$. This issue is addressed in Sect. 2.6.

2.6 Optimization of the quantum circuit

We now consider how to optimize the quantum circuit V_{MPS} and obtain an MPS-based function approximation.

First, we extend the circuit V_{MPS} in Sect. 2.5 using $\{V_{\text{OF}}^i\}_i$ as follows. For each $i \in [d]$, we add $m_{\text{gr}} - m_{\text{deg}}$ qubits after the im_{deg} -th qubits in the original circuit and denote the system consisting of $S_{\text{deg},i}$ and the added qubits by $S_{\text{gr},i}$. Note that the resultant system is the same as the system S for O_f , which consists of the d m_{gr} -qubit registers. We then perform V_{OF}^i on $S_{\text{deg},i}$. The resultant circuit V_{App} is shown in Fig. 3. Note that this circuit generates

$$\begin{aligned}
 V_{\text{App}}|0\rangle^{\otimes d} = |\tilde{f}\rangle := & \sum_{l_1=0}^{D-1} \cdots \sum_{l_d=0}^{D-1} \sum_{k_1=1}^r \cdots \sum_{k_{d-2}=1}^r \sum_{j_1=0}^{n_{\text{gr}}-1} \cdots \sum_{j_d=0}^{n_{\text{gr}}-1} U_{l_1,k_1}^1 U_{k_1,l_2,k_2}^2 \\
 & \cdots U_{k_{d-3},l_{d-2},k_{d-2}}^{d-2} U_{k_{d-2},l_{d-1},l_d}^{d-1} P_{l_1}^1(x_{1,j_1}) \cdots P_{l_d}^d(x_{d,j_d}) |j_1\rangle \cdots |j_d\rangle,
 \end{aligned}
 \tag{27}$$

where $|n\rangle$ with $n \in \mathbb{N}_0$ now denotes the computational basis state on $S_{\text{gr},1}, \dots, S_{\text{gr},d}$. That is, this is the quantum state that amplitude-encodes \tilde{f} in Eq. (18), the approximation of f by the orthogonal function expansion and the MPS approximation of the coefficients. Therefore, if we obtain V_{App} that generates $|\tilde{f}\rangle$ close to $|f\rangle$, we also obtain $\{U^i\}_i$ for which \tilde{f} in Eq. (18) well approximate f at least on the grid points, by reading out their entries from the quantum gates $\{V^i\}_i$ in V_{App} , except for the overall factor C .

Next, we consider how to obtain such V_{App} , especially $\{V^i\}_i$ in it. We aim to maximize the fidelity

$$F = \langle f | \tilde{f} \rangle.
 \tag{28}$$

Note that maximizing F is equivalent to minimizing the sum of the squared differences between the two normalized functions

$$\sum_{\vec{j} \in [n_{\text{gr}}]_0^d} \left(\frac{f(\vec{x}_{\vec{j}})}{C} - \tilde{f}(\vec{x}_{\vec{j}}) \right)^2,
 \tag{29}$$

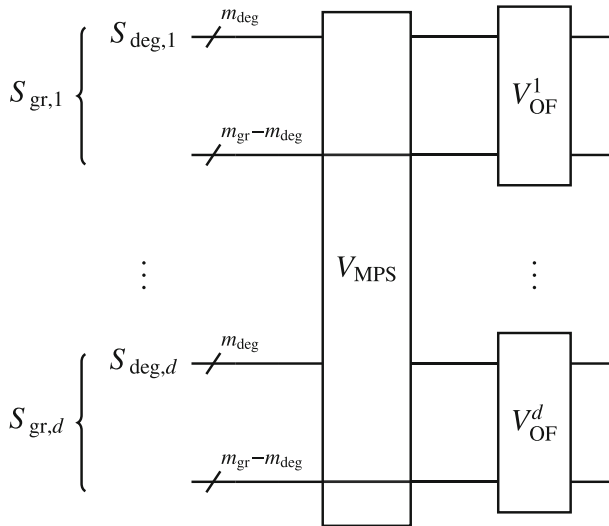


Fig. 3 Diagram of the quantum circuit V_{App} that generates the state (27) in which the approximation \tilde{f} in Eq. (18) of f is amplitude-encoded. V_{MPS} is the circuit depicted in Fig. 2; the lines that bypass it are not used in it. All the dm_{gr} qubits are initialized to $|0\rangle$. The subsystems consisting of a portion of the qubits, which are indicated at the left end, are described in the body text

which, in the large n_{gr} limit, is equivalent to

$$\int_{\Omega} \left(\frac{f(\vec{x}_{\vec{j}})}{C} - \tilde{f}(\vec{x}_{\vec{j}}) \right)^2 w^1(x_1) \cdots w^d(x_d) d\vec{x}, \tag{30}$$

that is, the squared L^2 norm of $\frac{f}{C} - \tilde{f}$, the common metric in function approximation.

The procedure for maximizing F is similar to that presented in [56]. We attempt to optimize each V^i alternately. In other words, we optimize each V^i with the others fixed, setting i to $1, 2, \dots, d - 1$ in turn. This loop can be repeated an arbitrary number of times. The optimization step for V^i proceeds as follows. We define

$$F_i = \text{Tr}_{\tilde{S}'_{\text{deg},i}} [|\Psi_{i+1}\rangle\langle\Phi_{i-1}|]. \tag{31}$$

Here, $\tilde{S}'_{\text{deg},i}$ denotes the system consisting of the qubits except those in $S'_{\text{deg},i}$, and, for any subsystem s in S , Tr_s denotes the partial trace over the Hilbert space corresponding to s . $|\Psi_{i+1}\rangle$ is defined as

$$|\Psi_{i+1}\rangle = \begin{cases} \left(V^{i+1} \otimes I_{\tilde{S}'_{\text{deg},i+1}} \right)^\dagger \cdots \left(V^{d-1} \otimes I_{\tilde{S}'_{\text{deg},d-1}} \right)^\dagger (V_{\text{OF}}^1 \otimes \cdots \otimes V_{\text{OF}}^{d-1})^\dagger |f\rangle & ; \text{ for } i = 1, \dots, d - 2 \\ (V_{\text{OF}}^1 \otimes \cdots \otimes V_{\text{OF}}^{d-1})^\dagger |f\rangle & ; \text{ for } i = d - 1 \end{cases} \tag{32}$$

where $I_{\bar{S}'_{\text{deg},i}}$ denotes the identity operator on $\bar{S}'_{\text{deg},i}$, and thus $V^i \otimes I_{\bar{S}'_{\text{deg},i}}$ denotes the i -th block in V_{MPS} in Fig. 2. $|\Phi_{i-1}\rangle$ is defined as

$$|\Phi_{i-1}\rangle = \begin{cases} \left(V^{i-1} \otimes I_{\bar{S}'_{\text{deg},i-1}} \right) \cdots \left(V^1 \otimes I_{\bar{S}'_{\text{deg},1}} \right) |0\rangle^{\otimes d} & ; \text{ for } i = 2, \dots, d - 1 \\ |0\rangle^{\otimes d} & \text{for } i = 1 \end{cases} \tag{33}$$

We can regard this F_i as an $M \times M$ matrix, where $M = rD$ for $i \in [d-2]$ and $M = D^2$ for $i = d - 1$. Its entries are given by $\langle l|F_i|l'\rangle$, where $|l\rangle, |l'\rangle \in \{|0\rangle, |1\rangle, \dots, |M-1\rangle\}$ are now the computational basis states on $S'_{\text{deg},i}$. Supposing that we know F_i , we perform its singular-value decomposition (SVD)

$$F_i = XDY, \tag{34}$$

where X and Y are the $M \times M$ unitaries and D is the diagonal matrix having the singular values of F_i as its diagonal entries. Finally, we update V^i by

$$V^i = XY. \tag{35}$$

We obtain F_i through Pauli decomposition and the Hadamard test [56]. We set $m = \log_2 M$, and, for any $k \in [m]$, we denote the identity operator, the Pauli-X gate, the Pauli-Y gate, and the Pauli-Z gate on the k -th qubit in $S'_{\text{deg},i}$ by $\hat{\sigma}_0^k, \hat{\sigma}_1^k, \hat{\sigma}_2^k$ and $\hat{\sigma}_3^k$, respectively. We also define

$$\hat{\sigma}_{\vec{\alpha}} := \hat{\sigma}_{\alpha_1}^1 \otimes \cdots \otimes \hat{\sigma}_{\alpha_m}^m \tag{36}$$

