Amplitude estimation without phase estimation

Yohichi Suzuki[1] · Shumpei Uno[1,2] · Rudy Raymond[1,3] · Tomoki Tanaka[1,4] · Tamiya Onodera[1,3] · Naoki Yamamoto[1,5]

Abstract

This paper focuses on the quantum amplitude estimation algorithm, which is a core subroutine in quantum computation for various applications. The conventional approach for amplitude estimation is to use the phase estimation algorithm, which consists of many controlled amplification operations followed by a quantum Fourier transform. However, the whole procedure is hard to implement with current and near-term quantum computers. In this paper, we propose a quantum amplitude estimation algorithm without the use of expensive controlled operations; the key idea is to utilize the maximum likelihood estimation based on the combined measurement data produced from quantum circuits with different numbers of amplitude amplification operations. Numerical simulations we conducted demonstrate that our algorithm asymptotically achieves nearly the optimal quantum speedup with a reasonable circuit length.

Keywords Quantum amplitude estimation · Classical post-processing · Maximum likelihood estimation · Cramér–Rao bound

Yohichi Suzuki and Shumpei Uno are equally contributing authors.

✉ Naoki Yamamoto
  yamamoto@appi.keio.ac.jp

[1] Quantum Computing Center, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan

[2] Mizuho Information and Research Institute, Inc., 2-3 Kanda-Nishikicho, Chiyoda-ku, Tokyo 101-8443, Japan

[3] IBM Research - Tokyo, 19-21 Nihonbashi Hakozaki-cho, Chuo-ku, Tokyo 103-8510, Japan

[4] Mitsubishi UFJ Financial Group, Inc. and MUFG Bank, Ltd., 2-7-1 Marunouchi, Chiyoda-ku, Tokyo 100-8388, Japan

[5] Department of Applied Physics and Physico-Informatics, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan

1 Introduction

Quantum computers are expected to allow us to perform high-speed computations over classical computations for problems in a wide range of scientific and technological fields. Environments in which quantum algorithms can be executed by real quantum devices are currently being provided [1–3]. Real quantum devices with several tens of qubits will soon be realized in near future, although those are the so-called noisy intermediate-scale quantum (NISQ) devices that impose several practical limitations on their use [4], both in the number of gate operations and in the number of available qubits. Hence, several custom subroutines taking into account these constraints have been proposed, typically the variational quantum eigensolver [5,6].

In this paper, we focus on the amplitude estimation algorithm, which is a core subroutine in quantum computation for various applications, e.g., in chemistry [7,8], finance [9,10], and machine learning [11–14]. In particular, quantum speedup of Monte Carlo sampling via amplitude estimation [15] lies in the heart of these applications. Therefore, in light of its importance, we followed the aforementioned direction and developed a new amplitude estimation algorithm that can be executed in NISQ devices.

Note that Ref. [16] demonstrated that the amplitude estimation problem can be formulated as a phase estimation problem [17], where the amplitude to be estimated is inferred from the eigenvalue of the corresponding amplification operator. Owing to the ubiquitous nature of the eigenvalue estimation problem, some versions of the phase estimation algorithm suitable for NISQ devices [18–22] have been proposed (with the last one appeared slightly after ours), and they all rely on classical post-processing statistics such as the Bayes method. However, these modified phase estimation algorithms as well as the original scheme [17] still involve many controlled operations (e.g., the controlled amplification operation in the case of Ref. [16]) that can be difficult to implement on NISQ devices. Therefore, a new algorithm specialized to the amplitude estimation problem is required, one that does not use expensive controlled operations.

The goal of amplitude estimation is, in its simplest form, to estimate the unknown parameter θ contained in the state $|\Psi\rangle = \sin\theta \, |good\rangle + \cos\theta \, |bad\rangle$, where $|good\rangle$ and $|bad\rangle$ are given orthogonal state vectors. Our scheme is composed of the amplitude amplification process and the maximum likelihood (ML) estimation; the controlled operations and the subsequent quantum Fourier transform (QFT) are not involved. The amplification process transforms the coefficient of $|good\rangle$ to $\sin((2m + 1)\theta)$ with m being the number of operations; if θ is known, then, by suitably choosing m, we can enhance the probability of hitting "good" quadratically greater than the classical case, where no amplitude amplification is utilized [23]. However, the function $\sin((2m+1)\theta)$ does not always take a relatively large value for a certain m because θ is unknown in this problem, meaning that an effective quantum speedup is not always available; also, the ML estimate is not uniquely determined due to the periodicity of this function. Our strategy is the first to make measurements on the transformed quantum state and to construct likelihood functions for several m, say $\{m_0, \ldots, m_M\}$, and then combine them to construct a single likelihood function that uniquely produces the ML estimate; see Fig. 1. The broad concept behind this scheme is to combine the data produced from different quantum circuits, and the scheme might be performed even on NISQ devices

to compute a target value faster than classical algorithms via some post-processing. Actually a numerical simulation demonstrates that, by appropriately designing $\{m_k\}$, compared with the classical sampling we can achieve nearly a square root reduction in the total number of queries to reach the specified estimation precision; notably, only relatively short-depth circuits are required to achieve this quantum speedup. We also show that, in an application of the amplitude estimation to Monte Carlo integration, our algorithm requires many fewer controlled NOT (CNOT) gates than the conventional phase-estimation-based approach, so it is suitable for obtaining quantum advantages with NISQ devices. Note that Ref. [24] also took the approach without using the phase estimation method, but it needed to change the query in each iteration, which is highly demanding in practice. Also the paper Ref. [25] gave an amplitude estimation scheme that employs a Bayes rule together with applying random Unitary operations (subjected to the Haar measure) to ideally realize the quadratic speedup, without a controlled Unitary operation; this scheme is applicable to low-dimensional quantum circuits, due to the hardness to implement the random Unitaries.

2 Preliminary

We herein briefly describe the quantum amplitude amplification, which is the basis of our approach for the amplitude estimation problem.

Our proposed algorithm mainly consists of two parts: quantum amplitude amplification and amplitude estimation based on likelihood analysis. The amplitude amplification [26,27] is the generalization of the Grover's quantum searching algorithm [23]. Similar to quantum searching, the amplitude amplification is known to achieve quadratic speedup over the corresponding classical algorithm.

We assume a unitary operator $\mathscr{A}$ that acts on $(n + 1)$ qubits, such that $|\Psi\rangle = \mathscr{A}|0\rangle_{n+1} = \sqrt{a}|\tilde{\Psi}_1\rangle|1\rangle + \sqrt{1 - a}|\tilde{\Psi}_0\rangle|0\rangle$, where $a \in [0, 1]$ is the unknown parameter to be estimated, while $|\tilde{\Psi}_1\rangle$ and $|\tilde{\Psi}_0\rangle$ are the n-qubit normalized good and bad states. The query complexity of estimating a is counted by the number of the operations of $\mathscr{A}$, which is often denoted as the number of *queries* for simplicity. By performing measurements on $|\Psi\rangle$ repeatedly, we can infer a from the ratio of obtaining the good and bad states, but the number of queries is exactly the same as the classical one in this case.

