

Efficient State Preparation for Quantum Amplitude Estimation

Almudena Carrera Vazquez^{1,2} and Stefan Woerner^{1,*}

¹IBM Quantum, IBM Research - Zurich, Rueschlikon, Switzerland

²ETH Zurich, Zurich, Switzerland



(Received 2 November 2020; revised 10 November 2020; accepted 15 February 2021; published 9 March 2021)

Quantum amplitude estimation (QAE) can achieve a quadratic speedup for applications classically solved by Monte Carlo simulation. A key requirement to realize this advantage is efficient state preparation. If state preparation is too expensive, it can diminish the quantum advantage. Preparing arbitrary quantum states has exponential complexity with respect to the number of qubits, and thus, is not applicable. Currently known efficient techniques require problems based on log-concave probability distributions, involve learning an unknown distribution from empirical data, or fully rely on quantum arithmetic. In this paper, we introduce an approach to simplify state preparation, together with a circuit optimization technique, both of which can help reduce the circuit complexity for QAE state preparation significantly. We demonstrate the introduced techniques for a numerical integration example on real quantum hardware, as well as for option pricing under the Heston model, i.e., based on a stochastic volatility process, using simulation.

DOI: [10.1103/PhysRevApplied.15.034027](https://doi.org/10.1103/PhysRevApplied.15.034027)

I. INTRODUCTION

Quantum amplitude estimation (QAE) is a quantum algorithm that can achieve a quadratic speedup over classical Monte Carlo simulation [1–5]. It has many possible applications, such as option pricing or risk analysis [6–9], or numerical integration [10,11].

A key requirement to apply QAE is to be able to efficiently load the problem of interest. In other words, we need a quantum circuit with a depth polynomial in the number of qubits that prepares a quantum state corresponding to the problem we like to solve. This can be achieved by using quantum arithmetic, which, although polynomial in the number of qubits, usually requires a significant overhead in terms of gates and ancilla qubits [12]. Another approach is to directly prepare a quantum state that corresponds to a probability distribution. However, preparing generic quantum states requires an exponential number of gates [13,14], and is thus not applicable here since it would diminish the quantum advantage. Efficient approaches either require the distribution to be log concave [15], leverage quantum machine learning techniques to train a quantum operator to approximate an unknown underlying distribution given empirical data [16], or approximate smooth, differentiable functions using piecewise polynomial approximations and matrix product states [17].

To model practically relevant problems, QAE necessitates rather deep quantum circuits with large numbers of qubits [8,9], and thus, likely requires a fault-tolerant error-corrected quantum computer. To get there, many technological advances are necessary, such as increasing the number of available qubits as well as improving the error rates and coherence times of quantum devices. Further, the overhead of quantum error-correction protocols needs to be reduced, and the performance of the used quantum algorithms as well as the scaling of the considered models needs to be improved. In this paper we contribute to solving this challenge from the very top of the stack, i.e., we propose a way to model particular problems of interest, and thus, help to lower the requirements on the actual quantum hardware.

In particular, we improve the known techniques and show how to simplify multiplication and addition of functions on the amplitude level, i.e., reducing part of the overhead introduced by quantum arithmetic. Furthermore, we introduce a circuit optimization technique that helps to asymptotically halve the circuit depth for QAE. We discuss QAE from a numerical integration point of view, which allows us to use standard error estimates from numerical integration and to study how the model error reduces with respect to the number of qubits used. This highlights the broad applicability of the introduced technique, e.g., to solve partial differential equations [18]. Finally, we demonstrate the introduced techniques, first by using a numerical integration experiment on a real quantum device, and second, by showing how to price a European

*wor@zurich.ibm.com

call option under the Heston model, i.e., considering a stochastic volatility process for the price of the underlying asset.

The remainder of this paper is structured as follows. Section II defines QAE and the corresponding state preparation problem. Section III introduces a more efficient state preparation scheme and extends it to stochastic processes. Section IV shows how the resulting quantum circuits can be significantly simplified. In Sec. V we analyze the error resulting from approximating a continuous function using a finite number of qubits, i.e., grid points. We link it to basic numerical integration and show how to reduce this error while keeping the number of qubits constant. Section VI illustrates our results using both simulation and quantum hardware, and Sec. VII concludes the paper.

II. QUANTUM AMPLITUDE ESTIMATION

Suppose an operator \mathcal{A} acting on $n + 1$ qubits as

$$\mathcal{A}|0\rangle_n|0\rangle = \sqrt{1-a}|\psi_0\rangle_n|0\rangle + \sqrt{a}|\psi_1\rangle_n|1\rangle, \quad (1)$$

where $|\psi_0\rangle, |\psi_1\rangle$ are normalized quantum states, and $a \in [0, 1]$ is the probability of measuring the last qubit in state $|1\rangle$. Following on from [1], we call states with $|1\rangle$ in the last qubit *good states*, and the others *bad states*.

Accordingly, QAE is a quantum algorithm that allows estimation of a with a quadratic speedup over classical Monte Carlo simulation. The algorithm repeatedly applies the operator $\mathcal{Q} = \mathcal{A}\mathcal{S}_0\mathcal{A}^\dagger\mathcal{S}_{\psi_0}$ to $\mathcal{A}|0\rangle_{n+1}$, where $\mathcal{S}_0 = \mathbb{I}_{n+1} - 2|0\rangle\langle 0|_{n+1}$ and $\mathcal{S}_{\psi_0} = \mathbb{I}_{n+1} - 2|\psi_0\rangle\langle 0|_{n+1}$ are reflections and \mathbb{I}_{n+1} is the identity operator on $n + 1$ qubits. The resulting error scales as $\mathcal{O}(1/M)$, where M denotes the number of (quantum) samples, i.e., applications of \mathcal{A} . In contrast, the error resulting from Monte Carlo simulation scales as $\mathcal{O}(1/\sqrt{M})$ for M (classical) samples [1, 19].

The canonical form of QAE requires controlled applications of \mathcal{Q} within quantum phase estimation [1]. Recently, different, simpler variants have been proposed, which require only $\mathcal{Q}^k\mathcal{A}|0\rangle_{n+1}$ to be ran for different powers k , such as the *maximum-likelihood amplitude estimation* (MLAE) [3] or the *iterative quantum amplitude estimation* (IQAE) [5]. In the remainder of this paper we consider MLAE for demonstrations, although our results are applicable to every variant of QAE. MLAE does not provide a theoretical guarantee on the result, however, it performs well in practice and is well suited as a benchmark, since it allows freedom on the choice of k . For a discussion and comparison of different variants of QAE, we refer to Ref. [5].

Due to the definition of \mathcal{A} in Eq. (1), \mathcal{S}_{ψ_0} can easily be constructed by considering only the ancilla qubit. Note that the original formulation of QAE does not require *good states* and *bad states* to be flagged by an ancilla qubit but is

more generic [1]. In general, it may be a bit more complex to implement the reflection \mathcal{S}_{ψ_0} , but the underlying theory holds as well. We leverage this fact later, when we flag *good states* and *bad states* using an encoding of multiple ancilla qubits.

The canonical QAE is based on quantum phase estimation (QPE), which introduces an overhead in terms of the number of required qubits and circuit depth. However, variants of QAE that can achieve a quadratic speedup without QPE [3–5] have recently been proposed. In order to apply these algorithms to a more generic problem than Eq. (1), we need to adjust \mathcal{S}_{ψ_0} to identify *good states* and *bad states*, and to be able to decide from a single measurement of all qubits whether we observe part of a *good state* or not. The latter can be achieved by using an encoding of multiple ancilla qubits.