for any $\vec{\alpha} = (\alpha_1, \dots, \alpha_m) \in \{0, 1, 2, 3\}^m$. Then, we can always decompose F_i as

$$F_i = \sum_{\vec{\alpha} \in \mathcal{A}} \tilde{F}_{i,\vec{\alpha}} \hat{\sigma}_{\vec{\alpha}}, \tag{37}$$

with $\tilde{F}_{i,\vec{\alpha}} \in \mathbb{R}$, where, rather than $\{0, 1, 2, 3\}^m$, the index tuple $\vec{\alpha}$ runs over the subset

$$\mathcal{A} := \{ \vec{\alpha} \in \{0, 1, 2, 3\}^m \mid \text{the number of 2's in } \alpha_1, \dots, \alpha_m \text{ is even.} \} \tag{38}$$

because F_i is real. Because

$$\text{Tr}_{S'_{\text{deg},i}} [F_i \hat{\sigma}_{\vec{\alpha}}] = M \tilde{F}_{i,\vec{\alpha}} \tag{39}$$

holds, we obtain $\tilde{F}_{i,\vec{\alpha}}$ by estimating $\text{Tr}_{S'_{\text{deg},i}} [F_i \hat{\sigma}_{\vec{\alpha}}]$. Noting that

$$\text{Tr}_{S'_{\text{deg},i}} [F_i \hat{\sigma}_{\vec{\alpha}}] = \langle \Phi_{i-1} | \hat{\sigma}_{\vec{\alpha}} | \Psi_{i+1} \rangle, \tag{40}$$

we can estimate this using the Hadamard test, in which the circuit V_{MPS} with replacement of V^i by $\hat{\sigma}_{\vec{\alpha}}$ is used. In one update of each V^i , the total number of estimations of the quantities (40) is $|\mathcal{A}| = M(M + 1)/2$. We refer to [56] for the details of the estimations.

After we optimize the $\{V^i\}_i$ and read out $\{U^i\}_i$ from them, the remaining task is just multiplying the factor C to \tilde{f} constructed from $\{U^i\}_i$. In this paper, we do not go into the details of estimating this factor but simply assume that we have some estimate for it. In some quantum algorithms for solving PDEs, the method of estimating this factor, the root of the squared sum of the function values on the grid points, is presented [19].

The entire procedure to obtain an approximation of f is summarized as Algorithm 1.

Algorithm 1 Proposed algorithm to obtain an approximation of the function $f : \Omega \rightarrow \mathbb{R}$.

Input:

- 1:
 - $D = 2^{m_{\text{deg}}}$ with $m_{\text{deg}} \in \mathbb{N}$: the degree of the orthogonal functions
 - $r = 2^{m_{\text{BD}}}$ with $m_{\text{BD}} \in \mathbb{N}$: the bond dimension
 - The oracle O_f in Eq. (8).
 - The orthogonal functions $\{P_l^i\}_{i \in [d], l \in [D]_0}$ satisfying Eq. (3).
 - The iteration $n_{\text{iter}} \in \mathbb{N}$ of the optimization loop.
 - The initial values of the $rD \times rD$ unitaries V^1, \dots, V^{d-2} and the $D^2 \times D^2$ unitary V^{d-1} , which may be chosen randomly.
 - C in Eq. (9).
 - 2: **for** $i_{\text{iter}} = 1$ to n_{iter} **do**
 - 3: **for** $i = 1$ to $d - 1$ **do**
 - 4: **for** $\vec{\alpha} \in \{0, 1, 2, 3\}^m$ **do**
 - 5: Estimate $\langle \Phi_{i-1} | \hat{\sigma}_{\vec{\alpha}} | \Psi_{i+1} \rangle$ using the Hadamard test, and divide it by M to obtain $\tilde{F}_{\vec{\alpha}}$.
 - 6: **end for**
 - 7: Set F_i as per Eq. (37).
 - 8: Perform the SVD of F_i as per Eq. (34).
 - 9: Update V^i as per Eq. (35).
 - 10: **end for**
 - 11: **end for**
 - 12: Set $U^1 \in \mathbb{R}^{D \times r}$, $U^i \in \mathbb{R}^{r \times D \times r}$ for $i \in \{2, \dots, d - 2\}$, and $U^{d-1} \in \mathbb{R}^{r \times D \times D}$ as per Eqs. (23), (24) and (25), respectively.
 - 13: Define \tilde{f} as per Eq. (18).
 - 14: Output $C\tilde{f}$ as an approximation of f .
-

2.7 Situation in which our method is useful

In the introduction, we mentioned solving PDEs using quantum algorithms as a main use case of our method. One might think that if the solution of the PDE can be approximated by MPS as Eq. (19), we can directly plug the ansatz (19) into the PDE and optimize the parameters $\{U_{k_{i-1}, l_i, k_i}^i\}$ using a classical computer. In fact, such a method has been studied in [59–62]. However, the scheme we are now considering, that is, generating the quantum state that encodes the solution in the amplitude by a quantum algorithm and reading out the solution in MPS-based approximation, is applicable to the broader range of problems. For example, in the initial value problem, it is possible that MPS-based approximation is valid at the terminal time but not at

some intermediate time points. In the context of quantum many-body physics, this is applicable to the wave function of a system that changes to a low-entangled state via a highly entangled state, such as quantum annealing [63–65]. In such a case, finding the function at the terminal time using a quantum PDE solver and then reading it out by MPS-based approximation might work, although using MPS throughout does not seem to work.

3 Numerical experiment

We now confirm the feasibility of our method through a numerical experiment on approximating a finance-motivated function.

3.1 Problem setting

As a reasonable instance of the target function for the approximation, we take the price of a *financial derivative* (simply, *derivative*).

Specifically, we consider the worst-of put option written on the d underlying assets that obey the Black–Scholes (BS) model and take its present price as a function $f(\vec{s})$ of the asset prices $\vec{s} = (s_1, \dots, s_d)$ as the approximation target. This is the solution of the so-called Black–Scholes PDE and thus fits the current situation in which a quantum PDE solver outputs the state $|f(\vec{s})\rangle$. Details are provided in “Appendix C.”

We approximated $f(\vec{s})$ on the hyperrectangle $[L_1, U_1] \times \dots \times [L_d, U_d]$. We took cosine functions as orthogonal functions. Specifically, for any $i \in [d]$ and $l \in [D]_0$, we set

$$P_l^i(s_i) = \cos\left(l \frac{s_i - L_i}{U_i - L_i} \pi\right). \tag{41}$$

The settings for U_i and L_i are presented in “Appendix C.” The orthogonal relation (3) is satisfied with the grid points set as

$$s_{i,j} = \frac{j + \frac{1}{2}}{n_{\text{gr}}}(U_i - L_i) + L_i \tag{42}$$

for $j \in [n_{\text{gr}}]_0$ and c_l^j being

$$c_l^j = \begin{cases} n_{\text{gr}}; & \text{for } l = 0 \\ \frac{n_{\text{gr}}}{2}; & \text{for } l \in [D - 1] \end{cases}. \tag{43}$$

Let us comment on the choice of cosines as orthogonal functions. We consider the case where the grid points are equally spaced points in the hyperrectangle, which we expect to be the simplest and most common. In this setting, the cosine functions (41) satisfy the orthogonal relation (5) exactly. Therefore, the choice was natural in this

case. Generally, it is plausible to take orthogonal functions such that the orthogonal relation holds for the given grid points.

3.2 Algorithm modifications to run the numerical experiment

We used Algorithm 1 but made the following modification, since we performed all calculations on a classical computer, and thus there was a memory space limitation. Rather than F in Eq. (28), we attempt to maximize

$$F' = \langle a | \tilde{a} \rangle \tag{44}$$

Here, $|\tilde{a}\rangle$ is given in Eq. (21) and

$$|a\rangle := \sqrt{\frac{1}{\sum_{\vec{l} \in [D]_0^d} |a_{\vec{l}}|^2}} \sum_{\vec{l} \in [D]_0^d} a_{\vec{l}} |l_1\rangle \cdots |l_d\rangle, \tag{45}$$

where we calculated the coefficients $a_{\vec{l}}$ for each $\vec{l} \in [D]_0^d$ using Eq. (7) and the values of $f(\vec{s})$ on the grid points computed by Monte Carlo integration (see ‘‘Appendix C’’). We take the minimal grid points to calculate the coefficients for a given D , which means $n_{\text{gr}} = D$, although it is assumed that $n_{\text{gr}} > D$ when our algorithm is used on a future quantum computer with the oracle O_f .