The advantage offered by the quantum amplitude amplification is that, instead of measuring right after the single operation of $\mathscr{A}$, we can amplify the probability of obtaining the good state by applying the following operator.

$$\mathbf{Q} = -\mathscr{A}\mathbf{S}_0\mathscr{A}^{-1}\mathbf{S}_\chi, \tag{1}$$

where the operator $\mathbf{S}_\chi$ puts a negative sign to the good state, i.e., $\mathbf{S}_\chi|\tilde{\Psi}_1\rangle|1\rangle = -|\tilde{\Psi}_1\rangle|1\rangle$, and does nothing to the bad state. Similarly, $\mathbf{S}_0$ puts a negative sign to the all-zero state $|0\rangle_{n+1}$ and does nothing to the other states. $\mathscr{A}^{-1}$ is the inverse of $\mathscr{A}$, the operation of which requires the same query complexity as $\mathscr{A}$.

By defining a parameter $\theta_a \in [0, \pi/2]$ such that $\sin^2 \theta_a = a$, we have

$$\mathscr{A}|0\rangle_{n+1} = \sin \theta_a |\tilde{\Psi}_1\rangle|1\rangle + \cos \theta_a |\tilde{\Psi}_0\rangle|0\rangle. \tag{2}$$

Brassard et al. [16] showed that repeatedly applying $\mathbf{Q}$ for m times on $|\Psi\rangle$ results in

$$\mathbf{Q}^m|\Psi\rangle = \sin((2m + 1)\theta_a)|\tilde{\Psi}_1\rangle|1\rangle + \cos((2m + 1)\theta_a)|\tilde{\Psi}_0\rangle|0\rangle. \tag{3}$$

This equation represents that, after applying $\mathbf{Q}$ m times (with $2m$ queries), we can obtain the good state with a probability of at least $4m^2$ times larger than that obtained from $\mathscr{A}|0\rangle_{n+1}$ for sufficiently small a. This is in contrast with having $2m$ number of measurements from $\mathscr{A}|0\rangle_{n+1}$, which only gives the good state with probability $2m$ times larger. This intuitively gives the quadratic speedup obtained from the amplitude amplification: if we can infer the ratio of the good state after the amplitude amplification, we can estimate the value of a from the number of queries required to obtain such a ratio.

The conventional amplitude estimation [16] utilizes the quantum phase estimation which requires a quantum circuit that implements the multiple controlled $\mathbf{Q}$ operations, namely, Controlled-$\mathbf{Q}$: $|m\rangle|\Psi\rangle \rightarrow |m\rangle\mathbf{Q}^m|\Psi\rangle$. Performing the controlled operations simultaneously on many m's consecutively and gathering the amplitude by the inverse QFT enables an accurate estimation of a [16]. However, this approach suffers from the need for many controlled gates (thus, CNOT gates) and additional ancilla qubits (the number of which is dictated by the required accuracy). Such an approach can be problematic for NISQ devices.

3 Amplitude estimation without phase estimation

3.1 Algorithm

This section shows the quantum algorithm to estimate θ_a in Eq. (3) without using the conventional phase-estimation-based method [16]. The first stage of the algorithm is to make good or bad measurements on the quantum state $\mathbf{Q}^{m_k}|\Psi\rangle$ for a chosen set of $\{m_k\}$. Let N_k be the number of measurements (shots) and h_k be the number of measuring good states for the state $\mathbf{Q}^{m_k}|\Psi\rangle$; then, because the probability measuring the good state is $\sin^2((2m_k + 1)\theta_a)$, the likelihood function representing this probabilistic event is given by

$$L_k(h_k; \theta_a) = \left[\sin^2((2m_k + 1)\theta_a)\right]^{h_k} \left[\cos^2((2m_k + 1)\theta_a)\right]^{N_k - h_k}. \tag{4}$$

The second stage of the algorithm is to combine the likelihood functions $L_k(h_k; \theta_a)$ for several $\{m_0, \ldots, m_M\}$ to construct a single likelihood function $L(\mathbf{h}; \theta_a)$:

$$L(\mathbf{h}; \theta_a) = \prod_{k=0}^{M} L_k(h_k; \theta_a), \tag{5}$$
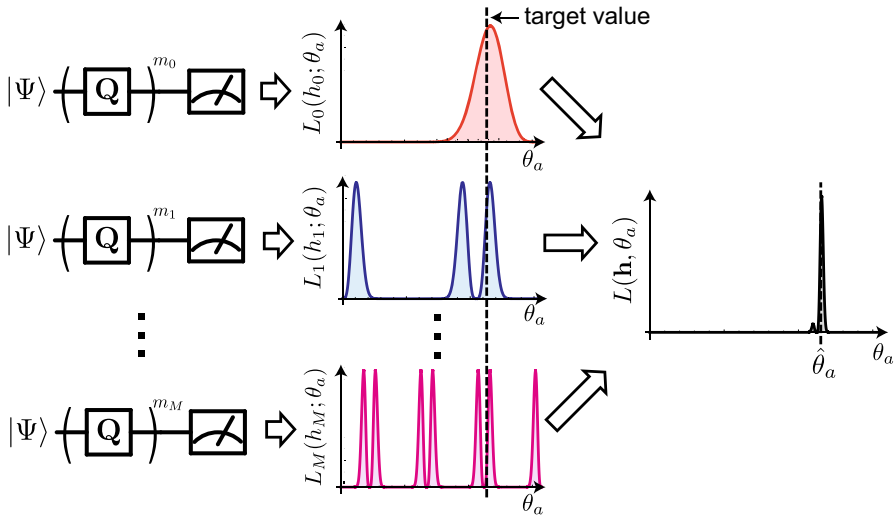
Fig. 1 Schematic picture of our amplitude estimation algorithm using the ML estimation. After preparing the states $\mathbf{Q}^{m_k} |\Psi\rangle$, the numbers of measuring good states, i.e., h_k are obtained (left). Based on the obtained h_k, the likelihood functions $L_k(h_k; \theta_a)$ are constructed (center). Finally, a single likelihood function $L(\mathbf{h}; \theta_a)$ is introduced by combining the likelihood functions $L_k(h_k; \theta_a)$ (right). The ML estimate is the value that maximizes the likelihood function $L(\mathbf{h}; \theta_a)$

where $\mathbf{h} = (h_0, h_1, \ldots, h_M)$. The ML estimate is defined as the value that maximizes $L(\mathbf{h}; \theta_a)$:

$$\hat{\theta}_a = \arg \max_{\theta_a} L(\mathbf{h}; \theta_a) = \arg \max_{\theta_a} \ln L(\mathbf{h}; \theta_a) \tag{6}$$