A common way to construct \mathcal{A} is to first load a probability distribution and then apply an objective function, as outlined in the following. Assume n qubits, a random variable X defined by the possible values $x_i = a \times i + b$, $a, b \in \mathbb{R}$, with their corresponding probabilities $p_i \in [0, 1]$, $i = 0, \dots, 2^n - 1$, and a quantum operator \mathcal{U} acting as

$$\mathcal{U}|0\rangle_n = \sum_{i=0}^{2^n-1} \sqrt{p_i}|i\rangle_n. \quad (2)$$

Furthermore, suppose an objective function $g : \mathbb{R} \rightarrow [0, 1]$, one additional qubit in state $|0\rangle$, and a corresponding quantum operator \mathcal{G} defined by

$$\mathcal{G} : |i\rangle_n|0\rangle \mapsto |i\rangle_n \left[\sqrt{1-g(x_i)}|0\rangle + \sqrt{g(x_i)}|1\rangle \right]. \quad (3)$$

If we set $\mathcal{A} = \mathcal{G}(\mathcal{U} \otimes \mathbb{I})$ and apply it to $|0\rangle_{n+1}$, then the probability of measuring $|1\rangle$ in the last qubit is given by

$$\sum_{i=0}^{2^n-1} p_i g(x_i), \quad (4)$$

which is equal to the expected value $\mathbb{E}[g(X)]$.

Thus, if we can construct \mathcal{U} and \mathcal{G} efficiently, we can achieve a quadratic speedup to estimate $\mathbb{E}[g(X)]$ by using QAE for \mathcal{A} . Efficient ways to approximate \mathcal{G} for polynomial g are discussed in Refs. [7, 9]. For more general functions, \mathcal{G} can be constructed using quantum arithmetic, by first computing $\sin^{-1}[\sqrt{g(x_i)}]$ into an ancilla qubit register and then using controlled Y rotations to prepare the amplitude of the ancilla qubit.

However, as discussed in Sec. I, constructing \mathcal{U} is more challenging, and in general requires an exponential number of gates. In the following, we show how an alternative approach that allows efficient construction of \mathcal{A} whenever the probabilities p_i and g are given by efficiently computable functions. We show how this extends to multivariate problems as well as stochastic processes.

III. EFFICIENT STATE PREPARATION

Let us first assume the simple case where $p_i = 1/2^n$, i.e., X follows a uniform distribution. This is easy to prepare by applying Hadamard gates to all n state qubits. Then, applying \mathcal{G} , defined as before, leads to the state

$$\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \left[\sqrt{1-g(x_i)} |0\rangle + \sqrt{g(x_i)} |1\rangle \right], \quad (5)$$

with the probability of measuring $|1\rangle$ in the ancilla qubit being equal to

$$\frac{1}{2^n} \sum_{i=0}^{2^n-1} g(x_i), \quad (6)$$

as proposed in Ref. [2]. This can be interpreted as a *left Riemann sum* [20], i.e., a 2^n -point approximation of the integral $\int_{x=0}^1 g(x)dx$, assuming $x_i = i/2^n$.

Next, suppose a random variable X with a corresponding probability density function (PDF) $f : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$. We can replace g in Eq. (5) by the product of f and g , which allows us to approximate the expectation value $\mathbb{E}_f[g(X)]$. Depending on f , we may need to normalize the problem such that $f(x)g(x) \in [0, 1]$ for all x . In the following, we introduce an alternative approach to estimate $\mathbb{E}_f[g(X)]$, which can easily be extended to stochastic processes.

As just introduced, suppose a random variable X and the corresponding (normalized) PDF $f : \mathbb{R} \rightarrow [0, 1]$. Furthermore, suppose n state qubits and two ancilla qubits. Following the approaches outlined in Sec. II, we can prepare operators \mathcal{F} and \mathcal{G} such that

$$\begin{aligned} \mathcal{F} : |i\rangle_n |0\rangle |j\rangle &\mapsto \\ |i\rangle_n \left[\sqrt{1-f(x_i)} |0\rangle + \sqrt{f(x_i)} |1\rangle \right] |j\rangle, \end{aligned} \quad (7)$$

and

$$\begin{aligned} \mathcal{G} : |i\rangle_n |j\rangle |0\rangle &\mapsto \\ |i\rangle_n |j\rangle \left[\sqrt{1-g(x_i)} |0\rangle + \sqrt{g(x_i)} |1\rangle \right], \end{aligned} \quad (8)$$

where \mathcal{F} prepares the first ancilla qubit and \mathcal{G} prepares the second one. We now apply Hadamard gates to the first n qubits of $|0\rangle_n |00\rangle$ followed by \mathcal{F} and \mathcal{G} , which leads to

the state

$$\begin{aligned} &\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{1-f(x_i)} \sqrt{1-g(x_i)} |00\rangle \\ &+ \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{1-f(x_i)} \sqrt{g(x_i)} |01\rangle \\ &+ \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{f(x_i)} \sqrt{1-g(x_i)} |10\rangle \\ &+ \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{f(x_i)} \sqrt{g(x_i)} |11\rangle. \end{aligned} \quad (9)$$

Subsequently, the probability of measuring $|11\rangle$ for the last two qubits is given by

$$\frac{1}{2^n} \sum_{i=0}^{2^n-1} f(x_i)g(x_i), \quad (10)$$

which is again equal to a Riemann sum and approximates the expected value $\mathbb{E}_f[g(X)]$. In this case, we define $\mathcal{S}_{\psi_0} = (\mathbb{I}_{n+2} - 2|\psi_1\rangle\langle 11|)|\psi_1\rangle\langle 11|$ to construct \mathcal{Q} , i.e., *good states* and *bad states* are identified by the two ancilla qubits and we do not have to explicitly multiply f and g using quantum arithmetic. Note that we deviate from the usual definition here, i.e., we multiply the *good states* ($|\psi_1\rangle$) by -1 , instead of the *bad states* ($|\psi_0\rangle$). However, this implies only a difference in the global phase and can be ignored in the following.

Similar to multiplication, we can also construct operators to realize addition of functions. Suppose the operators \mathcal{G} and \mathcal{H} , corresponding to functions g and h , that not only share the control qubits but also the target qubit. Furthermore, let us add an ancilla qubit in state $|0\rangle$ and consider the initial state $|i\rangle_n |0\rangle |0\rangle$. Applying a Hadamard gate to the added ancilla and then the two (controlled) operators

$$\mathcal{G} \otimes |0\rangle\langle 0| + \mathbb{I}_{n+1} |1\rangle\langle 1|, \quad (11)$$

$$\mathcal{H} \otimes |1\rangle\langle 1| + \mathbb{I}_{n+1} |0\rangle\langle 0|, \quad (12)$$

leads to the state

$$\begin{aligned} &\frac{1}{\sqrt{2}} |i\rangle_n \left[\sqrt{1-g(x_i)} |0\rangle + \sqrt{g(x_i)} |1\rangle \right] |0\rangle \\ &+ \frac{1}{\sqrt{2}} |i\rangle_n \left[\sqrt{1-h(x_i)} |0\rangle + \sqrt{h(x_i)} |1\rangle \right] |1\rangle. \end{aligned} \quad (13)$$

Applying another Hadamard gate to the ancilla qubit transforms the state into

$$\begin{aligned} & \frac{1}{2} \left[\sqrt{1 - g(x_i)} + \sqrt{1 - h(x_i)} \right] |i\rangle_n |0\rangle |0\rangle \\ & + \frac{1}{2} \left[\sqrt{1 - g(x_i)} - \sqrt{1 - h(x_i)} \right] |i\rangle_n |0\rangle |1\rangle \\ & + \frac{1}{2} \left[\sqrt{g(x_i)} + \sqrt{h(x_i)} \right] |i\rangle_n |1\rangle |0\rangle \\ & + \frac{1}{2} \left[\sqrt{g(x_i)} - \sqrt{h(x_i)} \right] |i\rangle_n |1\rangle |1\rangle. \end{aligned} \quad (14)$$

If we now define the *good states* again as those with $|1\rangle$ in the target qubit, i.e., the second to last qubit, then the probability of measuring these states equals $[g(x_i) + h(x_i)]/2$, i.e., we can add the functions g and h . This construction is closely related to the *linear combination of unitaries* (LCU) [21]. However, LCU requires the ancilla qubit to be measured in a particular state to get the desired (nonunitary) operation. In the present context, this is not necessary, since we can adjust \mathcal{S}_{ψ_0} accordingly within QAE. Note that the addition introduces a factor of $1/2$ that we need to take into account by multiplying the resulting estimate by a factor of 2. This also increases the estimation error accordingly, which means that the error increases exponentially with the number of additions, which might limit the number of settings where this leads to a favorable scaling. Note that the expected value is linear, i.e., we may also realize a sum by estimating the terms individually.