With this modification, the quantum circuit under consideration becomes small, and the calculation becomes feasible on a classical computer. The gates $\{V_{\text{OF}}^i\}$ do not appear in our experiment, and their roles are absorbed into the aforementioned preprocessing to calculate $\{a_{\vec{l}}\}$. Most importantly, note that $\{U^i\}$ calculated under the above modification are the same as those output by our algorithm without modification.

3.3 Result of the approximation

Next, we attempted to obtain an approximation of $f(\vec{s})$. We set the parameters as follows: $d = 5$, $D = r = 16$, $n_{\text{iter}} = 5$. For C , we used the root squared sum of the values of $f(\vec{s})$ on the grid points, which are calculated using Monte Carlo integration. We used the ITensor library [66] for the tensor calculations.

We now demonstrate the accuracy of the obtained MPS-based approximation of $f(\vec{s})$. However, because we are considering a high-dimensional space, it is difficult to display the accuracy over the entire space. Therefore, we show the accuracy on the following two sets of selected points: (i) the ‘‘diagonal’’ line $s_1 = \cdots = s_d$ and (ii) 10^4 random points sampled according to the BS model (see ‘‘Appendix C’’ for details).

Figure 4 displays results for the diagonal line, depicting the MPS-based approximation $\tilde{f}(\vec{s})$ of $f(\vec{s})$ along with Monte Carlo integration values and the cosine expansion approximation as per Eq. (6). The MPS-based approximation fits the Monte Carlo integration values well within an error of about 0.5 or less, except for the region near the lower end of the domain of the approximations. Note that the cosine expansion approximation already has an error from the Monte Carlo values, and this acts as a

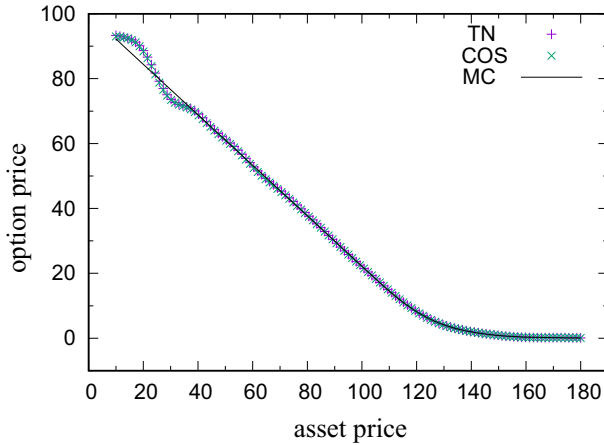


Fig. 4 Five-asset worst-of put option prices $f(\vec{s})$ for $s_1 = \dots = s_5 = s$ calculated in three different ways. The horizontal and vertical axes, respectively, correspond to s and $f(\vec{s})$. The +, ×, and black line, respectively, indicate results obtained from MPS-based approximation, cosine expansion approximation, and Monte Carlo integration

Table 1 Maximum differences among the values of the five-asset worst-of put option prices $f(\vec{s})$ calculated by Monte Carlo integration (MC), MPS-based approximation (TN), and cosine expansion using the full coefficients (COS), on the random sample points

$\max TN - MC $	0.5086
$\max COS - MC $	0.5102
$\max TN - COS $	0.3164

lower bound on the error of the MPS-based approximation. The MPS-based approximation almost overlaps the cosine expansion approximation, which means that the MPS-based approximation worked well.

The results for the random sample points are summarized in Table 1. On these points, the maximum differences among the values of $f(\vec{s})$ calculated by Monte Carlo integration, MPS-based approximation, and cosine expansion using the full coefficients without MPS approximation are about 0.5 or less, similar to most regions in the diagonal line case. This result supports the proposed method.

In the current MPS-based approximation, the number of DOF is 12544, which is smaller than the number of cosine expansion coefficients (1048576) by two orders of magnitude. Therefore, we achieved a large parameter reduction while maintaining the approximation accuracy.

3.4 Relationship between approximation accuracy and degrees of freedom

To investigate the relationship between the DOF and the approximation accuracy, we performed the following additional experiment. We replaced the circuit V_{MPS} in Fig. 2 by a different V'_{MPS} having the different DOF. Here, with $m_{bl} \in \{2, \dots, dm_{deg} -$

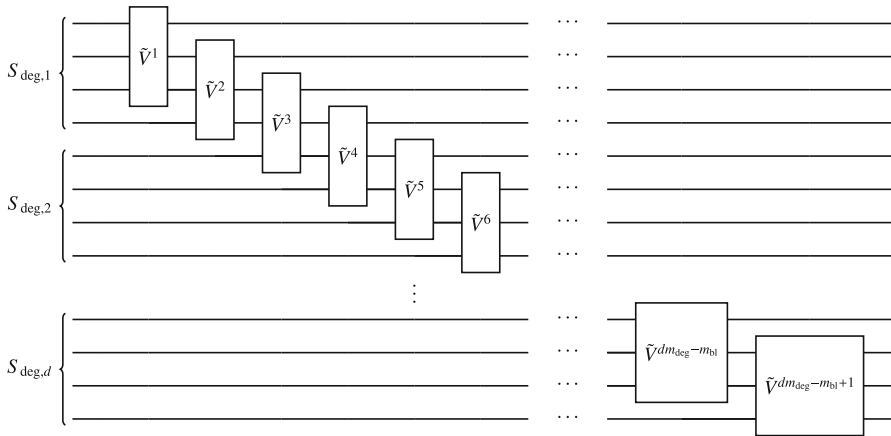


Fig. 5 Diagram of the quantum circuit V'_{MPS} used in the additional numerical experiment instead of V_{MPS} for $m_{\text{deg}} = 4$ and $m_{\text{bl}} = 3$

1}, V'_{MPS} consists of the gates $\tilde{V}^1, \dots, \tilde{V}^{dm_{\text{deg}}-m_{\text{bl}}+1}$ that act on the systems of m_{bl} qubits displaced one by one, as shown in Fig. 5. The DOF of this circuit is

$$2^{m_{\text{bl}}} + 2^{2m_{\text{bl}}-1} (dm_{\text{deg}} - m_{\text{bl}}). \tag{46}$$

We alternately optimized the blocks $\{\tilde{V}^i\}$ in a manner similar to Algorithm 1 so that $\langle a|\tilde{a}'\rangle$ is maximized, where

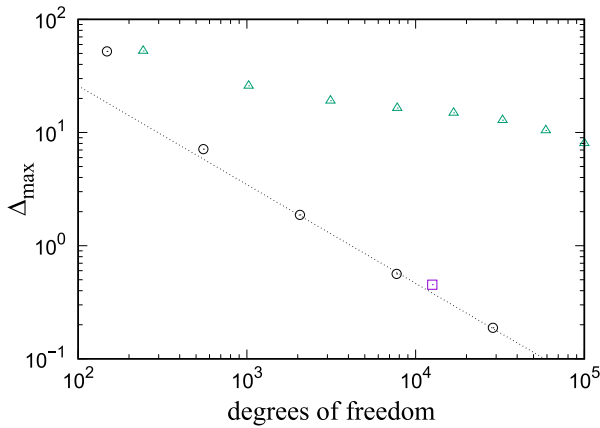
$$|\tilde{a}'\rangle := \sum_{\vec{l} \in [D]_0^d} \tilde{a}'_l |l_1\rangle \cdots |l_d\rangle \tag{47}$$

is the state generated by V'_{MPS} . Then, we obtained the approximation of $f(\vec{s})$ as

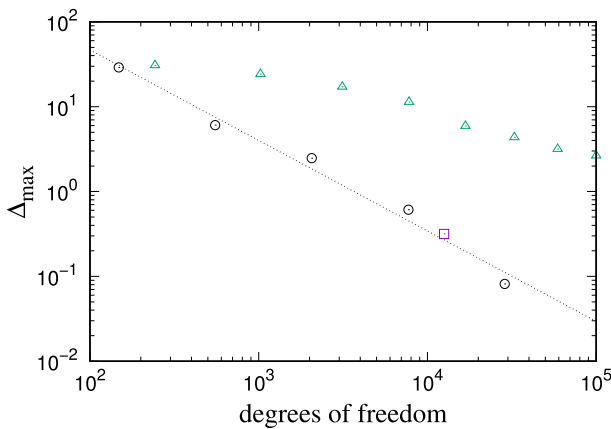
$$\tilde{f}'(\vec{s}) := C \sum_{\vec{l} \in [D]_0^d} \tilde{a}'_l P_l(\vec{s}). \tag{48}$$

In Fig. 6a, we display the maximum difference of \tilde{f}' from the cosine expansion approximation on the diagonal line $s_1 = \dots = s_d$, taking $m_{\text{bl}} = 2, \dots, 6$, along with that of the MPS-based approximation. This figure shows that there is a power law relationship between the approximation accuracy and DOF in the region with large DOF. This behavior is similar to that often observed in critical MPS systems applied to one-dimensional quantum systems or two-dimensional classical systems [67–73]. If this behavior also appears in the case of $d \gg 1$, one might make a similar argument to the finite bond dimension (entanglement) scaling refined in the study of MPS and evaluate \tilde{f} with appropriate extrapolations with respect to DOF.