The whole procedure is summarized in Fig. 1. Now a and θ_a are uniquely related through $a = \sin^2 \theta_a$ in the range $0 \leq \theta_a \leq \pi/2$, and $\hat{a} := \sin^2 \hat{\theta}_a$ is the ML estimate for a; thus, in what follows, $L(\mathbf{h}; a)$ is denoted as $L(\mathbf{h}; \theta_a)$. Note that the random variables $h_0, h_1, \ldots, h_M$ are independent but not identically distributed because the probability distribution for obtaining h_k, i.e., $p_k(h_k; \theta_a) \propto L_k(h_k; \theta_a)$, is different for each k; however, the set of multidimensional random variables $\mathbf{h} = (h_0, h_1, \ldots, h_M)$ is independently generated from the identical joint probability distribution $p(\mathbf{h}; \theta_a) \propto L(\mathbf{h}; \theta_a)$.

This algorithm has two caveats: (i) if only a single amplitude amplification circuit is used like in the Grover search algorithm, i.e., the case $M = 0$ and $m_0 \neq 0$, the ML estimate $\hat{\theta}_a$ cannot be uniquely determined due to the periodicity of $L_0(h_0; \theta_a)$, and (ii) if no amplification operator is applied, i.e., $m_k = 0 \ \forall k$, then the ML estimate is unique, but it does not have any quantum advantages, as shown later. Hence, the heart of our algorithm can be regarded as the *quantum circuit fusion* technique that combines some quantum circuits to determine the target value uniquely, while some quantum advantage is guaranteed.

3.2 Statistics: Cramér–Rao bound and Fisher information

The remaining to be determined in our algorithm was to design the sequences $\{m_k, N_k\}$ so that the resulting ML estimate $\hat{\theta}_a$ might have a distinct quantum advantage over the classical one. Here, we introduce a basic statistical method to carry out this task and, based on that method, give some specific choice of $\{m_k, N_k\}$.

First, in general, the *Fisher information* $\mathscr{I}(a)$ is defined as

$$\mathscr{I}(a) = \mathbb{E}\left[\left(\frac{\partial}{\partial a}\ln L(x; a)\right)^2\right], \tag{7}$$

where the expectation is taken over a random variable x subjected to a given probability distribution $p(x; a)$ with an unknown parameter a. The importance of the Fisher information can be clearly seen from the fact that any estimate $\hat{a}$ satisfies the following *Cramér–Rao inequality*.

$$\text{var}(\hat{a}) = \mathbb{E}[(\hat{a} - \mathbb{E}[\hat{a}])^2] \geq \frac{[1 + b'(a)]^2}{\mathscr{I}(a)}, \tag{8}$$

where $b(a)$ represents the bias defined by $b(a) = \mathbb{E}[\hat{a} - a]$ and $b'(a)$ indicates the derivative of $b(a)$ with respect to a. It is easy to see that the mean squared estimation error satisfies

$$\mathbb{E}[(\hat{a} - a)^2] \geq \frac{[1 + b'(a)]^2}{\mathscr{I}(a)} + b(a)^2. \tag{9}$$

A specifically important property of the ML estimate, which maximizes the likelihood function $\prod_k p(x_k; a)$ with the measurement data x_k, is that it becomes unbiased, i.e., $b(a) = 0$, and further achieves the equality in Eq. (9) in the large number limit of measurement data [28]; that is, the ML estimate is asymptotically optimal.

In our case, by substituting Eqs. (4) and (5) into Eq. (7) together with a straight forward calculation $\mathbb{E}[h_k] = N_k \sin^2((2m_k + 1)\theta_a)$, we find that

$$\mathscr{I}(a) = \frac{1}{a(1 - a)}\sum_{k=0}^{M} N_k(2m_k + 1)^2. \tag{10}$$

Also, for any sequences $\{m_k, N_k\}$, the total number of queries is given as

$$N_q = \sum_{k=0}^{M} N_k(2m_k + 1). \tag{11}$$

As stated before, the coefficient 2 multiplying m_k in Eq. (11) originates from the fact that the operator $\mathbf{Q}$ uses $\mathscr{A}$ and $\mathscr{A}^{-1}$, and the constant $+1$ is due to the initial state preparation of $|\Psi\rangle = \mathscr{A}|0\rangle_{n+1}$. If $\mathbf{Q}$ is not applied to $|\Psi\rangle$ and if only the final

measurements are performed for $|\Psi\rangle$, i.e., $m_k = 0$ for all k, the total number of queries is identical to that of classical random sampling. Because N_k and $(2m_k+1)$ are positive integers, the Fisher information in Eq. (10) satisfies the following relation.

$$\mathscr{I}(a) \;\leq\; \frac{1}{a(1-a)}\left(\sum_{k=0}^{M} N_k(2m_k+1)\right)^2 \;=\; \frac{1}{a(1-a)}N_q^2. \qquad (12)$$

Here, $\hat{a}$ is set to the ML estimate (6), and the estimation error is considered to be $\hat{\varepsilon} = \sqrt{\mathbb{E}[(\hat{a}-a)^2]}$ in this case. The total number of measurements $\sum_{k=0}^{M} N_k$ is assumed to be sufficiently large, in which case the ML estimate asymptotically converges to an unbiased estimate and achieves the lower bound of the Cramér–Rao inequality (8), as aforementioned. Hence, from Eqs. (8) and (12), the error $\hat{\varepsilon}$ satisfies

$$\hat{\varepsilon} \;\to\; \frac{1}{\mathscr{I}(a)^{1/2}} \;\geq\; \frac{\sqrt{a(1-a)}}{N_q}. \qquad (13)$$

(More precisely, $\hat{\varepsilon}\,\mathscr{I}(a)^{1/2} \to 1$.) That is, the lower bound of the estimation error is on the order of $\mathscr{O}(N_q^{-1})$, which is referred to as the Heisenberg limit. This is in stark contrast to the classical sampling method, the estimation error of which is lower bounded by $\sqrt{a(1-a)}/N_q^{1/2}$, obtained by setting $m_k = 0 \;\forall k$ (i.e., a case with no amplitude amplification) in Eqs. (10) and (11); that is, the lower bound is at best on the order of $\mathscr{O}(N_q^{-1/2})$ in the classical case.