Exploiting the presented approach, we can construct arbitrary combinations of additions and multiplications of functions for which we have oracles of the form given for $\mathcal{F}, \mathcal{G}, \mathcal{H}$. In case an addition should take place after one or more multiplications, i.e., in situations where the *good states* are flagged by multiple ancilla qubits being in state $|1\rangle$, it might be necessary to add an additional ancilla qubit, and apply a multicontrolled NOT gate to reduce back to a single qubit flagging the *good states*. A corresponding circuit is illustrated in Fig. 1.

So far, we focus on univariate problems. It is straightforward to extend this approach to multivariate problems as well, e.g., by representing the dimensions by separate quantum registers, possibly with a different number of qubits each. The required number of qubits will scale linearly in the dimension, unlike classical numerical integration schemes where the dependence is usually exponential—except for Monte Carlo simulation. In the following, we show how this approach can be extended to load stochastic processes.

Suppose a stochastic process given by an initial probability distribution $f_0(x^0)$ and the transition probabilities $f_t(x^t | x^{t-1})$, i.e., the probability to reach state x^t at time t , given the state history x^{t-1} . Here, to simplify the notation, we assume Markov processes, i.e., f_t depends only

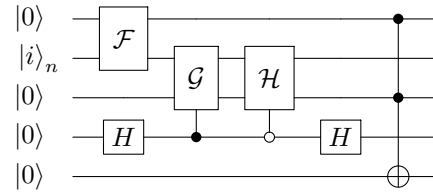


FIG. 1. Addition and multiplication of amplitudes for QAE: suppose functions f, g, h and the corresponding operators $\mathcal{F}, \mathcal{G}, \mathcal{H}$, as defined in Eq. (7), all sharing the control qubits $|i\rangle_n$, but with the difference that \mathcal{F} targets the qubit above $|i\rangle_n$ and the other two operators the qubit below $|i\rangle_n$. The illustrated circuit then prepares a state with the probability of measuring the bottom qubit in state $|1\rangle$ equal to $f(x_i)[g(x_i) + h(x_i)]$. The Toffoli gate is necessary in case another function should be added, otherwise, adjusting \mathcal{S}_{ψ_0} is sufficient.

on x^t and x^{t-1} , not the full history of the process. However, it is straightforward to extend our approach to generic (discretized) stochastic processes. Furthermore, suppose n_t qubits to represent the state x^t at time t , for $t = 0, \dots, T$, add $T + 1$ ancilla qubits, and define $n = n_0 + \dots + n_T$. Then, as in Eq. (7), we can construct an operator \mathcal{F}_0 corresponding to f_0 that prepares the first ancilla qubit. In addition, we can construct operators

$$\begin{aligned} \mathcal{F}_t : |i\rangle_{n_{t-1}} |j\rangle_{n_t} |0\rangle & \mapsto |i\rangle_{n_{t-1}} |j\rangle_{n_t} \\ & \times \left[\sqrt{1 - f_t(x_j^t | x_i^{t-1})} |0\rangle + \sqrt{f_t(x_j^t | x_i^{t-1})} |1\rangle \right]. \end{aligned} \quad (15)$$

If we first apply Hadamard gates to all state qubits and then apply $\mathcal{F}_0, \dots, \mathcal{F}_T$ to the corresponding qubit registers and ancilla qubits, we can construct the $(n + T + 1)$ -qubit state

$$\begin{aligned} & \frac{1}{\sqrt{2^n}} \sum_{i_0=0}^{2^{n_0}-1} \dots \sum_{i_T=0}^{2^{n_T}-1} |i_0\rangle_{n_0} \dots |i_T\rangle_{n_T} \\ & \left[\dots + \sqrt{f_0(x_{i_0}^0)} \prod_{t=1}^T f_t(x_{i_t}^t | x_{i_{t-1}}^{t-1}) |1 \dots 1\rangle_{T+1} \right], \end{aligned} \quad (16)$$

where we drop the terms without $|1 \dots 1\rangle_{T+1}$ in the ancilla qubits.

Given an objective function $g : \mathbb{R}^{T+1} \rightarrow [0, 1]$ and the corresponding operator \mathcal{G} , we can add another ancilla qubit and apply \mathcal{G} . The resulting probability of measuring $|1 \dots 1\rangle_{T+2}$ in all ancilla qubits is then given by

$$\frac{1}{2^n} \sum_{i_0, \dots, i_T} f_0(x_{i_0}^0) \prod_{t=1}^T f_t(x_{i_t}^t | x_{i_{t-1}}^{t-1}) g(x_{i_0}^0, \dots, x_{i_T}^T), \quad (17)$$

which approximates the expectation value $\mathbb{E}[g(X)]$ where $X = (x^0, \dots, x^T)$ represents the possible paths of the (discretized) stochastic process defined by the f_t .

Note that the complexity of S_{ψ_0} essentially equals the complexity of a multicontrolled NOT gate applied to the target qubits. Thus, its gate complexity is always less than the complexity of S_0 , which corresponds to a multicontrolled NOT gate that is controlled by all but one qubit (excluding work qubits that are used and then cleaned within the construction of \mathcal{A} , e.g., to realize quantum arithmetic). A detailed analysis of the complexity of implementing S_0 and S_{ψ_0} can be found in Ref. [8].

In principle, stochastic processes could also be loaded using a similar approach to Eq. (2). However, that would require the construction of operators

$$\mathcal{U}_t : |x_{t-1}\rangle_{n_{t-1}} |0\rangle_{n_t} \mapsto |x_{t-1}\rangle_{n_{t-1}} \sum_{i_t=0}^{2^{n_t}-1} \sqrt{f_t(x_{i_t}|x_{t-1})} |x_{i_t}\rangle_{n_t}, \quad (18)$$

which, like \mathcal{U} in Eq. (2), cannot be done efficiently for generic processes.

IV. SPIN-ECHO CIRCUIT OPTIMIZATION

Within this section we present a technique to optimize the circuits resulting from the construction introduced in Sec. III. More precisely, we leverage an effect that is also known as *spin echo* [22]. This can help to significantly reduce the circuit depth required to construct $\mathcal{Q}^k \mathcal{A} |0\rangle_n |0\rangle$.

Suppose circuits of the form

$$R_U(\theta) V R_U(-\theta), \quad (19)$$

where $U, V \in \{X, Y, Z\}$, i.e., R_U denotes a single-qubit Pauli rotation and V a single-qubit Pauli gate. In the case $U = V$, the gates commute and the circuit equals V . In all other cases, it can easily be seen that

$$R_U(\theta) V R_U(-\theta) = R_U(2\theta) V, \quad (20)$$

which is called the *spin echo*. Note that the right-hand side could also be written as $V R_U(-2\theta)$.

For a given function $f : \mathbb{R} \rightarrow \mathbb{R}$ let us define an operator R_U^f on $n+1$ qubits as

$$R_U^f : |x\rangle_n |\phi\rangle \mapsto |x\rangle_n R_U[f(x)] |\phi\rangle, \quad (21)$$

for an arbitrary single-qubit state $|\phi\rangle$ and $x \in \{0, \dots, 2^n - 1\}$. If we replace R_U in Eq. (20) by R_U^f , the same identity holds and we have

$$R_U^f(\mathbb{I}_n \otimes V) R_U^{(-f)} = R_U^{(2f)}(\mathbb{I}_n \otimes V), \quad (22)$$

where V is applied to the last qubit only.