We then show the relationship between the accuracy of the cosine expansion approximation and its DOF in Fig. 6a. We plot the maximum error of the approximated



(a) On the diagonal line. $a = e^{7.27}$ and $b = -0.87$.



(b) On the random sample points. $a = e^{8.75}$ and $b = -1.07$.

Fig. 6 Maximum difference Δ_{\max} of the various approximations for the five-asset worst-of put option price from the cosine expansion approximation of degree $D = 16$. The circles indicate the approximations based on the circuit V'_{MPS} in Fig. 5. From left to right, the points correspond to $m_{\text{bl}} = 2, \dots, 6$, respectively. The dotted lines represent the function $\Delta_{\max}(x) = ax^b$ of the degrees of freedom x with a and b fitted with respect to the data for $m_{\text{bl}} = [4, 6]$ for (a) and $m_{\text{bl}} = [2, 6]$ for (b). The square indicates the maximum difference of the MPS-based approximation, and the triangles indicate those of the cosine expansion approximations of degree $D = 3, \dots, 10$, which correspond to the points from left to right, respectively. The vertical and horizontal lines correspond to the maximum difference and DOF of the approximations, respectively

worst-of put option price by the cosine expansions of low degree $D = 3, \dots, 10$ on the line $s_1 = \dots = s_d$, taking that of degree $D = 16$ as the reference value. The error of the low-degree cosine expansion is much larger than that of \tilde{f}' with comparable DOF. Figure 6b is similar to Fig. 6a but for the random sample points. Again, the maximum difference of \tilde{f}' from the cosine expansion with $D = 16$ displays a power

law with respect to DOF and is much smaller than the error of the low-degree cosine expansion.

These results provide additional evidence of the advantages of our MPS-based approximation and the approximation by the circuit V'_{MPS} over the simple cosine expansion.

4 Summary

In this study, we have considered how to extract the function encoded in the amplitudes of the quantum state as classical data. Such a task necessarily accompanies quantum algorithms such as PDE solvers, but to date there has not been a proposal for a general method to accomplish this task, even though it risks ruining quantum speedup. We proposed a method based on orthogonal function expansion and tensor network. Orthogonal function expansion is widely used for function approximation but suffers from an exponential increase in the number of the parameters, the expansion coefficients, with respect to the dimension, that is, the number of variables of the function. We then use an MPS, a type of tensor network, to approximate the coefficients as a high-order tensor and reduce the DOF. We presented a quantum circuit that produces the state corresponding to such a function approximation. Such a circuit is, in fact, constructible because an MPS is encoded in a quantum state by a simple circuit, and so are orthogonal functions because of their orthogonal relation. We also presented the procedure to optimize the quantum circuit and MPS-based approximation, based on the alternating method proposed in [56]. Finally, we conducted a numerical experiment to approximate a finance-motivated multivariate function and found that our method works in this case.

This study has scope for further investigation. For example, we did not explicitly present the bound on the total query complexity for the oracles O_f and V'_{OF} to obtain an approximating function with a given accuracy ϵ . Presenting such a complexity estimation is desired but difficult at present. In general, function approximation in a high-dimensional space is a long-standing problem in computational mathematics. Although a recent study showed that there is an MPS-based approximation that achieves a given accuracy with DOF subexponential with respect to d for any function with some property [54], to the best of the authors' knowledge, there is no known algorithm that certainly outputs such an approximation. The convergence property of the alternating optimization method explained above is not well known theoretically, even though its good performance is empirically known. In these situations, we have considered a quantitative discussion on the accuracy and complexity of our method beyond the scope of this study and left it for future work, presenting some numerical evidence instead.

We expect that the proposed method can be used widely in combination with various quantum algorithms that output function-encoding states, whether for FTQC or NISQ. In future work, we will attempt to combine this method with concrete function-finding quantum algorithms on concrete problems and present a complete set of the algorithms, along with a more quantitative analysis of complexity.

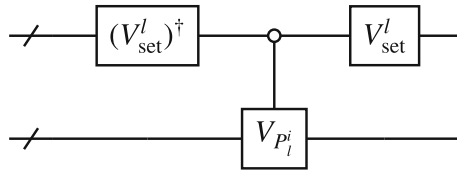
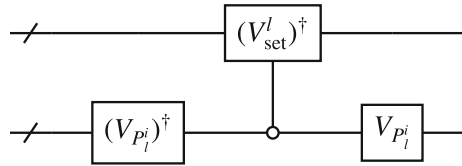


Fig. 7 Implementation of $V_{P_l^i}^l$ in Eq. (A3). The open circle represents the control on the gate $V_{P_l^i}$ such that it is activated if and only if all the qubits in the first register take $|0\rangle$

Fig. 8 Implementation of $V_{reset}^{i,l}$ in Eq. (A6)



Acknowledgements This work was partially supported by KAKENHI Grant Numbers JP22K11924, JP21H04446, JP21H05182, and JP21H05191 from JSPS of Japan and also by JST PRESTO No. JPMJPR1911, MEXT Q-LEAP Grant No. JPMXS0120319794, and JST COI-NEXT No. JPMJPF2014. H.U was supported by a COE research grant in computational science from Hyogo Prefecture and Kobe City through the Foundation for Computational Science.

Funding Open access funding provided by Osaka University.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: How to construct V_{OF}^i

To construct V_{OF}^i as per Eq. (10), we require some building block quantum gates. First, for any $i \in [d]$ and $l \in [D]_0$, we assume the availability of the gate $V_{P_l^i}$ that acts on an m_{gr} -qubit register as

$$V_{P_l^i} |0\rangle = |P_l^i\rangle, \tag{A1}$$

where $|P_l^i\rangle$ is given in Eq. (10). In fact, commonly used orthogonal functions such as trigonometric functions and orthogonal polynomials are explicitly given as elementary functions, and thus the state generation oracle as in Eq. (A1) is constructed using the Grover-Rudolph method [74, 75]. Second, for any $n \in \mathbb{N}_0$, we denote by V_{set}^n the gate that acts on a quantum register with a sufficient number of qubits as

$$V_{\text{set}}^n |0\rangle = |n\rangle. \tag{A2}$$

This can be constructed by putting a Pauli-X gate (resp. nothing) on the a -th qubit in the register if the binary representation of n contains a 1 (resp. 0) in the a -th digit.

Then, for any $i \in [d]$ and $l \in [D]_0$, using the controlled $V_{P_l^i}$ and V_{set}^l , we can implement the following gate on the system of two m_{gr} -qubit registers

$$V'_{P_l^i} := |l\rangle\langle l| \otimes V_{P_l^i} + \sum_{l' \in [n_{\text{gr}}]_0 \setminus \{l\}} |l'\rangle\langle l'| \otimes I, \tag{A3}$$

where I is the identity operator, as shown in Fig. 7. Combining gates of this type, we obtain

$$V_{\text{OF1}}^i := \prod_{l=0}^{D-1} V'_{P_l^i} = \sum_{l=0}^{D-1} |l\rangle\langle l| \otimes V_{P_l^i} + \sum_{l=D}^{n_{\text{gr}}-1} |l\rangle\langle l| \otimes I, \tag{A4}$$

which acts as

$$V_{\text{OF1}}^i |l\rangle|0\rangle = |l\rangle|P_l^i\rangle \tag{A5}$$

for any $l \in [D]_0$.