Now, we can consider the problem posed at the beginning of this subsection: designing the sequences $\{m_k, N_k\}$ so that the resulting ML estimate $\hat{\theta}_a$ outperforms the classical limit $\mathscr{O}(N_q^{-1/2})$ and hopefully achieves the Heisenberg limit $\mathscr{O}(N_q^{-1})$, i.e., the quantum quadratic speedup. Although the problem can be formulated as a maximization problem of Fisher information (10) with respect to $\{m_k, N_k\}$ under some constraints on these variables, here we fix N_k's to a constant and provide just two examples of the sequence $\{m_k\}$:

– Linearly incremental sequence (LIS): $N_k = N_{\text{shot}}$ for all k, and $m_k = k$, i.e., it increases as $m_0 = 0, m_1 = 1, m_2 = 2, \ldots, m_M = M$.
– Exponentially incremental sequence (EIS): $N_k = N_{\text{shot}}$ for all k, and m_k increases as $m_0 = 0, m_1 = 2^0, m_2 = 2^1, \ldots, m_M = 2^{(M-1)}$.

In the case of LIS, the Fisher information (7) and the number of queries (11) are calculated as $\mathscr{I}(a) = N_{\text{shot}}(2M+3)(2M+1)(M+1)/(3a(1-a))$ and $N_q = N_{\text{shot}}(M+1)^2$, respectively. Because $N_q \sim N_{\text{shot}}M^2$ and $\mathscr{I}(a) \sim N_{\text{shot}}M^3/(3a(1-a))$ when $M \gg 1$, the lower bound of the estimation error is evaluated as $\hat{\varepsilon} = 1/\mathscr{I}(a)^{1/2} \sim N_q^{-3/4}$; hence, a distinct quantum advantage occurs, although it does not reach the Heisenberg limit. Next for the case of EIS, we find $N_q \sim N_{\text{shot}}2^{M+1}$ and $\mathscr{I}(a) \sim N_{\text{shot}}2^{2(M+1)}/3$, which as a result lead to $\hat{\varepsilon} \sim N_q^{-1}$. Therefore, this choice is asymptotically optimal; we again emphasize that the statistical method certainly serves as a guide for us to find an optimal sequence $\{m_k\}$, achieving an optimal quantum amplitude estimation algorithm. But note that these quantum advantages are guaranteed only in the asymptotic

regime and that the realistic performance with the finite (or rather short) circuit depth should be analyzed. We will carry out a numerical simulation to see this realistic case in the following.

3.3 Numerical simulation

In this section, the ML estimates $\hat{\theta}_a$ and errors $\hat{\varepsilon}$ are evaluated numerically for several fixed target probabilities $a = \sin^2 \theta_a$. Based on the chosen sequence of $\{N_k\}$ and $\{m_k\}$ shown in the previous subsection, h_k's in Eq. (5) are generated using the Bernoulli sampling with probability $\sin^2((2m_k + 1)\theta_a)$ for each k. The global maximum of the likelihood function can be obtained by using a modified brute-force search algorithm; the global maximum of $\prod_{k=0}^{m} L_k(h_k; \theta_a)$ is determined by searching around the vicinity of the estimated global maximum for $\prod_{k=0}^{m-1} L_k(h_k; \theta_a)$. The errors $\hat{\varepsilon}$ are evaluated by repeating the aforementioned procedures 1000 times for each N_q.

In Fig. 2, the relationship between the number of queries and errors is plotted for the target probabilities $a = \sin^2 \theta_a = 2/3, 1/3, 1/6, 1/12, 1/24$, and $1/48$ with $N_{\text{shot}} = 100$. The (red) triangles and (black) circles in Fig. 2 are errors that are obtained using LIS and EIS, respectively. For comparison, numerical simulations with $m_k = 0$ for all k are also performed, and the results are plotted as (blue) squares in Fig. 2. In addition, the lower bounds of errors (13) when the estimate is not biased are also plotted as (red) dotted and (black) solid lines for LIS and EIS, respectively. The (blue) dashed lines in Fig. 2 are the lower bounds for classical random sampling, i.e., $\sqrt{a(1-a)/N_q}$.

The slopes of the simulated results with the target probability $a = \sin^2 \theta_a = 1/48$ ranging from $N_q \simeq 10^3$ to $N_q \simeq 10^5$ in Fig. 2 are fitted by $\log \hat{\varepsilon} = \gamma \cdot \log N_q + \delta$, and the fitted parameters corresponding to the slope are obtained as $\gamma = -0.76$, $\gamma = -0.95$ and $\gamma = -0.50$ for LIS and EIS, and classical random sampling, respectively. Similar slopes are obtained with other target probabilities. The fitted values of γ for LIS and EIS are consistent with the slopes obtained using the Fisher information, although γ slightly deviated from the theoretical values. This slight deviation indicates that $\hat{a}$ is a biased estimate; in fact, this deviation decreases as N_{shot} increases, which is consistent with the fact that, in general, the ML estimate becomes unbiased asymptotically as the sampling number increases. Also, the efficiency of the ML estimate can be observed in the numerical simulation; the estimation error approaches the Cramér–Rao lower bound (13). In Appendix A, we show the comparison of the error for the conventional phase-estimation-based approach with that of EIS. As a result, their estimation errors are found to be comparable.