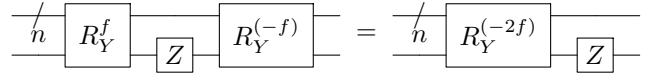


FIG. 2. Spin-echo circuit optimization for $U = Y$ and $V = Z$.

Suppose now that we want to use QAE to estimate the integral of f . We can set $U = Y$ and construct $\mathcal{A} = R_Y^f(H^{\otimes n} \otimes \mathbb{I})$. Furthermore, we set $V = Z$, which then corresponds to S_{ψ_0} in the definition of \mathcal{Q} . When constructing $\mathcal{Q}^k \mathcal{A} |0\rangle_n |0\rangle$, we repeatedly have the pattern $\mathcal{A}^\dagger S_{\psi_0} \mathcal{A}$, k times in total, which equals

$$(H^{\otimes n} \otimes \mathbb{I}) R_Y^{(-f)}(\mathbb{I}_n \otimes Z) R_Y^f(H^{\otimes n} \otimes \mathbb{I}). \quad (23)$$

The Hadamard gates at the beginning and the end are dominated by R_Y^f in terms of circuit complexity and we can ignore them in the following analysis. Thus, we have

$$R_Y^{(-f)}(\mathbb{I}_n \otimes Z) R_Y^f, \quad (24)$$

which, following Eq. (22), can be simplified to

$$R_Y^{(-2f)}(\mathbb{I}_n \otimes Z), \quad (25)$$

as illustrated in Fig. 2. In other words, we can drop the R_Y^f in \mathcal{A} and include its effect in the corresponding part of \mathcal{A}^\dagger by doubling the rotation angle. This means that $\mathcal{Q}^k \mathcal{A}$ can be constructed using only $k+1$ instead of $2k+1$ applications of \mathcal{A} (again ignoring $H^{\otimes n}$), i.e., essentially a reduction by a factor of 2 for larger k . Since, for larger problems, the complexity of \mathcal{A} will dominate the overall complexity [8], this directly translates to a corresponding reduction of the circuit depth for QAE.

This circuit optimization is compatible with the addition and multiplication techniques introduced in Sec. IV. Suppose, e.g., an operator $R_Y^{f(g+h)}$ corresponding to the circuit in Fig. 1, the spin-echo circuit optimization can be applied and allows us to remove half of the applications of $R_Y^{f(g+h)}$ in $\mathcal{Q}^k \mathcal{A}$.

V. ERROR ANALYSIS AND HIGHER-ORDER SCHEMES

In this section we consider QAE in the context of numerical integration and analyze the approximation error resulting from discretization using n qubits, denoted by E_n . Using basic insights from numerical quadrature, we show how to reduce the approximation error without increasing the number of discretization qubits. The discretization error needs to be added on top of the QAE estimation error, since QAE does not estimate the exact value but only an approximation. The total resulting error behaves like $E_n + \mathcal{O}(1/M)$. Understanding all error terms is crucial to balancing the number of qubits used to discretize with the target accuracy set for QAE.

Leveraging different approaches from numerical integration allows trading off classical repetitions of QAE with the resulting estimation error E_n . For more details on numerical integration, we refer to Ref. [20].

First, we consider the univariate problem $\int_{x=0}^1 g(x)dx$, for $g : [0, 1] \rightarrow [0, 1]$. Suppose we use n qubits to discretize the support of g , i.e., we use 2^n equally spaced grid points $x_i = i/2^n$, $i = 0, \dots, 2^n - 1$. Denote the left Riemann sum by R_n^{left} , where we use the number of qubits n as the index instead of the number of grid points 2^n . For simplicity, we assume throughout this section that g is an analytical function, i.e., continuously differentiable.

For R_n^{left} we know that the estimation error E_n^{left} , defined as

$$E_n^{\text{left}} = |\mathbb{E}[g(X)] - R_n^{\text{left}}|, \quad (26)$$

is bounded by

$$E_n^{\text{left}} \leq \frac{1}{2} \frac{\max_{x \in [0,1]} |\partial_x g(x)|}{2^n}. \quad (27)$$

In other words, the discretization error decreases exponentially with the number of qubits. The same holds true if we set $x_i = (i + 1)/2^n$, i.e., if we evaluate the *right Riemann sum* R_n^{right} .

If we define $x_i = (i + 1/2)/2^n$ instead, we are evaluating the *midpoint rule*, which results in R_n^{mid} . This leads to a better scaling, since the estimation error E_n^{mid} is bounded by

$$E_n^{\text{mid}} \leq \frac{1}{24} \frac{\max_{x \in [0,1]} |\partial_x^2 g(x)|}{2^{2n}}, \quad (28)$$

i.e., the error drops quadratically faster than for R_n^{left} or R_n^{right} , while the algorithm has exactly the same complexity and uses the same number of qubits.

The average of R_n^{left} and R_n^{right} leads to the *trapezoidal rule* as well as the corresponding estimator R_n^{trapez} , whose estimation error can be bounded by double the bound for the midpoint rule but requires two runs of QAE, one for the left and one for the right Riemann sum.

Taking the weighted average of the trapezoidal rule and the midpoint rule $(2R_n^{\text{mid}} + R_n^{\text{trapez}})/3$ leads to *Simpson's rule*, with the resulting estimator R_n^{Simpson} . Simpson's rule leads to an even better scaling, since the estimation error E_n^{Simpson} is bounded by

$$E_n^{\text{Simpson}} \leq \frac{1}{2880} \frac{\max_{x \in [0,1]} |\partial_x^4 g(x)|}{2^{4n}}, \quad (29)$$

i.e., by running QAE three times, we can significantly improve the estimation error.

Other quadrature rules for numerical integration, e.g., *Romberg's method* and *Richardson extrapolation*, are also

possible and lead to even higher orders of convergence, while keeping the number of grid points and qubits constant. Our approach can also be extended to nonequidistant grids, which allows the use of more advanced quadrature schemes to improve the performance, such as Gaussian quadrature [20]. However, nonequidistant grids require computing the grid points using quantum arithmetic before evaluating the function g , leading to longer circuits than equidistant grids and complicating the performance comparison. Computing the grid points first would also allow extending the scheme to approximate integrals over infinite domains.

Left and right Riemann sums as well as the midpoint rule can be easily extended to multivariate problems. For instance, suppose a d -dimensional function $g : [0, 1]^d \rightarrow [0, 1]$. The resulting error for the midpoint rule $E_{n,d}^{\text{mid}}$ is bounded by

$$E_{n,d}^{\text{mid}} \leq \frac{1}{24} \sum_{i=1}^d \frac{\max_{x \in [0,1]^d} |\partial_{x_i}^2 g(x)|}{2^{2n}}, \quad (30)$$

assuming n qubits per dimension, and analogously for the left and right Riemann sums. For all three rules, we can achieve an exponential number of grid points compared to the number of discretization qubits. Thus, unlike in the classical setting, they also scale efficiently for high-dimensional integrals. Extending the other approaches to higher dimensions does not scale as favorably, since, e.g., the number of combinations of left and right Riemann sums for different dimensions to evaluate the trapezoidal rule increases exponentially in d .

VI. RESULTS

Within this section, we demonstrate the developed methodology. First, we perform a small numerical integration experiment on real quantum hardware, leveraging spin-echo circuit optimization. Second, we show how to load the *Heston model*, a stochastic volatility model from mathematical finance, and use it to price a European call option. Both test cases are implemented in *Qiskit* [23].

A. Quantum hardware: Numerical integration

Consider the integral

$$\int_{x=0}^y \sin^2(\pi x) dx = \frac{2\pi y - \sin(2\pi y)}{4\pi}, \quad (31)$$

for $y \in [0, 1]$, and define $g(x) = \sin^2(\pi x)$ for further reference. In the following, we use the methodology introduced in this paper to approximate the integral using two and three qubits on real hardware. This analysis is similar to the simulation study in Ref. [3] and we compare the circuit complexity after spin-echo optimization to the numbers reported in Ref. [3].