In addition, for any $i \in [d]$ and $l \in [D]_0$, using the controlled V_{set}^l and $V_{P_l^i}$, we construct the following gate on the system of two m_{gr} -qubit registers

$$V_{\text{reset}}^{i,l} := (V_{\text{set}}^l)^\dagger \otimes |P_l^i\rangle\langle P_l^i| + \sum_{|\psi_\perp\rangle} I \otimes |\psi_\perp\rangle\langle\psi_\perp|. \tag{A6}$$

Here, in the second term of the RHS, $|\psi_\perp\rangle$ runs over the $n_{\text{gr}} - 1$ states that constitute the orthonormal basis of H , the Hilbert space on the m_{gr} -qubit register, in combination with $|P_l^i\rangle$. The implementation is as shown in Fig. 8. Note that, in the circuit in this figure, $(V_{\text{set}}^l)^\dagger$ is activated if the second register takes $|P_l^i\rangle$ but not activated if it takes the state $|\psi\rangle$ orthogonal to $|P_l^i\rangle$, because $(V_{P_l^i})^\dagger|\psi\rangle$ is orthogonal to $(V_{P_l^i})^\dagger|P_l^i\rangle = |0\rangle$. We then obtain

$$V_{\text{OF2}}^i := \prod_{l=0}^{D-1} V_{\text{reset}}^{i,l} = \sum_{l=0}^{D-1} (V_{\text{set}}^l)^\dagger \otimes |P_l^i\rangle\langle P_l^i| + \sum_{|\psi_\perp\rangle} I \otimes |\psi_\perp\rangle\langle\psi_\perp|, \tag{A7}$$

where, in the second sum, $|\psi\rangle_{\perp}$ runs over the $n_{\text{gr}} - D$ states that constitute the orthonormal basis of H in combination with $|P_0^i\rangle, \dots, |P_{D-1}^i\rangle$.

Note that, for any $l \in [D]_0$,

$$V_{\text{OF2}}^i V_{\text{OF1}}^i |l\rangle|0\rangle = |0\rangle|P_l^i\rangle \tag{A8}$$

holds. Therefore, $V_{\text{OF2}}^i V_{\text{OF1}}^i$ with a SWAP gate added in the last position is V_{OF}^i , with the second register deemed ancillary.

Appendix B: Summary of basic properties of the matrix product state

The matrix product state (MPS) is a representation of a quantum state and, more generally, a high-order tensor, which was first introduced as a way to describe one-dimensional quantum systems efficiently [76–79]. It has since become an increasingly popular tool in studying quantum many-body systems. In addition, it has been used to investigate various physical phenomena, including topological phases of matter, quantum phase transitions, and quantum criticality (see review articles [46, 47] and references therein).

As in Sect. 2.4, let us consider the MPS representation of a tensor Ψ_{l_1, \dots, l_d} with d indices, each of which runs from 0 to $D - 1$. Any tensor can always be represented by an MPS, especially the right canonical form of MPS utilized in this paper, by recursively applying the Schmidt/singular-value decomposition as follows. First, let the degrees of freedom of $\{l_1, \dots, l_{d-2}\}$ be row components and the degrees of freedom of $\{l_{d-1}, l_d\}$ be column components for Ψ_{l_1, \dots, l_d} , and applying the singular-value decomposition (SVD), we obtain

$$\Psi_{l_1, \dots, l_d} = \sum_{k_{d-2}=1}^{\rho_{d-2}} V_{l_1, \dots, l_{d-2}, k_{d-2}}^1 \Lambda_{k_{d-2}}^1 U_{k_{d-2}, l_{d-1}, l_d}^{d-1}, \tag{B1}$$

where $\rho_{d-2} := \min\{D^{d-2}, D^2\}$, and left singular vectors $V_{l_1, \dots, l_{d-2}, k_{d-2}}^1$, singular values $\Lambda_{k_{d-2}}^1$, and right singular vectors $U_{k_{d-2}, l_{d-1}, l_d}^{d-1}$ for the SVD satisfy following orthonormal conditions

$$\sum_{l_1, \dots, l_{d-2}=0}^{D-1} \left(V_{l_1, \dots, l_{d-2}, k_{d-2}}^1 \right)^* V_{l_1, \dots, l_{d-2}, k'_{d-2}}^1 = \delta_{k_{d-2}, k'_{d-2}}, \tag{B2}$$

$$\sum_{k_{d-2}=1}^{\rho_{d-2}} \left(\Lambda_{k_{d-2}}^1 \right)^2 = \sum_{l_1, \dots, l_d=0}^{D-1} |\Psi_{l_1, \dots, l_d}|^2 := C^2, \tag{B3}$$

$$\sum_{l_{d-1}, l_d=0}^{D-1} U_{k_{d-2}, l_{d-1}, l_d}^{d-1} \left(U_{k'_{d-2}, l_{d-1}, l_d}^{d-1} \right)^* = \delta_{k_{d-2}, k'_{d-2}}. \tag{B4}$$

Then, we define the coefficients

$$\Psi_{l_1, \dots, l_{d-n-1}, k_{d-n-1}}^n = V_{l_1, \dots, l_{d-n-1}, k_{d-n-1}}^n \Lambda_{k_{d-n-1}}^n \tag{B5}$$

with $n = 1$, where n means the number of times SVD is applied.

Second, with $n = 2$, letting the degrees of freedom of $\{l_1, \dots, l_{d-n-1}\}$ be row components and the degrees of freedom of $\{l_{d-n}, k_{d-n}\}$ be column components for Ψ^{n-1} , and applying the singular-value decomposition (SVD) again, we obtain

$$\Psi_{l_1, \dots, l_{d-n}, k_{d-n}}^{n-1} = \sum_{k_{d-n-1}=1}^{\rho_{d-n-1}} V_{l_1, \dots, l_{d-n-1}, k_{d-n-1}}^n \Lambda_{k_{d-n-1}}^n U_{k_{d-n-1}, l_{d-n}, k_{d-n}}^{d-n}, \tag{B6}$$

where $\rho_{d-n-1} := \min\{D^{d-n-1}, D^{n+1}\}$, and V^n , Λ^n , and U^{d-n} satisfy

$$\sum_{l_1, \dots, l_{d-n-1}=0}^{D-1} \left(V_{l_1, \dots, l_{d-n-1}, k_{d-n-1}}^n \right)^* V_{l_1, \dots, l_{d-n-1}, k'_{d-n-1}}^n = \delta_{k_{d-n-1}, k'_{d-n-1}}, \tag{B7}$$

$$\sum_{k_{d-n-1}=1}^{\rho_{d-n-1}} \left(\Lambda_{k_{d-n-1}}^n \right)^2 = C^2, \tag{B8}$$

$$\sum_{l_{d-n}=0}^{D-1} \sum_{k_{d-n}=1}^{\rho_{d-n}} U_{k_{d-n-1}, l_{d-n}, k_{d-n}}^{d-n} \left(U_{k'_{d-n-1}, l_{d-n}, k_{d-n}}^{d-n} \right)^* = \delta_{k_{d-n-1}, k'_{d-n-1}}. \tag{B9}$$

Repeating the sequence of Eqs. (B5) to (B9) until $n = d - 2$ and defining $U_{l_1, k_1}^1 = \Psi_{l_1, k_1}^{d-2}$, which satisfies

$$\sum_{l_1=0}^{D-1} \sum_{k_1=1}^{\rho_1} |U_{l_1, k_1}^1|^2 = C^2, \tag{B10}$$

we can rewrite Ψ_{l_1, \dots, l_d} as the right canonical form of MPS

$$\Psi_{l_1, \dots, l_d} = \sum_{k_1=1}^{\rho_1} \dots \sum_{k_{d-2}=1}^{\rho_{d-2}} U_{l_1, k_1}^1 U_{k_1, l_2, k_2}^2 \dots U_{k_{d-2}, l_{d-1}, l_d}^{d-1}, \tag{B11}$$

and this form is equivalent to Eq. (13). In the practical calculation of MPS, the dimension ρ_{d-n-1} of the matrix Λ^n , which diverges exponentially with respect to d , is truncated to r , which we call the bond dimension. The numerical error ε that occurs when the truncation of the dimension in the MPS is introduced at the n -th SVD during the repeating procedure is the error that appears in the low-rank approximation of the

SVD, namely

$$\varepsilon = C^2 - \sum_{k_{d-n-1}=1}^{\min\{\rho_{d-n-1}, r\}} \left(\Lambda_{k_{d-n-1}}^n\right)^2. \tag{B12}$$

Therefore, the shape of the decay function of the singular value $\Lambda_{k_{d-n-1}}^n$, which is a monotonically decreasing nonnegative real number for k_{d-n-1} , is directly related to the error. It is known that, in the critical region of one-dimensional quantum systems, this decay function is power law and ε thus decays with a power law when we increase r and, accordingly, the DOF in the MPS representation, as seen in Fig. 6a and b.