Finally, we remark that the computational complexity for naively finding the maximum of the likelihood function is on the order of $\mathcal{O}((1/\varepsilon) \ln(1/\varepsilon))$ if m_k exponentially grows, as in EIS. This is because the computational complexity to obtain the likelihood function $\ln L(\mathbf{h}; \theta_a)$ is evaluated as $\mathcal{O}(M)$ in this case. The order of the error ε is estimated as $\mathcal{O}(N_q^{-1})$ based on the Cramér–Rao bound. Because $N_q \sim 2^M N_{\text{shot}}$, the complexity of evaluating the likelihood function is $\mathcal{O}(\ln(1/\varepsilon))$. Assuming that the brute-force search among $1/\varepsilon$ segments is performed to find the global maximum of the likelihood function, the complexity of finding the maximum is $\mathcal{O}((1/\varepsilon) \ln(1/\varepsilon))$. In the case of LIS, the order of the computational complexity can also be evaluated
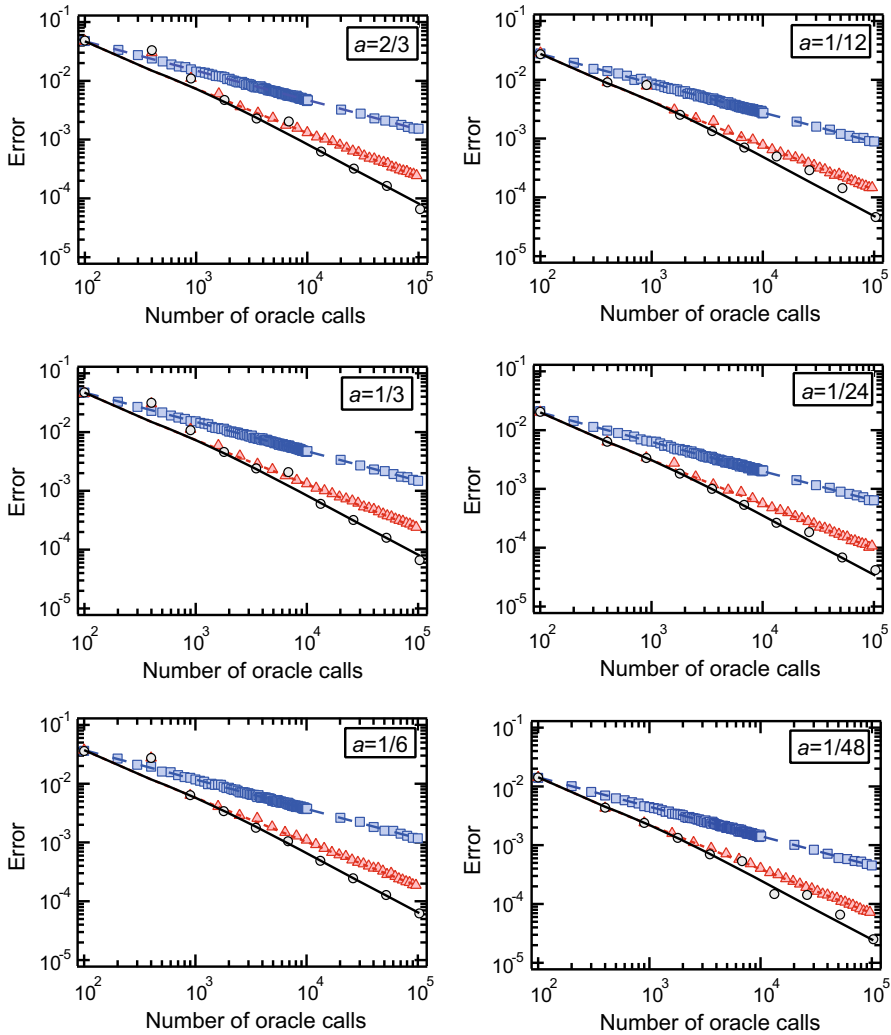
Fig. 2 Relationships between the number of queries and the estimation error for several target probabilities $a = \sin^2 \theta_a$. The lower bounds of estimation error based on the Cramér–Rao inequality are depicted as lines, the (blue) dashed line is for $m_k = 0$ for all k (classical random sampling), the (red) dotted line is for $m_0 = 0, m_1 = 1, \ldots m_M = M$ (LIS), and the (black) solid line is for $m_0 = 0, m_1 = 2^0, \ldots m_M = 2^{M-1}$ (EIS), respectively. The estimation errors obtained by numerical simulations are also plotted as symbols, the (blue) squares are for classical random sampling, the (red) triangles are for LIS, and the (black) circles are for EIS (Color figure online)

as $\mathcal{O}(\varepsilon^{-5/3})$ in the same manner as before. It should be noted that the brute-force search algorithm for finding global minima of $\prod_{k=0}^{M} L_k(h_k; \theta_a)$ is not necessary if m_k is zero for all k (classical case), since the target value is simply obtained by $\hat{a} = \sum_{k=0}^{M} h_k / \sum_{k=0}^{M} N_{\text{shot}}$. The error can be obtained as $\mathcal{O}(N_q^{-1/2})$ based on the Cramér–Rao bound. Due to the fact that $N_q = N_{\text{shot}} M$, the computational complexity

Table 1 Summary of the complexities for estimating target value with given error ε. The query complexity and computational complexity of post-processing for different update rules of m_k are listed

Update rule of m_k	Query complexity	Computational complexity of post-processing
Classical ($m_k = 0 \forall k$)	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$
Linearly incremental sequence (LIS) ($m_0 = 0, m_1 = 1, m_2 = 2, \ldots, m_M = M$)	$\mathcal{O}(\varepsilon^{-4/3})$	$\mathcal{O}(\varepsilon^{-5/3})$
Exponentially incremental sequence (EIS) ($m_0 = 0, m_1 = 2^0, m_2 = 2^1, \ldots, m_M = 2^{(M-1)}$)	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-1} \ln \varepsilon^{-1})$

in the classical case is $\mathcal{O}(\varepsilon^{-2})$. The evaluated computational complexities of post-processing for different update rules of m_k are summarized together with the query complexities in Table 1.

4 Application to the Monte Carlo integration

We conduct a Monte Carlo integration as an example of the application of our algorithm, as follows. In this section, we first review the quantum algorithm to calculate the Monte Carlo integration by amplitude estimation [15] and then explain the amplitude amplification operator used in our algorithm. Next, we present the integral of the sine function as a simple example of Monte Carlo integration. Using this example, we discuss the number of CNOT gates and qubits required for our algorithm and the conventional amplitude estimation [16].

4.1 The Monte Carlo integration as an amplitude estimation

One purpose of the Monte Carlo integration is to calculate the expected value of real-valued function $0 \le f(x) \le 1$ defined for n-bit input $x \in \{0, 1\}^n$ with probability $p(x)$:

$$\mathbb{E}[f(x)] = \sum_{x=0}^{2^n-1} p(x) f(x). \tag{14}$$

In the quantum algorithm for the Monte Carlo integration, an additional (ancilla) qubit is introduced and assumed to be rotated as

$$\mathcal{R}|x\rangle_n |0\rangle = |x\rangle_n \left(\sqrt{f(x)}|1\rangle + \sqrt{1 - f(x)}|0\rangle \right), \tag{15}$$

where $\mathcal{R}$ is a unitary operator acting on $n + 1$ qubits. In addition, an algorithm $\mathcal{P}$ is introduced, and operating $\mathcal{P}$ to n-qubit resister $|0\rangle_n$ yields

Fig. 3 Quantum circuit of
amplitude amplification for the
Monte Carlo integration
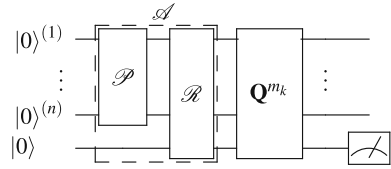
$$\mathscr{P}|0\rangle_n = \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle_n, \tag{16}$$

where all qubits in $|0\rangle_n$ are in the state $|0\rangle$. Operating $\mathscr{R}(\mathscr{P} \otimes \mathbf{I}_1)$ to the state $|0\rangle_n|0\rangle$ generates $|\Psi\rangle$:

$$|\Psi\rangle = \mathscr{R}(\mathscr{P} \otimes \mathbf{I}_1)|0\rangle_n|0\rangle \tag{17}$$

$$= \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle_n \left(\sqrt{f(x)}|1\rangle + \sqrt{1-f(x)}|0\rangle \right), \tag{18}$$

where $\mathbf{I}_1$ is the identity operator acting on an ancilla qubit. For convenience, we put $a = \sum_{x=0}^{2^n-1} p(x)f(x)$ and introduce two orthonormal bases:

$$|\tilde{\Psi}_1\rangle = \frac{1}{\sqrt{a}} \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{f(x)}|x\rangle_n|1\rangle, \tag{19}$$

$$|\tilde{\Psi}_0\rangle = \frac{1}{\sqrt{1-a}} \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{1-f(x)}|x\rangle_n|0\rangle. \tag{20}$$