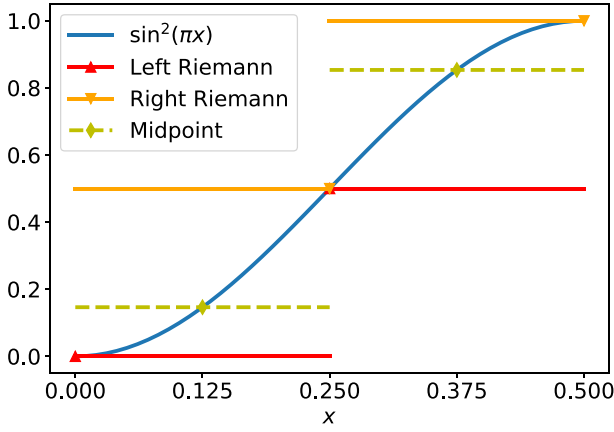


FIG. 3. Illustration of the left Riemann sum, right Riemann sum, and midpoint rule for $y = 1/2$, i.e., to approximate $\int_{x=0}^{1/2} \sin^2(\pi x) dx$.

For a given $y \in [0, 1]$, we discretize the interval $[0, y]$ to approximate the integral. We use one and two qubits for the discretization, i.e., two and four discretization points, and an additional qubit to represent the function g . We define the grid points x_i , $i = 0, \dots, 2^n - 1$, for n discretization qubits, according to the left Riemann sum, the right Riemann sum, and the midpoint rule, respectively. Having the results for all three quadrature rules also allows us to evaluate the corresponding values for Simpson's rule. Figure 3 illustrates the considered problem for $y = 1/2$ and a single discretization qubit. Note that for $y < 1$, we need to scale the result by y to adjust for the reduced interval length.

For the considered problem, the operator \mathcal{A} can be easily implemented using Hadamard gates and (controlled) Pauli Y -rotations R_y as illustrated for one discretization qubit in Fig. 4. The angles of the (controlled) Y rotations depend on x_i and are set such that

$$|i\rangle_n |0\rangle \mapsto |i\rangle R_y(2\pi x_i) |0\rangle. \quad (32)$$

The reflection S_{ψ_0} can be implemented by a single Pauli Z gate on the last qubit, while the reflection S_0 requires a (multi)controlled Z gate preceded and followed by X gates on all 6 qubits to achieve $|0 \dots 0\rangle_{n+1} \mapsto -|0 \dots 0\rangle_{n+1}$ instead of $|1 \dots 1\rangle_{n+1} \mapsto -|1 \dots 1\rangle_{n+1}$. The resulting \mathcal{Q} operator for one discretization qubit is illustrated in Fig. 5, and it is straightforward to extend this to multiple discretization qubits.

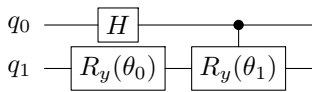


FIG. 4. Operator \mathcal{A} for $g(x) = \sin^2(\pi x)$ for $x \in [0, y]$. The angles need to be set according to the chosen discretization as $\theta_0 = 2\pi x_0$ and $\theta_1 = 2\pi(x_1 - x_0)$.

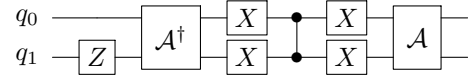


FIG. 5. Operator $\mathcal{Q} = \mathcal{A}S_0\mathcal{A}^\dagger S_{\psi_0}$ on two qubits, where the second qubit encodes the objective function g .

We use simulation as well as a real quantum device, *IBM Q Ourense*, which is accessible through the IBM Quantum Experience via Qiskit, to estimate $\mathbb{P}[|1\rangle]$ for the last qubit of $\mathcal{Q}^k \mathcal{A} |0\rangle_n |0\rangle$, for $n = 1, 2$, and for different values of k . We use MLAE to combine the different measurements to estimate the value of the considered integral. We set the power of \mathcal{Q} to $k = 2^j$ for $j = 0, \dots, k_{\max}$, for a chosen k_{\max} . The connectivity of IBM Q Ourense as well as the qubits used can be found in Appendix A, and the corresponding quantum circuits are shown in Appendix B.

We run every circuit using 8192 shots. To reduce the noise of the real quantum device, we apply readout error mitigation as well as error mitigation by inserting noisy gates combined with Richardson extrapolation [9,24–26]. Both techniques are described in more detail in Appendix C.

We estimate R_n^{left} , R_n^{right} , R_n^{mid} , and combine all three to get R_n^{Simpson} . The results are shown in Fig. 6, where we compare the analytic solution with the quantum estimates and show aggregated errors to illustrate how the performance changes with an increasing number of qubits and circuit depth. It can be nicely observed how the approximation error reduces with increasing order of the applied quadrature rule, as well as with the increasing number of discretization points, i.e., qubits. Furthermore, the quadratic speedup becomes apparent as we increase k_{\max} until we reach a point where the circuits are getting too long, i.e., the noise distorts the results too much and we do not see further improvements. Figure 6 also shows another important fact: for each quadrature rule there is a threshold, where the model error dominates the QAE estimation error, beyond which it would not make sense to increase k_{\max} . In other words, once that point is reached, the result cannot be improved anymore without increasing the number of discretization points. For three qubits, the error peaks for a few points. The reason for this is the lack of numerical robustness of MLAE, i.e., sometimes small errors in the measured data can lead to a large deviation of the maximum-likelihood estimate.

As mentioned above, our circuits leverage spin-echo circuit optimization to reduce the circuit depth. Table I shows the resulting numbers of CNOT gates for two and three qubits, in each case with and without the circuit optimization. For three qubits analyze all-to-all connectivity, as well as linear connectivity, which is the available layout on IBM Q Ourense for three qubits. The table shows that the introduced circuit-optimization technique can significantly

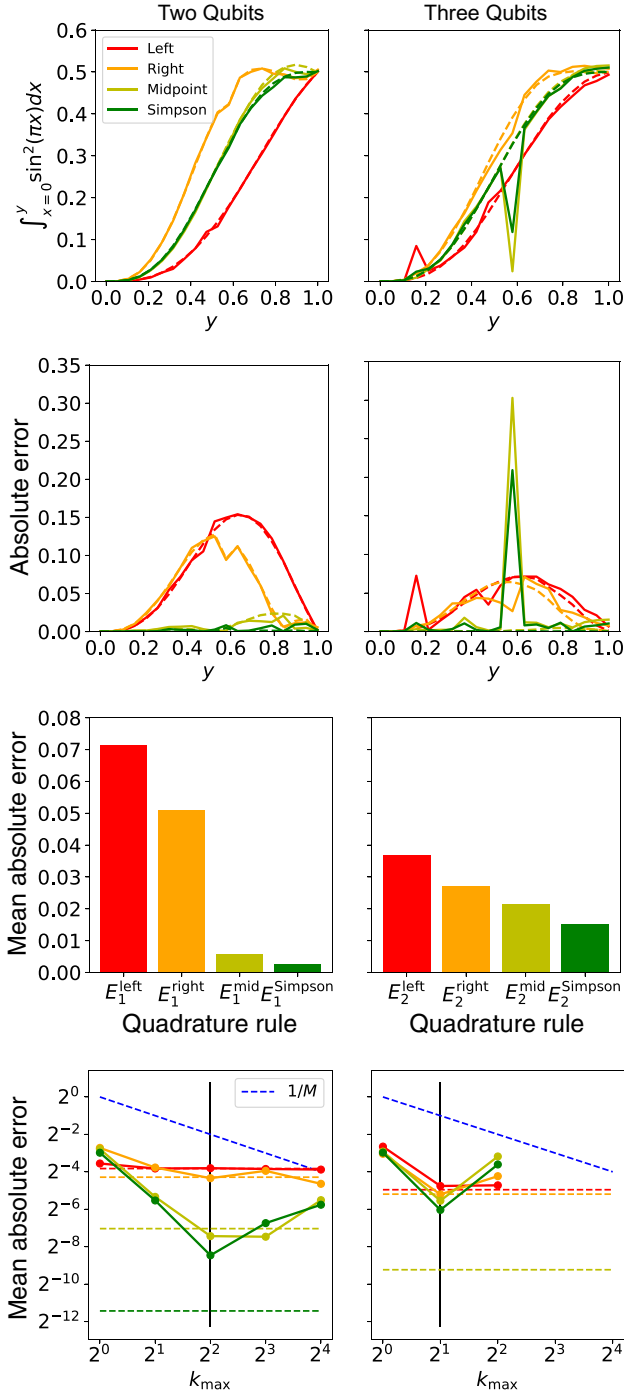


FIG. 6. MLAE results for numerical integration. Left: results for $n = 1$. Right: results for $n = 2$. First row: estimated values for different y , using $k_{\max} = 2$ (left) and $k_{\max} = 1$ (right)—the same parameters are used for rows two and three. Second row: absolute error with respect to the analytic result. Third row: mean absolute error over y . Fourth row: mean absolute error with respect to k_{\max} . The blue dashed line indicates $1/M$, i.e., the theoretical convergence rate as a reference. The other dashed lines indicate the errors resulting from the analytic evaluation of the quadrature rules. The black vertical line indicates the corresponding choice of k_{\max} for rows one to three.