The relationship between MPS and quantum circuits is as described in Sect. 2.5. We refer to the review articles [46, 47] and references therein for more detailed properties of MPS itself.

Appendix C: Derivative price as an approximation target function

Here, we explain the derivative price, which is considered as an approximation target in the above numerical experiment.

A derivative is a contract between two parties in which the amounts (*payoffs*) determined by the prices of some widely traded assets (*underlying assets*) such as stocks and bonds are paid and/or received between the parties. Under some mathematical models that describe the random movement of the underlying asset prices, we can use the established theory to calculate the derivative price (see [80, 81]).

In this study, we consider d underlying assets whose prices at time t are denoted by $\vec{S}(t) = (S_1(t), \dots, S_d(t))$ and obey the Black–Scholes (BS) model [82, 83] characterized by the following stochastic differential equation in the risk-neutral measure: for $i \in [d]$,

$$dS_i(t) = r_{\text{RF}}S_i(t)dt + \sigma_i S_i(t)dW_i(t). \tag{C1}$$

Here, r_{RF} is the real parameter called the risk-free interest rate, and $\sigma_1, \dots, \sigma_d$ are positive parameters called volatilities. W_1, \dots, W_d are Brownian motions and satisfy $dW_i dW_j = \rho_{ij}dt$ for $i, j \in [d]$, where the correlation matrix (ρ_{ij}) is symmetric and positive definite and satisfies $\rho_{11} = \dots = \rho_{dd} = 1$ and $-1 < \rho_{ij} < 1$ if $i \neq j$. Time $t = 0$ corresponds to the present.

We consider the derivative in which one party A receives the payoff from the other party B at a predetermined time $T > 0$, and its amount $f_{\text{pay}}(\vec{S}(T))$ depends on the underlying asset prices at T . Under some technical assumptions, the price of this derivative for A at time $t \in [0, T)$ with $\vec{S}(t)$ being $\vec{s} = (s_1, \dots, s_d)$ is given by

$$V(t, \vec{s}) = E \left[e^{-r_{\text{RF}}(T-t)} f_{\text{pay}}(\vec{S}(T)) \mid \vec{S}(t) = \vec{s} \right], \tag{C2}$$

where $E[\cdot]$ denotes the (conditional) expectation in the risk-neutral measure. This expectation can be calculated by Monte Carlo integration, that is, by generating many

sample paths of the time evolution of $\vec{S}(t)$ up to T and averaging $e^{-r_{RF}(T-t)} f_{\text{pay}}(\vec{S}(T))$ on the paths. It is also known that $V(t, \vec{s})$ can be obtained by solving the BS PDE

$$\begin{aligned} \frac{\partial}{\partial t} V(t, \vec{s}) + \frac{1}{2} \sum_{i,j=1}^d \sigma_i \sigma_j \rho_{ij} s_i s_j \frac{\partial^2}{\partial s_i \partial s_j} V(t, \vec{s}) \\ + r_{RF} \left(\sum_{i=1}^d s_i \frac{\partial}{\partial s_i} V(t, \vec{s}) - V(t, \vec{s}) \right) = 0 \end{aligned} \tag{C3}$$

backward from time T to t , with the boundary condition in the time direction being

$$V(T, \vec{s}) = f_{\text{pay}}(\vec{s}) \tag{C4}$$

and those in \vec{s} directions set according to the asymptotic behavior of $V(t, \vec{s})$ in the small and large asset price limits. Solving the BS PDE by quantum computing has been considered in previous studies [27, 32, 36–38].

Specifically, we consider the *worst-of put* option, which has the payoff function

$$f_{\text{pay}}(\vec{s}) = \max\{K - \min\{s_1, \dots, s_d\}, 0\}, \tag{C5}$$

and is often incorporated into exotic equity derivatives. Here, K is a positive constant called the strike.

We then regard $V(0, \vec{s})$ as a function $f(\vec{s})$ and attempt to approximate it. Because we cannot evaluate this analytically, we compute its values on the grid points using Monte Carlo integration with 10^5 sample paths and use these values in the numerical experiment. For this calculation, we used the TF Quant Finance library [84].

We set the upper bounds U_i and lower bounds L_i in the \vec{s} space as follows. U_i is set by

$$U_i = K \exp \left(\sqrt{2\sigma_i^2 T \log \frac{dK}{\epsilon}} \right) \tag{C6}$$

with $\epsilon = 0.01$, following [85] on the appropriate grid setting for solving the BS PDE. On the other hand, we simply set $L_i = 0.01K$.

In the numerical experiment described in Sect. 3, we set $r_{RF} = 0, \sigma_1 = \dots = \sigma_5 = 0.2, K = 100, T = 1$. The 10^4 random sample points in the \vec{s} space used for the numerical experiment were sampled from the distribution of $\vec{S}(t)$ at $t = 1$ with the initial value $\vec{S}(0)$ set to (K, \dots, K) . Therefore, we can regard the worst-of put option under consideration as the option 1 year after the start, at which it had a 2-year maturity and was at-the-money.⁴

⁴ The term ‘‘at-the-money’’ means the situation where the underlying asset prices are at the marginal point to yield the positive payoff.

References

1. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009). <https://doi.org/10.1103/PhysRevLett.103.150502>
2. Ambainis, A.: Variable time amplitude amplification and quantum algorithms for linear algebra problems. In: STACS'12 (29th Symposium on Theoretical Aspects of Computer Science), vol. 14 (LIPIcs, 2012), pp. 636–647. <https://doi.org/10.4230/LIPIcs.STACS.2012.636>
3. Clader, B.D., Jacobs, B.C., Sprouse, C.R.: Preconditioned quantum linear system algorithm. *Phys. Rev. Lett.* **110**, 250504 (2013). <https://doi.org/10.1103/PhysRevLett.110.250504>
4. Childs, A.M., Kothari, R., Somma, R.D.: Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J. Comput.* **46**, 1920 (2017). <https://doi.org/10.1137/16M1087072>
5. Chakraborty, S., Gilyén, A., Jeffery, S.: The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. In: 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019) (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019). <https://doi.org/10.4230/LIPIcs.ICALP.2019.33>
6. Subaşı, Y., Somma, R.D., Orsucci, D.: Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Phys. Rev. Lett.* **122**, 060504 (2019). <https://doi.org/10.1103/PhysRevLett.122.060504>
7. Lin, L., Tong, Y.: Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum* **4**, 361 (2020). <https://doi.org/10.22331/q-2020-11-11-361>
8. Tong, Y., An, D., Wiebe, N., Lin, L.: Fast inversion, preconditioned quantum linear system solvers, fast Green's-function computation, and fast evaluation of matrix functions. *Phys. Rev. A* **104**, 032422 (2021)
9. An, D., Lin, L.: Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm. *ACM Trans. Quantum Comput.* **3**, 5 (2022). <https://doi.org/10.1145/3498331>
10. Costa, P., An, D., Sanders, Y.R., Su, Y., Babbush, R., Berry, D.W.: Optimal scaling quantum linear systems solver via discrete adiabatic theorem (2021). arXiv preprint [arXiv:2111.08152](https://arxiv.org/abs/2111.08152)
11. Berry, D.W.: High-order quantum algorithm for solving linear differential equations. *J. Phys. A Math. Theor.* **47**, 105301 (2014). <https://doi.org/10.1088/1751-8113/47/10/105301>
12. Berry, D.W., Childs, A.M., Ostrander, A., Wang, G.: Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Commun. Math. Phys.* **356**, 1057 (2017). <https://doi.org/10.1007/s00220-017-3002-y>
13. Childs, A.M., Liu, J.-P.: Quantum spectral methods for differential equations. *Commun. Math. Phys.* **375**, 1427 (2020). <https://doi.org/10.1007/s00220-020-03699-z>
14. Xin, T., Wei, S., Cui, S., Xiao, J., Arrazola, I., Lamata, L., Kong, X., Lu, D., Solano, E., Long, G.: Quantum algorithm for solving linear differential equations: theory and experiment. *Phys. Rev. A* **101**, 032307 (2020). <https://doi.org/10.1103/PhysRevA.101.032307>
15. Cao, Y., Papageorgiou, A., Petras, I., Traub, J., Kais, S.: Quantum algorithm and circuit design solving the Poisson equation. *New J. Phys.* **15**, 013021 (2013). <https://doi.org/10.1088/1367-2630/15/1/013021>
16. Montanaro, A., Pallister, S.: Quantum algorithms and the finite element method. *Phys. Rev. A* **93**, 032324 (2016). <https://doi.org/10.1103/PhysRevA.93.032324>
17. Costa, P.C.S., Jordan, S., Ostrander, A.: Quantum algorithm for simulating the wave equation. *Phys. Rev. A* **99**, 012323 (2019). <https://doi.org/10.1103/PhysRevA.99.012323>
18. Childs, A.M., Liu, J.-P., Ostrander, A.: High-precision quantum algorithms for partial differential equations. *Quantum* **5**, 574 (2021). <https://doi.org/10.22331/q-2021-11-10-574>
19. Linden, N., Montanaro, A., Shao, C.: Quantum vs. classical algorithms for solving the heat equation. *Commun. Math. Phys.* (2022). <https://doi.org/10.1007/s00220-022-04442-6>
20. Liu, J.-P., Kolden, H.O., Krovi, H.K., Loureiro, N.F., Trivisa, K., Childs, A.M.: Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2026805118 (2021). <https://doi.org/10.1073/pnas.2026805118>
21. Xue, C., Wu, Y.-C., Guo, G.-P.: Quantum homotopy perturbation method for nonlinear dissipative ordinary differential equations. *New J. Phys.* **23**, 123035 (2021). <https://doi.org/10.1088/1367-2630/ac3eff>