By using these bases, the state $|\Psi\rangle$ can be rewritten as

$$|\Psi\rangle = \sqrt{a}|\tilde{\Psi}_1\rangle + \sqrt{1-a}|\tilde{\Psi}_0\rangle. \tag{21}$$

Then, the square root of expected value $a = \mathbb{E}[f(x)]$ appears in the amplitude of $|\tilde{\Psi}_1\rangle$, and the Monte Carlo integration can be regarded as an amplitude estimation of $|\tilde{\Psi}_1\rangle$. The operator $\mathbf{Q}$ defined in Eq. (1) can be achieved using $\mathbf{U}_\Psi \mathbf{U}_{\tilde{\Psi}_0}$, where $\mathbf{U}_{\tilde{\Psi}_0} = \mathbf{I} - 2|\tilde{\Psi}_0\rangle\langle\tilde{\Psi}_0|$, $\mathbf{U}_\Psi = \mathbf{I} - 2|\Psi\rangle\langle\Psi|$, and $\mathbf{I}$ is the identity acting on $n+1$ qubits [16]. In terms of a practical point of view, we use $\mathbf{U}_{\tilde{\Psi}_0} = \mathbf{I}_{n+1} - 2\mathbf{I}_n|0\rangle\langle0|$, where $\mathbf{I} = \mathbf{I}_{n+1} = \mathbf{I}_n \otimes (|0\rangle\langle0| + |1\rangle\langle1|)$. By putting $a = \sin^2\theta_a$ and using Eq. (3), we could apply our algorithm to the Monte Carlo integration. The circuit diagram of the amplitude amplification used in our algorithm is shown in Fig. 3. Note that the multi-qubit gate consisting of $\mathscr{P}$ and $\mathscr{R}$ in Fig. 3 corresponds to the quantum algorithm $\mathscr{A}$ shown in Sect. 2, and the only ancilla qubit for each k is measured when our algorithm is applied to the Monte Carlo. Similarly, the circuit of the conventional amplitude estimation [16] is shown in Fig. 4. In the following, we applied our algorithm to a very simple integral of the sine function and compared the number of CNOT gates and qubits with the results of the conventional amplitude estimation.
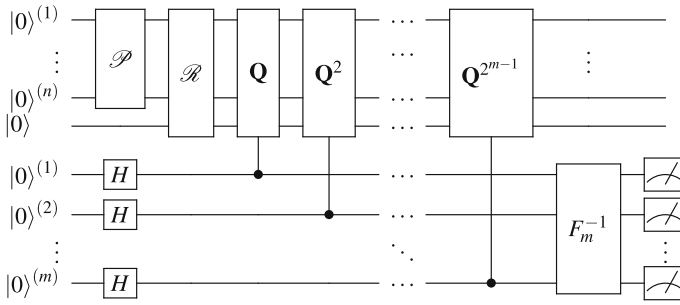
Fig. 4 Quantum circuit of conventional amplitude estimation for the Monte Carlo integration. F_m^{-1} represents the inverse QFT of m qubits

4.2 Simple example: integral of the sine function

As a simple example of the Monte Carlo integration, the following integral is considered.

$$I = \frac{1}{b_{\max}} \int_0^{b_{\max}} \sin(x)^2 dx, \tag{22}$$

where $b_{\max}$ is a constant that determines the upper limit of the integral. By discretizing this integral in n-qubit, we obtain

$$S = \sum_{x=0}^{2^n - 1} p(x) \sin^2 \left(\frac{\left(x + \frac{1}{2}\right) b_{\max}}{2^n} \right), \tag{23}$$

where $p(x) = \frac{1}{2^n}$ is a discrete uniform probability distribution. We now explicitly describe the operators $\mathscr{P}$ and $\mathscr{R}$ for applying our algorithm to calculate sum (23). The operator $\mathscr{P}$ acting on the n-qubit initial state can be defined as

$$\mathscr{P} : |0\rangle_n |0\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n - 1} |x\rangle_n |0\rangle. \tag{24}$$

The operator $\mathscr{P}$ can be constructed using n Hadamard gates. The operator $\mathscr{R}$ acting on the $(n + 1)$-qubit state $|x\rangle_n |0\rangle$ can be defined as
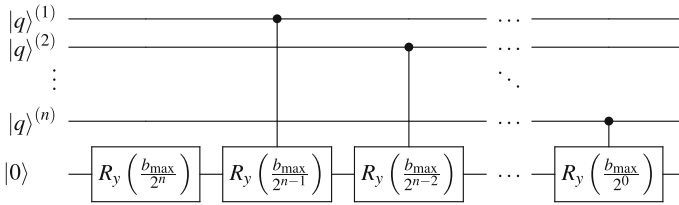
Fig. 5 Quantum circuit achieving the operator $\mathscr{R}$ in Eq. (25). In this circuit, $|x\rangle$ in Eq. (25) is represented by n qubits, denoted by $|q\rangle^{(1)}, |q\rangle^{(2)}, \ldots, |q\rangle^{(n)}$. $R_y(\theta)$ represents a Y-rotation with angle θ
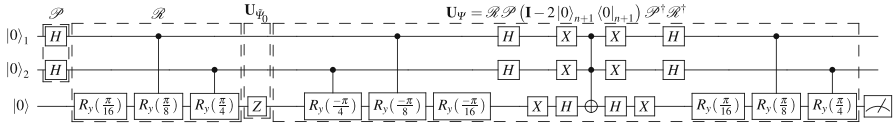

Fig. 6 Quantum circuit of amplitude amplification in the case of $n = 2$ with single **Q** operation
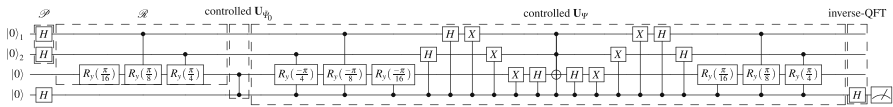

Fig. 7 Quantum circuit of conventional amplitude estimation in the case of $n = 2$ with a single **Q** operation

$$\mathscr{R} : |x\rangle_n |0\rangle \rightarrow |x\rangle_n \left(\sin\left(\frac{\left(x + \frac{1}{2}\right) b_{\max}}{2^n} \right) |1\rangle + \cos\left(\frac{\left(x + \frac{1}{2}\right) b_{\max}}{2^n} \right) |0\rangle \right). \quad (25)$$

The operator $\mathscr{R}$ can be constructed using controlled Y-rotations as illustrated in Fig. 5.