TABLE I. Number of CNOT gates to implement $\mathcal{Q}^k \mathcal{A} |0\rangle$: for two qubits, the hardware topology is irrelevant. For three qubits, the hardware topology matters and we compare all-to-all connectivity and linear connectivity (as used on IBM Q Ourense). We show all results with and without spin-echo circuit optimization. Note that during the circuit optimization, we can also drop the very last CNOT gate and correct the measurements accordingly with a classical postprocessing.

No. Qubits	2		3		
Topology	all-to-all	linear	all-to-all
Optimized	yes	no	yes	yes	no
	4	7	13	17	18
2	7	12	23	31	32
k 4	13	22	43	59	60
8	25	42	83	115	116
16	49	82	163	227	228

reduce the number of CNOT gates. Even when compiled to linear connectivity, the resulting number of CNOT gates is not larger than without spin-echo circuit optimization for all-to-all connectivity, which is equal to the numbers provided in Ref. [3].

For two qubits, i.e., $n = 1$, it is interesting to see that we achieve reasonable results for k_{\max} up to 16 (49 CNOT gates) for R_n^{left} , R_n^{right} , and R_n^{mid} . However, for R_n^{Simpson} , we can only go to $k_{\max} = 4$ (13 CNOT gates). The reason is that the resulting error for the first three methods is not dominated by QAE but by the quadrature rule. For R_n^{Simpson} and $k_{\max} > 4$ this seems to change and we cannot improve the results further. For three qubits, i.e., $n = 2$, we get a reasonable behavior for $k_{\max} \leq 2$ for all four quadrature rules, including R_n^{Simpson} , which corresponds to 31 CNOT gates on the real device. At a first glance, it might seem contradictory that we can run R_n^{Simpson} with more CNOT gates on three qubits than on two qubits. However, as before, the two-qubit circuit should result in a much smaller estimation error than the three-qubit circuit, which means that the three-qubit circuit can tolerate more noise before it starts to dominate the estimated values.

Note that for two qubits, it is possible to optimize the circuit to using only three CNOT gates, independently of k_{\max} . This can be achieved by first evaluating the resulting unitary four-by-four matrix and then decomposing it into gates again. However, this method neither scales well to larger numbers of qubits nor does it provide a benchmark on the number of gates we can apply on the real hardware. Nevertheless, we perform this experiment as well and report the results in Appendix D.

B. Simulation: Heston model

In this section we show how to price a European call option under the *Heston model*. A European call option

TABLE II. Probabilities. This table provides the probabilities of certain values for v_1 and S_1 , which are independent variables, as well as the conditional probabilities for S_2 given v_1 and S_1 .

v_1 (\mathbb{P})	S_1 (\mathbb{P})	$\mathbb{P}[S_2 = x \mid v_1, S_1]$			
		$x = 0$	$x = 1$	$x = 2$	$x = 3$
0.8 (0.50)	0.75 (0.38)	0.063	0.937	0.001	0.000
	1.25 (0.62)	0.007	0.631	0.361	0.001
1.2 (0.50)	0.75 (0.38)	0.105	0.890	0.005	0.000
	1.25 (0.62)	0.022	0.592	0.382	0.005
marginal distribution		0.040	0.725	0.233	0.002

gives its owner the right but not the obligation to buy an underlying stock at a fixed time, the maturity T , and a fixed price, the strike price K . The Heston model is an example of a stochastic volatility model used in financial mathematics to describe, for instance, the behavior of stock prices [27]. This model is chosen as an illustration, however, the introduced techniques are applicable to any arbitrary stochastic processes as long as the initial and transition probabilities are given as functions that can be (classically) calculated efficiently.

The Heston model consists of a first stochastic process describing the development of the volatility, and a second one—depending on the volatility—describing the development of the stock price. The model is specified by the rate of return μ of a stock, the long run average price variance θ , the rate κ of reverting to θ , the volatility of the volatility ξ , and the following two stochastic partial differential equations for the volatility v_t and the stock price S_t , for $t \in \mathbb{R}_{\geq 0}$:

$$dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t}dW_t^v, \quad (33)$$

$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^S, \quad (34)$$

where W_t^S, W_t^v are Wiener processes with correlation ρ .

Suppose a given initial volatility v_0 and a given initial stock price S_0 . We can then discretize the time $t = 0, 1, 2, \dots$ with a time-step size δt and we derive the discrete transition laws as

$$v_{t+1} = v_t + \kappa(\theta - v_t)\delta t + \xi\sqrt{v_t}X_t^v, \quad (35)$$

$$S_{t+1} = S_t + \mu S_t \delta t + \sqrt{v_t} S_t X_t^S, \quad (36)$$

where

$$(X_t^S, X_t^v) \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \delta t \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right]. \quad (37)$$

This allows us to derive the conditional probabilities

$$f_t(v_t, S_t \mid v_{t-1}, S_{t-1}) \quad (38)$$

as the PDFs of two-dimensional normal distributions.

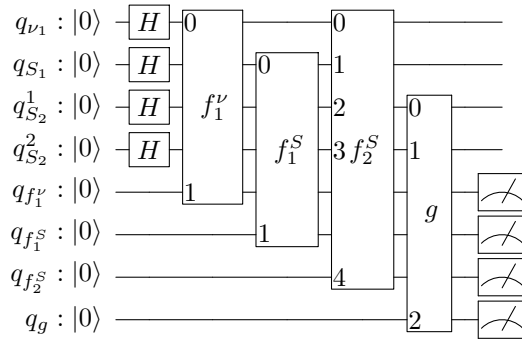


FIG. 7. A circuit for a European call option under the Heston model: first, Hadamard gates are applied to the state qubits to span the equal superposition for the state discretization. Then, we apply the rotations corresponding to the different probability density functions, where the last qubit of the gate always denotes the target. Last, we apply a gate corresponding to the objective function $g(S_2) = \max\{S_2 - K, 0\}$. The four measurements indicate that we need to take into account the last four qubits to estimate the expected payoff. More precisely, the probability of measuring $|1111\rangle$ corresponds to the (normalized) expected payoff we are interested in.

Given a model for the stock price at maturity of the option, i.e., for S_T , the expected payoff of a European call option is given by

$$\mathbb{E}[\max\{S_T - K, 0\}], \quad (39)$$

and equals the fair option price before discounting. For a constant interest rate, discounting just results in a simple correction term to compute the present value of the future payoff. It is also straightforward to extend to stochastic interest rates. We can add another stochastic process for the discount rate and then use quantum arithmetic to discount

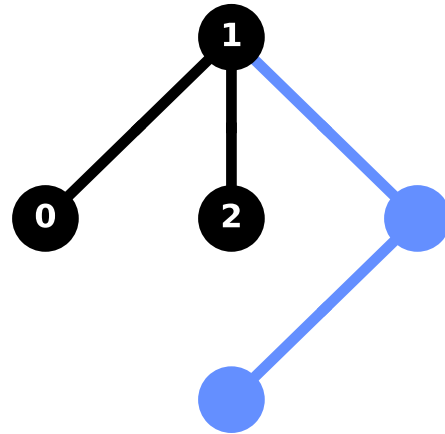


FIG. 8. The connectivity of IBM Q Ourense. For the two-qubit experiments we use qubits 0 and 1, and for the three-qubit experiments we use qubits 0, 1, and 2.