22. Lloyd, S., De Palma, G., Gokler, C., Kiani, B., Liu, Z.-W., Marvian, M., Tennie, F., Palmer, T.: Quantum algorithm for nonlinear differential equations (2020). arXiv preprint [arXiv:2011.06571](https://arxiv.org/abs/2011.06571). <https://doi.org/10.48550/ARXIV.2011.06571>
23. Krovi, H.: Improved quantum algorithms for linear and nonlinear differential equations (2022). arXiv preprint [arXiv:2202.01054](https://arxiv.org/abs/2202.01054). <https://doi.org/10.48550/ARXIV.2202.01054>
24. Jin, S., Liu, N.: Quantum algorithms for computing observables of nonlinear partial differential equations (2022). arXiv preprint [arXiv:2202.07834](https://arxiv.org/abs/2202.07834). <https://doi.org/10.48550/ARXIV.2202.07834>
25. An, D., Fang, D., Jordan, S., Liu, J.-P., Low, G.H., Wang, J.: Efficient quantum algorithm for nonlinear reaction–diffusion equations and energy estimation (2022). arXiv preprint [arXiv:2205.01141](https://arxiv.org/abs/2205.01141). <https://doi.org/10.48550/ARXIV.2205.01141>
26. Kiani, B.T., De Palma, G., Englund, D., Kaminsky, W., Marvian, M., Lloyd, S.: Quantum advantage for differential equation analysis. *Phys. Rev. A* **105**, 022415 (2022). <https://doi.org/10.1103/PhysRevA.105.022415>
27. Miyamoto, K., Kubo, K.: Pricing multi-asset derivatives by finite-difference method on a quantum computer. *IEEE Trans. Quantum Eng.* **3**, 3100225 (2021). <https://doi.org/10.1109/TQE.2021.3128643>
28. Endo, S., Sun, J., Li, Y., Benjamin, S.C., Yuan, X.: Variational quantum simulation of general processes. *Phys. Rev. Lett.* **125**, 010501 (2020). <https://doi.org/10.1103/PhysRevLett.125.010501>
29. Lubasch, M., Joo, J., Moinier, P., Kiffner, M., Jaksch, D.: Variational quantum algorithms for nonlinear problems. *Phys. Rev. A* **101**, 010301 (2020). <https://doi.org/10.1103/PhysRevA.101.010301>
30. Liu, H.-L., Wu, Y.-S., Wan, L.-C., Pan, S.-J., Qin, S.-J., Gao, F., Wen, Q.-Y.: Variational quantum algorithm for the Poisson equation. *Phys. Rev. A* **104**, 022418 (2021). <https://doi.org/10.1103/PhysRevA.104.022418>
31. Sato, Y., Kondo, R., Koide, S., Takamatsu, H., Imoto, N.: Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation. *Phys. Rev. A* **104**, 052409 (2021). <https://doi.org/10.1103/PhysRevA.104.052409>
32. Fontanela, F., Jacquier, A., Oumgari, M.: A quantum algorithm for linear PDEs arising in finance. *SIAM J. Financ. Math.* **12**, SC98 (2021). <https://doi.org/10.1137/21M1397878>
33. Kyriienko, O., Paine, A.E., Elfving, V.E.: Solving nonlinear differential equations with differentiable quantum circuits. *Phys. Rev. A* **103**, 052416 (2021). <https://doi.org/10.1103/PhysRevA.103.052416>
34. Leong, F.Y., Ewe, W.-B., Koh, D.E.: Variational quantum evolution equation solver. *Sci. Rep.* **12**, 10817 (2022). <https://doi.org/10.1038/s41598-022-14906-3>
35. García-Molina, P., Rodríguez-Mediavilla, J., García-Ripoll, J.J.: Solving partial differential equations in quantum computers. *Phys. Rev. A* **105**, 012433 (2022). <https://doi.org/10.1103/PhysRevA.105.012433>
36. Alghassi, H., Deshmukh, A., Ibrahim, N., Robles, N., Woerner, S., Zoufal, C.: A variational quantum algorithm for the Feynman–Kac formula. *Quantum* **6**, 730 (2022). <https://doi.org/10.22331/q-2022-06-07-730>
37. Gonzalez-Conde, J., Rodríguez-Rozas, Á., Solano, E., Sanz, M.: Pricing financial derivatives with exponential quantum speedup (2021). arXiv preprint [arXiv:2101.04023](https://arxiv.org/abs/2101.04023). <https://doi.org/10.48550/ARXIV.2101.04023>
38. Radha, S.K.: Quantum option pricing using Wick rotated imaginary time evolution (2021). arXiv preprint [arXiv:2101.04280](https://arxiv.org/abs/2101.04280). <https://doi.org/10.48550/ARXIV.2101.04280>
39. Joo, J., Moon, H.: Quantum variational PDE solver with machine learning (2021). arXiv preprint [arXiv:2109.09216](https://arxiv.org/abs/2109.09216). <https://doi.org/10.48550/ARXIV.2109.09216>
40. Paine, A.E., Elfving, V.E., Kyriienko, O.: Quantum kernel methods for solving differential equations (2022). arXiv preprint [arXiv:2203.08884](https://arxiv.org/abs/2203.08884). <https://doi.org/10.48550/ARXIV.2203.08884>
41. Kubo, K., Miyamoto, K., Mitarai, K., Fujii, K.: Pricing multi-asset derivatives by variational quantum algorithms (2022). arXiv preprint [arXiv:2207.01277](https://arxiv.org/abs/2207.01277). <https://doi.org/10.48550/ARXIV.2207.01277>
42. Liu, Y., Chen, Z., Shu, C., Chew, S.C., Khoo, B.C.: Application of a variational hybrid quantum-classical algorithm to heat conduction equation (2022). arXiv preprint [arXiv:2207.14630](https://arxiv.org/abs/2207.14630). <https://doi.org/10.48550/ARXIV.2207.14630>
43. Aaronson, S.: Read the fine print. *Nat. Phys.* **11**, 291 (2015). <https://doi.org/10.1038/nphys3272>
44. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. *Contemp. Math.* **305**, 53 (2002). <https://doi.org/10.1090/conm/305>
45. Suzuki, Y., Uno, S., Raymond, R., Tanaka, T., Onodera, T., Yamamoto, N.: Amplitude estimation without phase estimation. *Quantum Inf. Process.* **19**, 75 (2020). <https://doi.org/10.1007/s11128-019-2565-2>

