



OPEN

A general quantum algorithm for numerical integration

Guoqiang Shu^{1,2}, Zheng Shan^{1,2}, Jinchen Xu^{1,2}, Jie Zhao¹ & Shuya Wang¹

Quantum algorithms have shown their superiority in many application fields. However, a general quantum algorithm for numerical integration, an indispensable tool for processing sophisticated science and engineering issues, is still missing. Here, we first proposed a quantum integration algorithm suitable for any continuous functions that can be approximated by polynomials. More impressively, the algorithm achieves quantum encoding of any integrable functions through polynomial approximation, then constructs a quantum oracle to mark the number of points in the integration area and finally converts the statistical results into the phase angle in the amplitude of the superposition state. The quantum algorithm introduced in this work exhibits quadratic acceleration over the classical integration algorithms by reducing computational complexity from $O(N)$ to $O(\sqrt{N})$. Our work addresses the crucial impediments for improving the generality of quantum integration algorithm, which provides a meaningful guidance for expanding the superiority of quantum computing.

Keywords Quantum algorithm, Numerical integration, Phase estimation, Quadratic acceleration

Numerical integration is a classical and important problem with a wide range of applications in many fields of physics, chemistry, biology, computer and finance^{1–3}. However, existing classical integration algorithms face high computational complexity, $O(n)$, where n represents the number of points used for integration calculation over the interval. Fortunately, quantum algorithms exhibit acceleration superiority over classical algorithms in some problems. For example, Shor's integer factorization algorithm, which shows exponential acceleration over classical algorithms⁴. Grover's algorithm has quadratic acceleration over the classical algorithms in solving unordered database search problems⁵. The HHL algorithm for solving systems of linear equations also exhibits exponential acceleration and is widely used in machine learning and other fields⁶. Nevertheless, a general quantum algorithm for numerical integrations is still absent.

Definite integrations are commonly approximated by classical numerical methods. For example, interpolation integration formulas include Newton–Cotes formula, Complex quadrature formula, Romberg formula and Gauss formula^{7–9}. The algebraic accuracy of Newton–Cotes formula and Complex quadrature formula is n , where n is the number of interpolation nodes. But when $n > 7$, these methods lose their effectiveness. The algebraic accuracy of Romberg formula is $2n$, and the algebraic accuracy of Gauss formula is $2n + 1$. These interpolation integration formulas utilize the n -order polynomials value at n interpolation points to approximate the integrations, and their complexity is $O(n)$. Besides that, the monte carlo integration (MCI) method generates n random numbers in the integration area S . The number of random numbers in the integration area is m , thus the integration is approximately equal to mS/n . The complexity of MCI method is $O(n)$ and error is $O(1/\sqrt{n})$ ¹⁰, which the error can be reduced by key sampling and Quasi-monte carlo method, etc.^{11,12}. R. P. Kanwal et al. presented an integration method using Taylor expansion with the precision n ^{13,14} and complexity $O(n)$. In all, the complexity of these classical algorithms is $O(n)$ that is high compared with quantum algorithms.

For taking advantage of quantum acceleration to reduce the complexity of classical algorithms, some quantum algorithms were proposed. For example, Acioli et al. proposed a Quantum Monte Carlo (QMC)^{15,16} integration algorithm with quadratic acceleration for periodic functions¹⁷. Abrams et al. proposed a fast quantum integration algorithm by using the Grover's mean and quantum counting, which obtains exponential speed acceleration and quadratic acceleration in comparison with classical deterministic and probabilistic methods respectively, however, the function is required to be discrete Boolean type in this method¹⁸.

Shimada et al. presented the quantum coin algorithm based on the quantum supersampling algorithm achieving quadratic acceleration^{19,20}, but the value of the functions is limited to $[0,1]$. Heinrich raised a quantum integration algorithm with quadratic acceleration for Sobolev-like high-dimensional function²¹. DeWitt-Morette et al. proved a quantum algorithm with exponential acceleration over deterministic classical algorithms on the

¹Laboratory for Advanced Computing and Intelligence Engineering, Zhengzhou 450001, China. ²These authors contributed equally: Guoqiang Shu, Zheng Shan and Jinchen Xu. ✉email: sstronger21@163.com

functions of Holder class²². Heinrich proposed a quantum integration algorithm on the Lebesgue space, which proves the algorithm is optimal²³. Rebentrost et al. presented an optical quantum multi-dimensional integration algorithm, which demonstrates quadratic acceleration over MC method²⁴. These algorithms mentioned above have limitations on the type of integration function. In addition, some quantum and classical hybrid integration algorithms have also been proposed, such as Suzuki et al. raised a hybrid integration method to simplify the circuit depth^{25–27}. Moreover, some are controversial over acceleration capability, such as the MCI method based on quantum amplitude estimation (QAE) can bring quadratic acceleration^{28,29}, but Kaneko et al. confirmed that the probability distribution of the initial state coding prepared by the Grover-Rudolph method does not reflect the quantum advantage³⁰.