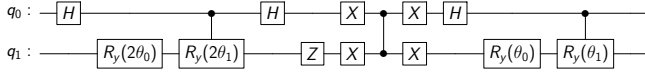


FIG. 9. Spin-echo optimized circuit for $\mathcal{QA}|0\rangle$ for two qubits. The angles need to be set, according to the discretization, as $\theta_0 = 2\pi x_0$ and $\theta_1 = 2\pi(x_1 - x_0)$. Note that the first set of R_y rotations has a factor of 2 for the angle due to the spin-echo optimization.

the expected payoff and directly use QAE to estimate the fair option price. However, for ease of presentation, we ignore discounting and just focus on the expected payoff.

For an illustrative example, we now set $K = 1$, $\delta t = 1$, and $T = 2$, i.e., $t = 0, 1, 2$. Furthermore, we assume $v_0 = 1$, $S_0 = 1$ and set $\kappa = 1$, $\theta = 1$, $\xi = 0.5$, $\mu = 1$, and $\rho = 0$. We discretize $v_1 \in \{0.8, 1.2\}$ and $S_1 \in \{0.75, 1.25\}$ using one qubit each, and $S_2 \in \{0, 1, 2, 3\}$ using two qubits. Note that we do not need to represent v_2 since S_2 does not depend on it, and thus, it does not contribute to the option price. Thus, we use four qubits in total to discretize the volatility and stock price processes.

Given the grid points x_{i-1} , x_i , x_{i+1} for some index i , and the continuous probability density functions f_i introduced above, the probability for x_i is defined as the probability of the interval $[(x_{i-1} + x_i)/2, (x_i + x_{i+1})/2]$. In case x_i is the first (last) grid point, the lower bound (upper bound) is replaced by negative (positive) infinity.

For given v_{t-1} and S_{t-1} , and assuming $\rho = 0$, v_t and S_t are independent. Thus, we can split the functions f_i into f_i^v and f_i^S to simplify the computation and we leverage the multiplication technique introduced in Sec. III. We add three ancilla qubits to represent f_1^v, f_1^S, f_2^S and another one to represent the actual payoff, i.e., four ancillas in total. The resulting probabilities are provided as a reference in Table II. Note that we need to normalize the payoff function $g(S_2) = \max\{S_2 - K, 0\}$ such that it takes values in $[0, 1]$. Given $K = 1$ and the range for S_3 , this implies that we need to divide g by 2. The resulting eight-qubit circuit is illustrated in Fig. 7. For simplicity, we use uniformly controlled Pauli rotations as provided by Qiskit [23,28] to implement the different operations, i.e., we precompute the rotation angles for the different cases. In general, we would need to compute the actual transition probabilities using quantum arithmetic.

We simulate the circuit introduced above by using the simulators provided by Qiskit, and evaluate the probability

of measuring $|1111\rangle$ in the last four qubits, which equals 0.1185. This can be easily verified using the marginal distribution for S_2 provided in Table II and the (normalized) payoff function $g(S_2)$. Thus, the operator \mathcal{A} illustrated in Fig. 7 corresponds to pricing a European call option under the Heston model and can directly be used with QAE and its variants to achieve a quadratic speedup over classical Monte Carlo simulation.

Within this section, we focus on a European call option, i.e., a very simple type of option. However, it is straightforward to extend the results in this paper to more exotic options, e.g., path-dependent options, following the techniques introduced in Ref. [9]. Similarly, the Heston model is used only as illustration and we can extend the methodology to arbitrary stochastic processes.

VII. CONCLUSIONS

In the present paper we introduce an efficient approach to preparing quantum states for QAE and show how basic numerical integration can help to reduce the approximation error while keeping the number of qubits constant. Furthermore, we develop a generic circuit optimization technique for QAE and demonstrate our insights on a simple numerical integration problem using real quantum hardware as well as on a relevant model from financial mathematics using simulation.

This is a significant enhancement of the state of the art. We do not require the probability distribution functions to be log concave and do not impose any other requirements on their structure except being efficiently computable. Note that our approach is also more efficient than the loading scheme for log-concave functions proposed in Ref. [15]. The improvement we demonstrate in this paper is only possible because we are not treating state preparation separately, but in the context of QAE, i.e., together with the algorithm where the prepared state is being used.

Determining the most efficient loading scheme will depend on the exact situation and is a task for future research. It is likely that this will result in a combination of different approaches automatically constructed by future quantum compilers.

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

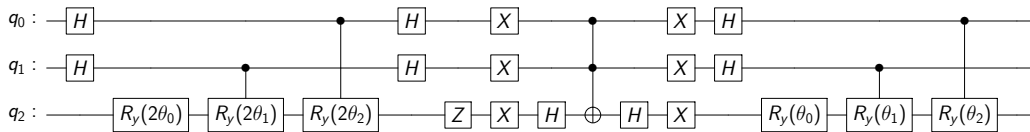


FIG. 10. Spin-echo optimized circuit for $\mathcal{QA}|0\rangle$ for three qubits. The angles need to be set, according to the (equidistant) discretization, as $\theta_0 = 2\pi x_0$, $\theta_1 = 2\pi(x_1 - x_0)$, and $\theta_2 = 2\pi(x_2 - x_0)$. Note that the first set of R_y rotations has a factor of 2 for the angle due to the spin-echo optimization.

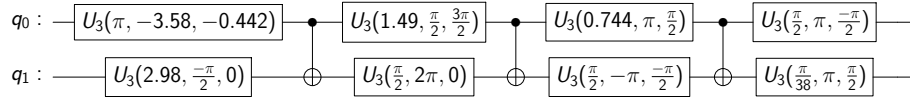


FIG. 11. The numerical integration circuit for the midpoint rule with $k = 1$ using one qubit for discretization after optimizing to only three CNOT gates.

ACKNOWLEDGMENTS

We thank David Sutter and Dmitri Maslov for the constructive technical discussions on data loading and circuit optimization. We further acknowledge the support of the National Centre of Competence in Research Quantum Science and Technology (QSIT).

IBM, IBM Q, and Qiskit are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product or service names may be trademarks or service marks of IBM or other companies.

All authors researched, collated, and wrote this paper.

APPENDIX A: IBM Q OURENSE

Figure 8 shows the connectivity of IBM Q Ourense, the quantum device used for the experiments on real hardware, as well as the qubits used.

APPENDIX B: QUANTUM CIRCUITS

This section describes the quantum circuits that are evaluated in Sec. VIA. Figures 9 and 10 show the circuit for $\mathcal{Q}\mathcal{A}|0\rangle$, for two and three qubits, respectively.

It is straightforward to expand to multiple applications of \mathcal{Q} by repeating \mathcal{Q} and taking into account the spin-echo circuit optimization as indicated in the figures.

APPENDIX C: ERROR MITIGATION

We apply the same strategy to mitigate errors as, for instance, also used in Ref. [9]. First, readout errors are mitigated by running a calibration sequence over all basis states to construct a matrix representing the conditional distribution of measurements given a prepared state. This can subsequently be used to correct the measurements in our experiments. More details can be found in Refs. [23,29]. Second, to mitigate the error of CNOT gates, we first amplify the noise and then extrapolate to the zero-noise limit. More precisely, we run a circuit as given, then we replace every CNOT gate by three CNOT gates, and last, by five CNOT gates. In theory, inserting these gates should have no effect, since an odd number of CNOT gates should be equal to a single CNOT gate. In practice, this amplifies the error of the CNOT gates to 3 times and 5 times the original error. Having these three data points with the increasing error allows us to do a quadratic extrapolation to the zero-noise limit, which leads to the results shown in

this paper. More details on this technique can be found in Refs. [24–26].