We now explicitly show an example of the circuits for the amplitude amplification used in our algorithm and a conventional amplitude estimation with a single **Q** operation, which calculates sum (23). For simplicity, the circuit for $b_{\max} = \pi/4$ and $n = 2$ is shown here. The quantum circuits for amplitude amplification and conventional amplitude estimation are shown in Figs. 6 and 7, respectively. In these circuits, all-to-all qubit connectivity is assumed. From these figures, we can see that the circuit for conventional amplitude estimation tends to have more gates and qubits than that of our algorithm. Furthermore, the multi-controlled operation in the conventional amplitude estimation circuit of Fig. 7 may require several ancilla qubits.

Table 2 shows the number of CNOT gates and qubits as a function of the number of **Q** operators required for conventional amplitude estimation and our algorithm. Here, we assume the gate set supported by Qiskit ver. 0.7 [29]. Because the number of CNOT gate operations is restricted in NISQ devices due to the error accumulation, the numbers of CNOT gates in our algorithm only those for the circuit with the largest m_k are evaluated. The numbers of CNOT gates in our algorithm are about 7–18 times smaller than those of conventional amplitude estimation. The number of qubits required for conventional amplitude estimation increases as the number of **Q** operations increased, while that for our algorithm kept constant. The source code for Monte Carlo integration based on our proposed algorithm is available at [30].

Table 2 Number of CNOT gates and qubits to calculate (23) as a function of **Q** operations

# Operators **Q**	Conventional amplitude estimation		Our algorithm	
	# CNOT gates	# Qubits	# CNOT gates	# Qubits
0	–	–	4	3
2^0	135	7	18	3
2^1	399	8	32	3
2^2	927	9	60	3
2^3	1981	10	116	3
2^4	4085	11	228	3
2^5	8287	12	452	3
2^6	16,683	13	900	3
2^7	33,465	14	1796	3
2^8	67,017	15	3588	3

5 Conclusion

We proposed a quantum amplitude estimation algorithm achieving quantum speedup by reducing controlled gates with ML estimation. The essential idea of the proposed algorithm is constructing a likelihood function using the outcomes of measurements on several quantum states, which are transformed by the amplitude amplification process. Although the probability measuring good or bad states depends on the number of amplitude amplification operations, the outcomes are correlated due to the fact that each amplified probability is a function of a single parameter. To test the efficiency of the proposed algorithm, we performed numerical simulations and analyzed the relationships between the number of queries and estimation error. Empirical evidences showed the algorithm could estimate the target value with fewer queries than the classical algorithm. We also presented the lower bound of the estimation error in terms of the Fisher information and found that the estimation error observed in a numerical simulation was sufficiently close to the Heisenberg limit. In addition, we experimented the proposed algorithm for a Monte Carlo integration and found that fewer CNOT gates and qubits were required in comparison with the conventional amplitude estimation. These facts indicate that our algorithm could work well even with noisy intermediate-scale quantum devices.

Shortly after the publication of our results, simplified quantum counting and amplitude estimation without QFT with rigorous proofs were shown [31]. In contrast to our approach that can be run in parallel on multiple quantum devices, the simplified algorithms are adaptive and have to be run sequentially. They also require large constant-factor overhead, e.g., millions of measurement samples, which could be expensive in practice. Nevertheless, there are several interesting directions for future work as pointed out in [31], such as, obtaining rigorous proofs for our parallel approach and achieving quantum speedups with depth-limited quantum circuits.

Appendix A: Comparison of estimation errors with conventional amplitude estimation

We compare the estimation error between the conventional amplitude estimation algorithm and our proposed algorithm. The details of conventional amplitude estimation algorithm are presented in Ref. [16]. For simplicity, only the result of $a = \sin^2 \theta_a = 1/48$ is shown here.

Figure 8 shows the relations between the number of queries and estimation error for the conventional amplitude estimation and our proposed algorithm. In the figure, the (black) circles, which represent the data of conventional algorithm, are generated in the following manner. The conventional algorithm outputs four integers closest to the target value $\theta_a M/\pi$ and $M - \theta_a M/\pi$ with success probability of at least $8/\pi^2 \times 100\% \sim 81\%$ after $M = 2^m - 1$ times application of controlled $\mathbf{Q}$ operation followed by QFT [16]. The largest estimation error calculated from these four integers is plotted in Fig. 8. The (red) triangles and (blue) squares represent the data of our proposed
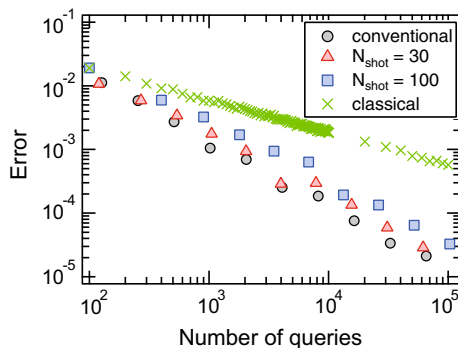

Fig. 8 The relationship between the number of queries and estimation error, for our proposed algorithm and the conventional amplitude estimation algorithm [16]. The (black) circles are generated from the conventional phase-estimation-based approach. The (red) triangles and (blue) squares are the 81 percentile values of estimation error with numerical simulations for $N_{\text{shot}} = 30$ and $N_{\text{shot}} = 100$ for EIS, respectively. For comparison, the 81 percentile values of estimation error with classical sampling are also shown as (green) crosses. Against a fixed total number of queries, the smaller N_{shot} becomes, the more queries for the quantum amplitude amplification are used and thereby the estimation error approaches more closely to that of the conventional algorithm (Color figure online)

method with $N_{shot} = 30$ and 100, respectively. The $8/\pi^2 \times 100 \sim 81$ percentile of the estimation error is plotted here for a fair comparison with the conventional algorithm, while the averaged error is described in the main text. The data of our algorithm is generated by the same manner as in Sect. 3.3.

This figure shows that the estimation error of our proposed method increases as the number of shots increases. This is because, as can be seen from Eqs. (10) and (11), the degree of quantum speedup becomes relatively smaller by increasing the number of shots to the limit that it is essentially a classical sampling where the number of shots is equal to the total number of queries. The figure shows that the estimation error of the conventional algorithm is almost the same as that of ours with $N_{shot} = 30$.