Existing classical integration algorithms face high computational complexity and quantum algorithms face low generality problems, as shown in Table 1. Here, we propose a general quantum integration algorithm (GQIA) that eliminates these shortages showing strong universality and operability. Firstly, our algorithm quantizes the classical Monte Carlo integration process and utilizes quantum superposition to possess exponential representation capabilities beyond classical methods. Secondly, to construct a quantum oracle with polynomial approximation of the integration function, use the parallelism of quantum to count the points on the integration region, and store them in the phase of the quantum state. Finally, amplitude amplification and phase estimation are used to obtain phase information with high probability and accuracy, whereupon the integration value is calculated. Compared with classical integration algorithms, GQIA demonstrates the quadratic acceleration and higher computational accuracy. Due to the use of Monte Carlo and polynomial approximations, GQIA has no limitations on the formal properties of integration functions, making it more versatile than other quantum algorithms.

Methods

For more comprehensive discussions of GQIA, we begin here by briefly reviewing some relevant results from quantum algorithm and quantum computation theory.

Amplitude amplification

The key idea of the amplitude amplification (AA) algorithm²⁴ originally came from an unordered database-search algorithm, known as Grover's quantum algorithm⁵. By amplifying the amplitude of a given pure state, the algorithm achieves the purpose of adjusting the measurement probability of the pure state. The specific implementation method of the algorithm is given as follows, first given an oracle \hat{A} and initial state $|0\rangle^{\otimes n}$

$$|\psi\rangle = \hat{A}|0\rangle^{\otimes n} = \hat{A}|0\dots 0\rangle = \cos\theta/2|\psi_0\rangle + \sin\theta/2|\psi_1\rangle \quad (1)$$

Quantum algorithm			
	Suitable area	Complexity	Precision
GQIA	Any continuous function	$O((1/\epsilon)\sqrt{N/M})$	$O\left(\frac{\sqrt{N}-1}{\sqrt{N}(x^{d+1})\epsilon}\right)^1$
Algorithm-1 ³¹	Holder function	$O((\log\epsilon^{-1})^{1/(1+\gamma)})$	$O(n^{-\alpha/(d-1)})^2$
Algorithm-2 ²¹	Sobolev function	$O(n)$	$O(1/n^{-r/(d-1)})^3$
Algorithm-3 ²³	Lebesgue function	$O\left((1/\epsilon)^{\frac{p}{2(p-1)}}\right)$, $1 \leq p < 2$; $O((1/\epsilon))$, $2 \leq p \leq \infty$	$O(n^{-2+2/p})$
Algorithm-4 ¹⁸	Any continuous function	$O(n)$	$O(n^{-1})$
Photonic quantum ²⁴	Multi-dimensional integrations	$O((1/\epsilon))$	$O(1/\epsilon^\delta)^4$
Quantum monte carlo ¹⁷	Periodic function	$O(n^{2-\delta})$, $0 \leq \delta \leq 1$	$O(1/\sqrt{n})$
Quantum coin metho ²⁰	Function takes the value [0,1]	$O(n)$	$O(1/\sqrt{n})$
Quantum supersampling ¹⁹	Boolean Function	$O(n)$	$O(1/n)$
Classical Algorithm			
	Suitable Area	Complexity	Precision
Gauss forum ^{7–9}	High precision with fewer nodes	$O(n)$	$O(1/x^{2n+1})$
Taylor expansion ^{13,14}	Each derivative of the function is known	$O(n)$	$O(1/x^n)$
Romberg forum ^{7–9}	less computation, high precision requirements	$O(n^k)$	$O(1/x^{2k})$
Monte carlo ¹⁰	Complex function or the form is unknown	$O(n)$	$O(1/\sqrt{n})$
Composite rule ^{7–9}	Degree < 8 and integration interval is large	$O(n)$	$O(1/x^n)$
Newton–cotes forum ^{7–9}	Degree < 8	$O(n)$	$O(1/x^n)$