APPENDIX D: QUANTUM HARDWARE RESULTS FOR TWO QUBITS WITH THREE CNOT GATES

This section shows the results for the numerical integration problem introduced in Sec. VIA for the two-qubit circuit optimized to use only three CNOT gates in total. This can be achieved by first classically evaluating the corresponding unitary matrix and then decomposing it into quantum gates again, for instance using the functionality provided by Qiskit. The resulting circuit for $k = 1$ is shown in Fig. 11. It should be noted that this approach does not scale to larger numbers of qubits, and is reported here for comparison only.

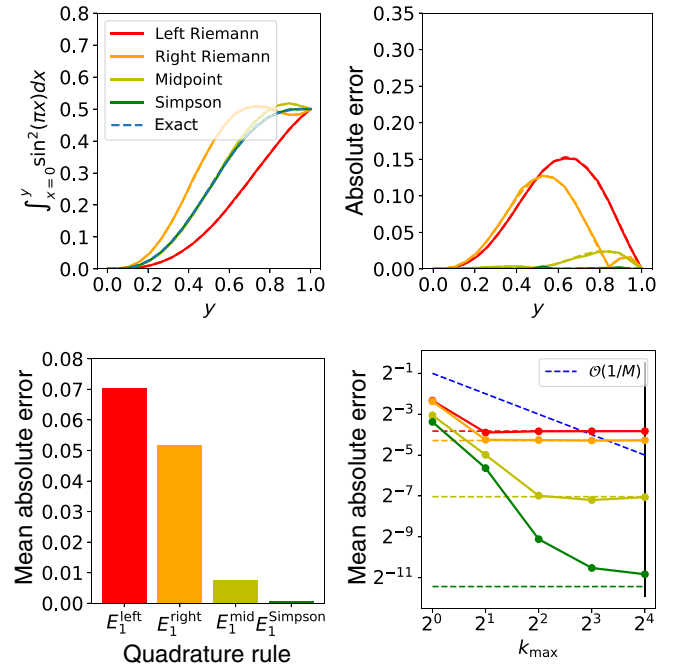


FIG. 12. MLAE results for two qubits compressed to three CNOT gates. Top left: estimated values for different y , using $k_{\max} = 16$ (same for top right and bottom left). Top right: absolute error with respect to analytic result. Bottom left: mean absolute error over y . Bottom right: mean absolute errors with respect to k_{\max} . The blue dashed line indicates $\mathcal{O}(1/M)$. The other dashed lines indicate errors resulting from the analytic evaluation of the quadrature rules. The black vertical line indicates the corresponding choice of k_{\max} for the other figures.

As before, the circuits are run on IBM *Q* Ourense using 8192 shots and error mitigation via insertion of noisy CNOT gates (cf. Sec. VI A and Appendix C). Figure 12 shows the results in the same way as presented in Sec. VI A. It can be seen that we get good results for all k_{\max} , although the convergence of Simpson's rule starts to slow down towards $k_{\max} = 16$, which is likely due to the remaining errors of the corresponding circuit starting to dominate as the estimated result gets more and more accurate. Nevertheless, we can estimate the integral with an average absolute error smaller than $2^{-10} \approx 10^{-3}$ using Simpson's rule.

-
- [1] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemp. Math.* **305**, 53 (2002).
 - [2] A. Montanaro, Quantum speedup of monte carlo methods, *Proc. R. Soc. A: Math., Phys. Eng. Sci.* **471**, 20150301 (2015).
 - [3] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Amplitude estimation without phase estimation, *Quantum Inf. Process.* **19**, 75 (2020).
 - [4] S. Aaronson and P. Rall, *Quantum Approximate Counting, Simplified*, Ser. Proceedings. Society for Industrial and Applied Mathematics (Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, US, 2019), p. 24.
 - [5] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative Quantum Amplitude Estimation, [arXiv:1912.05559](https://arxiv.org/abs/1912.05559) [quant-ph] (2019).
 - [6] P. Rebentrost, B. Gupt, and T. R. Bromley, Quantum computational finance: Monte carlo pricing of financial derivatives, *Phys. Rev. A* **98**, 022321 (2018).
 - [7] S. Woerner and D. J. Egger, Quantum risk analysis, *npj Quantum Inf.* **5**, 15 (2019).
 - [8] D. J. Egger, R. Garcia Gutierrez, J. Cahue Mestre, and S. Woerner, Credit risk analysis using quantum computers, *IEEE Trans. Comput.* **1** (2020).
 - [9] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, Option pricing using quantum computers, *Quantum* **4**, 291 (2020).
 - [10] D. S. Abrams and C. P. Williams, Fast quantum algorithms for numerical integrals and stochastic processes, [arXiv:quant-ph/9908083](https://arxiv.org/abs/quant-ph/9908083) (1999).
 - [11] A. Papageorgiou and J. F. Traub, *Quantum Algorithms and Complexity for Continuous Problems* (Springer, New York, NY, 2009), p. 7118.
 - [12] T. Häner, M. Roetteler, and K. M. Svore, Optimizing Quantum Circuits for Arithmetic, [arXiv:1805.12445](https://arxiv.org/abs/1805.12445) [quant-ph] (2018).
 - [13] V. V. Shende, S. S. Bullock, and I. L. Markov, Synthesis of quantum-logic circuits, *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.* **25**, 1000 (2006).
 - [14] M. Plesch and C. Brukner, Quantum-state preparation with universal gate decompositions, *Phys. Rev. A* **83**, 032302 (2010).
 - [15] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, [arXiv:quant-ph/0208112](https://arxiv.org/abs/quant-ph/0208112) (2002).
 - [16] C. Zoufal, A. Lucchi, and S. Woerner, Quantum generative adversarial networks for learning and loading random distributions, *npj Quantum Inf.* **5**, 103 (2019).
 - [17] A. Holmes and A. Matsuura, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Piscataway, New Jersey, United States, 2020), p. 169.
 - [18] F. Gaitan, Finding flows of a navier-stokes fluid through quantum computing, *npj Quantum Inf.* **6**, 61 (2020).
 - [19] P. Glasserman, *Monte Carlo Methods in Financial Engineering* (Springer-Verlag, New York, 2003).
 - [20] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration* (Academic Press, Cambridge, Massachusetts, US, 1984), 2nd ed.
 - [21] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian Dynamics with a Truncated Taylor Series, *Phys. Rev. Lett.* **114**, 090502 (2015).
 - [22] T. Das, A. Saha, and D. Roy, Quantum-mechanical analysis of spin-echo phenomena, *Proc. R. Soc. London* **227**, 407 (1955).
 - [23] G. Aleksandrowicz *et al.*, Qiskit: An open-source framework for quantum computing (2019), <https://raw.githubusercontent.com/Qiskit/qiskit/master/Qiskit.bib>.
 - [24] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, *Phys. Rev. Lett.* **119**, 1 (2017).
 - [25] A. Kandala, K. Temme, A. D. Corcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Error mitigation extends the computational reach of a noisy quantum processor, *Nature* **567**, 491 (2018).
 - [26] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, Cloud Quantum Computing of an Atomic Nucleus, *Phys. Rev. Lett.* **120**, 210501 (2018).
 - [27] S. L. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, *Rev. Financ. Stud.* **6**, 327 (1993).
 - [28] R. Iten, O. Reardon-Smith, L. Mondada, E. Redmond, R. S. Kohli, and R. Colbeck, Introduction to universalqcompiler, [arXiv:1904.01072](https://arxiv.org/abs/1904.01072) [quant-ph] (2019).
 - [29] A. Dewes, F. R. Ong, V. Schmitt, R. Lauro, N. Boulant, P. Bertet, D. Vion, and D. Esteve, Characterization of a Two-Transmon Processor with Individual Single-Shot Qubit Readout, *Phys. Rev. Lett.* **108**, 057002 (2012).