References

1. IBM Q Experience: https://quantumexperience.ng.bluemix.net/qx/editor (2019). Accessed 26 Mar 2019
2. Friis, N., Marty, O., Maier, C., Hempel, C., Holzäpfel, M., Jurcevic, P., Plenio, M.B., Huber, M., Roos, C., Blatt, R., Lanyon, B.: Observation of entangled states of a fully controlled 20-qubit system. Phys. Rev. X **8**, 021012 (2018)
3. Song, C., Xu, K., Liu, W., Yang, Cp, Zheng, S.B., Deng, H., Xie, Q., Huang, K., Guo, Q., Zhang, L., Zhang, P., Xu, D., Zheng, D., Zhu, X., Wang, H., Chen, Y.A., Lu, C.Y., Han, S., Pan, J.W.: 10-qubit entanglement and parallel logic operations with a superconducting circuit. Phys. Rev. Lett. **119**, 180511 (2017)
4. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018)
5. McClean, J.R., Romero, J., Babbush, R., Aspuru-Guzik, A.: The theory of variational hybrid quantum-classical algorithms. New J. Phys. **18**, 023023 (2016)
6. Yung, M.H., Casanova, J., Mezzacapo, A., McClean, J., Lamata, L., Aspuru-Guzik, A., Solano, E.: From transistor to trapped-ion computers for quantum chemistry. Sci. Rep. **4**, 3589 (2014)
7. Knill, E., Ortiz, G., Somma, R.D.: Optimal quantum measurements of expectation values of observables. Phys. Rev. A **75**, 012328 (2007)
8. Kassala, I., Jordan, S.P., Lovec, P.J., Mohsenia, M., Aspuru-Guzik, A.: Polynomial-time quantum algorithm for the simulation of chemical dynamics. Proc. Natl. Acad. Sci. USA **105**, 18681–18686 (2008)
9. Rebentrost, P., Gupt, B., Bromley, T.R.: Quantum computational finance: Monte Carlo pricing of financial derivatives. Phys. Rev. A **98**, 022321 (2018)
10. Woerner, S., Egger, D.J.: Quantum risk analysis. npj Quantum Inf. **5**, 15 (2019)
11. Wiebe, N., Kapoor, A., Svore, K.M.: Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. Quantum Inf. Comput. **15**, 316–356 (2015)
12. Wiebe, N., Kapoor, A., Svore, K.M.: Quantum deep learning. Quantum Inf. Comput. **16**, 541–587 (2016)
13. Wiebe, N., Kapoor, A., Svore, K.M.: Quantum perceptron models. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 4006–4014 (2016)
14. Kerenidis, I., Landman, J., Luongo, A., Prakash, A.: q-means: A quantum algorithm for unsupervised machine learning. arXiv:1812.03584 (2018)
15. Montanaro, A.: Quantum speedup of Monte Carlo methods. Proc. Royal Soc. A **471**, 20150301 (2015)
16. Brassard, G., Høyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemp. Math. Ser. Millenn. **305**, 53–74 (2002)
17. Kitaev, A.Y.: Quantum measurements and the Abelian stabilizer problem. Electron. Colloq. Comput. Complex. arXiv:quant-ph/9511026 (1996)
18. Svore, K.M., Hastings, M.B., Freedman, M.: Faster phase estimation. Quantum Inf. Comput. **14**, 306–328 (2014)
19. Wiebe, N., Granade, C.: Efficient Bayesian phase estimation. Phys. Rev. Lett. **117**, 010503 (2016)
20. O'Brien, T.E., Tarasinski, B., Terhal, B.M.: Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments. New J. Phys. **21**, 023022 (2019)

21. van den Berg, E.: Practical sampling schemes for quantum phase estimation. arXiv:1902.11168 (2019)
22. Wie, C.R.: Simpler quantum counting. arXiv:1907.08119 (2019)
23. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of 28th Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)
24. Abrams, D.S., Williams, C.P.: Fast quantum algorithms for numerical integrals and stochastic processes. arXiv:quant-ph/9908083 (1999)
25. Zintchenko, I., Wiebe, N.: Randomized gap and amplitude estimation. Phys. Rev. A **93**, 062306 (2016)
26. Brassard, G., Høyer, P.: An exact quantum polynomial-time algorithm for Simon's problem. In: Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems, pp. 12–23 (1997)
27. Grover, L.K.: Quantum computers can search rapidly by using almost any transformation. Phys. Rev. Lett. **80**, 4329–4332 (1998)
28. Rao, C.R.: Linear Statistical Inference and its Applications, vol. 2. Wiley, New York (1973)
29. Aleksandrowicz, G., Alexander, T., Barkoutsos, P., Bello, L., Ben-Haim, Y., Bucher, D., Cabrera-Hernádez, F.J., Carballo-Franquis, J., Chen, A., Chen, C.F., Chow, J.M., Córcoles-Gonzales, A.D., Cross, A.J., Cross, A., Cruz-Benito, J., Culver, C., González, S.D.L.P., Torre, E.D.L., Ding, D., Dumitrescu, E., Duran, I., Eendebak, P., Everitt, M., Sertage, I.F., Frisch, A., Fuhrer, A., Gambetta, J., Gago, B.G., Gomez-Mosquera, J., Greenberg, D., Hamamura, I., Havlicek, V., Hellmers, J., Herok, Ł., Horii, H., Hu, S., Imamichi, T., Itoko, T., Javadi-Abhari, A., Kanazawa, N., Karazeev, A., Krsulich, K., Liu, P., Luh, Y., Maeng, Y., Marques, M., Martín-Fernández, F.J., McClure, D.T., McKay, D., Meesala, S., Mezzacapo, A., Moll, N., Rodríguez, D.M., Nannicini, G., Nation, P., Ollitrault, P., O'Riordan, L.J., Paik, H., Pérez, J., Phan, A., Pistoia, M., Prutyanov, V., Reuter, M., Rice, J., Davila, A.R., Rudy, R.H.P., Ryu, M., Sathaye, N., Schnabel, C., Schoute, E., Setia, K., Shi, Y., Silva, A., Siraichi, Y., Sivarajah, S., Smolin, J.A., Soeken, M., Takahashi, H., Tavernelli, I., Taylor, C., Taylour, P., Trabing, K., Treinish, M., Turner, W., Vogt-Lee, D., Vuillot, C., Wildstrom, J.A., Wilson, J., Winston, E., Wood, C., Wood, S., Wörner, S., Akhalwaya, I.Y., Zoufal, C.: Qiskit: An open-source framework for quantum computing (2019). https://doi.org/10.5281/zenodo.2562110
30. Qiskit Community Tutorials: Amplitude estimation without quantum Fourier transform and controlled grover operators. https://github.com/Qiskit/qiskit-community-tutorials/blob/master/algorithms/SimpleIntegral_AEwoPE.ipynb (2019). Accessed 30 Oct 2019
31. Aaronson, S., Rall, P.: Quantum approximate counting, simplified. arXiv:1908.10846 (2019)

Ⓐ Springer