46. Orús, R.: A practical introduction to tensor networks: matrix product states and projected entangled pair states. *Ann. Phys.* **349**, 117 (2014). <https://doi.org/10.1016/j.aop.2014.06.013>
47. Okunishi, K., Nishino, T., Ueda, H.: Developments in the tensor network-from statistical mechanics to quantum entanglement. *J. Phys. Soc. Jpn.* **91**, 062001 (2022). <https://doi.org/10.7566/JPSJ.91.062001>
48. Nouy, A.: Low-rank methods for high-dimensional approximation and model order reduction. In: *Model Reduction and Approximation: Theory and Algorithms*, vol. 15, p. 3672148 (2017). <https://doi.org/10.1137/1.9781611974829.ch4>
49. Griebel, M., Harbrecht, H.: Analysis of tensor approximation schemes for continuous functions. *Found. Comput. Math.* (2021). <https://doi.org/10.1007/s10208-021-09544-6>
50. Ali, M., Nouy, A.: Approximation with tensor networks. Part I: approximation spaces (2020). arXiv preprint [arXiv:2007.00118](https://arxiv.org/abs/2007.00118). <https://doi.org/10.48550/ARXIV.2007.00118>
51. Ali, M., Nouy, A.: Approximation with tensor networks. Part II: approximation rates for smoothness classes (2020). arXiv preprint [arXiv:2007.00128](https://arxiv.org/abs/2007.00128). <https://doi.org/10.48550/ARXIV.2007.00128>
52. Ali, M., Nouy, A.: Approximation with tensor networks. Part III: multivariate approximation (2021). arXiv preprint [arXiv:2101.11932](https://arxiv.org/abs/2101.11932). <https://doi.org/10.48550/ARXIV.2101.11932>
53. Bachmayr, M., Nouy, A., Schneider, R.: Approximation by tree tensor networks in high dimensions: Sobolev and compositional functions (2021). arXiv preprint [arXiv:2112.01474](https://arxiv.org/abs/2112.01474). <https://doi.org/10.48550/ARXIV.2112.01474>
54. Griebel, M., Harbrecht, H., Schneider, R.: Low-rank approximation of continuous functions in Sobolev spaces with dominating mixed smoothness (2022). arXiv preprint [arXiv:2203.04100](https://arxiv.org/abs/2203.04100). <https://doi.org/10.48550/ARXIV.2203.04100>
55. Ran, S.-J.: Encoding of matrix product states into quantum circuits of one-and two-qubit gates. *Phys. Rev. A* **101**, 032310 (2020). <https://doi.org/10.1103/PhysRevA.101.032310>
56. Shirakawa, T., Ueda, H., Yunoki, S.: Automatic quantum circuit encoding of a given arbitrary quantum state (2021). arXiv preprint [arXiv:2112.14524](https://arxiv.org/abs/2112.14524). <https://doi.org/10.48550/ARXIV.2112.14524>
57. Zhizhishvili, L.: *Trigonometric Fourier Series and Their Conjugates*, vol. 372. Springer, Cham (1996). <https://doi.org/10.1007/978-94-009-0283-1>
58. Trefethen, L.N.: *Approximation Theory and Approximation Practice*, Extended Edition. SIAM, Philadelphia (2019). <https://doi.org/10.1137/1.9781611975949>
59. Boelens, A.M., Venturi, D., Tartakovsky, D.M.: Parallel tensor methods for high-dimensional linear PDEs. *J. Comput. Phys.* **375**, 519 (2018). <https://doi.org/10.1016/j.jcp.2018.08.057>
60. Dektor, A., Venturi, D.: Dynamically orthogonal tensor methods for high-dimensional nonlinear PDEs. *J. Comput. Phys.* **404**, 109125 (2020). <https://doi.org/10.1016/j.jcp.2019.109125>
61. Dektor, A., Venturi, D.: Dynamic tensor approximation of high-dimensional nonlinear PDEs. *J. Comput. Phys.* **437**, 110295 (2021). <https://doi.org/10.1016/j.jcp.2021.110295>
62. Dektor, A., Rodgers, A., Venturi, D.: Rank-adaptive tensor methods for high-dimensional nonlinear PDEs. *J. Sci. Comput.* **88**, 1 (2021). <https://doi.org/10.1007/s10915-021-01539-3>
63. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. *Phys. Rev. E* **58**, 5355 (1998). <https://doi.org/10.1103/PhysRevE.58.5355>
64. Morita, S., Nishimori, H.: Mathematical foundation of quantum annealing. *J. Math. Phys.* **49**, 125210 (2008). <https://doi.org/10.1063/1.2995837>
65. Hauke, P., Katzgraber, H.G., Lechner, W., Nishimori, H., Oliver, W.D.: Perspectives of quantum annealing: methods and implementations. *Rep. Prog. Phys.* **83**, 054401 (2020). <https://doi.org/10.1088/1361-6633/ab85b8>
66. Fishman, M., White, S.R., Stoudenmire, E.M.: The ITensor software library for tensor network calculations. *SciPost Phys. Codebases* (2022). <https://doi.org/10.21468/SciPostPhysCodeb.4>
67. Nishino, T., Okunishi, K., Kikuchi, M.: Numerical renormalization group at criticality. *Phys. Lett. A* **213**, 69 (1996). [https://doi.org/10.1016/0375-9601\(96\)00128-4](https://doi.org/10.1016/0375-9601(96)00128-4)
68. Tagliacozzo, L., de Oliveira, T.R., Iblisdir, S., Latorre, J.I.: Scaling of entanglement support for matrix product states. *Phys. Rev. B* **78**, 024410 (2008). <https://doi.org/10.1103/PhysRevB.78.024410>
69. Pollmann, F., Mukerjee, S., Turner, A.M., Moore, J.E.: Theory of finite-entanglement scaling at one-dimensional quantum critical points. *Phys. Rev. Lett.* **102**, 255701 (2009). <https://doi.org/10.1103/PhysRevLett.102.255701>
70. Pirvu, B., Vidal, G., Verstraete, F., Tagliacozzo, L.: Matrix product states for critical spin chains: finite-size versus finite-entanglement scaling. *Phys. Rev. B* **86**, 075117 (2012). <https://doi.org/10.1103/PhysRevB.86.075117>

71. Kjäll, J.A., Zaletel, M.P., Mong, R.S.K., Bardarson, J.H., Pollmann, F.: Phase diagram of the anisotropic spin-2 XXZ model: infinite-system density matrix renormalization group study. *Phys. Rev. B* **87**, 235106 (2013). <https://doi.org/10.1103/PhysRevB.87.235106>
72. Stojevic, V., Haegeman, J., McCulloch, I.P., Tagliacozzo, L., Verstraete, F.: Conformal data from finite entanglement scaling. *Phys. Rev. B* **91**, 035120 (2015). <https://doi.org/10.1103/PhysRevB.91.035120>
73. Ueda, H., Okunishi, K., Krčmár, R., Gendiar, A., Yunoki, S., Nishino, T.: Critical behavior of the two-dimensional icosahedron model. *Phys. Rev. E* **96**, 062112 (2017). <https://doi.org/10.1103/PhysRevE.96.062112>
74. Grover, L., Rudolph, T.: Creating superpositions that correspond to efficiently integrable probability distributions (2002). arXiv preprint quant-ph/0208112. <https://doi.org/10.48550/ARXIV.QUANT-PH/0208112>
75. Kaneko, K., Miyamoto, K., Takeda, N., Yoshino, K.: Quantum pricing with a smile: implementation of local volatility model on quantum computer. *EPJ Quantum Technol.* **9**, 7 (2022). <https://doi.org/10.1140/epjqt/s40507-022-00125-2>
76. Affleck, I., Kennedy, T., Lieb, E.H., Tasaki, H.: Valence bond ground states in isotropic quantum antiferromagnets. *Commun. Math. Phys.* **115**, 477 (1988). <https://doi.org/10.1007/BF01218021>
77. Fannes, M., Nachtergaele, B., Werner, R.F.: Finitely correlated states on quantum spin chains. *Commun. Math. Phys.* **144**, 443 (1992). <https://doi.org/10.1007/BF02099178>
78. Östlund, S., Rommer, S.: Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.* **75**, 3537 (1995). <https://doi.org/10.1103/PhysRevLett.75.3537>
79. Rommer, S., Östlund, S.: Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B* **55**, 2164 (1997). <https://doi.org/10.1103/PhysRevB.55.2164>
80. Hull, J.C.: *Options Futures and Other Derivatives*. Pearson, London (2003)
81. Shreve, S.E.: *Stochastic Calculus for Finance I & II*. Springer, Berlin (2004)
82. Scholes, M., Black, F.: The pricing of options and corporate liabilities. *J. Polit. Econ.* **81**, 637 (1973). https://doi.org/10.1142/9789814759588_0001
83. Merton, R.C.: Theory of rational option pricing. *Bell J. Econ. Manag. Sci.* **4**, 141 (1973). <https://doi.org/10.2307/3003143>
84. <https://github.com/google/tf-quant-finance>
85. Kangro, R., Nicolaidis, R.: Far field boundary conditions for Black–Scholes equations. *SIAM J. Numer. Anal.* **38**, 1357 (2000). <https://doi.org/10.1137/S0036142999355921>

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