Table 1. Performance analysis of different integration algorithms. ¹ N stands for the number of points of area S , $N = 2^{k_1+k_2}$, and has the same meaning as other n in the table. M represents the number of point in the area S_1 . ² α , d is real number. ³ r , d is real number. ⁴ δ is a constant that can be arbitrarily small.

where $\theta \in [0, \pi]$, $|\psi_1\rangle$ is the target pure state and $|\psi_0\rangle$ is the non-target pure state. The state vector $|\psi\rangle$ can be expressed as a parameterized vector $(\cos\theta/2, \sin\theta/2)$ on the space stretched by two basis vectors $|\psi_0\rangle$ and $|\psi_1\rangle$. The purpose is to promote the amplitude of the $|\psi_1\rangle$ first, while reducing the amplitude of the non-target state, because the sum of the squares of the two probability amplitudes is 1. At this time, the measurement probability of the target state will also increase. This is done by flipping the quantum state $|\psi\rangle$ towards the direction of the target state $|\psi_1\rangle$. i.e., making θ as much larger as possible.

In the AA algorithm, one amplitude amplification requires two flips. The first flip uses the unitary operation S_χ to flip the state $|\psi\rangle$ along $|\psi_0\rangle$ and change state $|\psi\rangle$ to $-|\psi\rangle$. The second flip uses the unitary operation $2|\psi\rangle\langle\psi| - I$ to flip the first result $S_\chi|\psi\rangle$ along $|\psi\rangle$. The amplitude of the final target state $|\psi_1\rangle$ becomes larger in the resulting state $(2|\psi\rangle\langle\psi| - I)S_\chi|\psi\rangle$.

The overall operation \hat{Q} is defined as

$$\begin{aligned} |\psi_{\text{result}}\rangle &= \hat{Q}|\psi\rangle = (2|\psi\rangle\langle\psi| - I)S_\chi|\psi\rangle \\ &= \left(2\hat{A}|00\dots 0\rangle\langle 00\dots 0| \hat{A}^{-1} - I\right)S_\chi|\psi\rangle \\ &= \hat{A}(2|00\dots 0\rangle\langle 00\dots 0| - I)\hat{A}^{-1}S_\chi|\psi\rangle \end{aligned}$$

where $(2|00\dots 0\rangle\langle 00\dots 0| - I)$ operation means to flipping the amplitude of all quantum states but keeping the amplitude of $|00\dots 0\rangle$ unchanged. By repeating this process many times, the quantum state $|\psi\rangle$ can be flipped to the target state $|\psi_1\rangle$ with an angle of θ each time. What we need to pay attention to is that the real value of θ does not need to be known in advance.

Phase estimation

Quantum phase estimation is an algorithm for estimating the phase information of quantum states, and it is the core of many quantum algorithms. Two quantum registers are necessary in the phase estimation circuit, which the first register requires t qubits whose initial state is $|0\rangle$ and the value of t depends on two aspects: the number of digits for precision and successful probability of phase estimation result.

The initial state of the second register is the prepared state $|u\rangle$ and the number of qubits is as many as possible to store $|u\rangle$. The process of phase estimation is roughly divided into three steps. First, the circuit begins with t Hadamard gates that are applied to all qubits in the first register, and the controlled- U operations are applied simultaneously on the second register, where the U -gate appears in consecutive integer powers of 2. The resulting state is

$$\frac{\left(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle\right)\dots\left(|0\rangle + e^{2\pi i 2^0\varphi}|1\rangle\right)}{2^{t/2}} = \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i \varphi k} |k\rangle \tag{2}$$

The second step of phase estimation is the application of inverse quantum Fourier transform on the first register, and this step can be accomplished in $O(t^2)$ stages. The third and final step of phase estimation is to measure the state in the first register and get an estimation of φ . The final state of the first register can be written as

$$\frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} e^{2\pi i \varphi j} |j\rangle |u\rangle \rightarrow |\tilde{\varphi}\rangle |u\rangle \tag{3}$$

where $\tilde{\varphi}$ denotes the estimator for φ when measured.

Proposed methods

A General quantum integration algorithm (GQIA)

We define S as the area consisting of integration area S_1 and non-integration area S_2 . N and M denote the number of points in S and S_1 respectively (Fig. 1a). The core idea of GQIA is to convert a definite integration problem on a continuous interval $[a, b]$ into the problem of getting the value of $N \cdot \sin^2(\theta/2)$ with quantum computing. The phase θ that contained in the amplitude of a superposition state is obtained by phase estimation, as shown in Fig. 1b. To achieve integration calculation with quantum method, we need to address the following challenges.

The first problem is to encode and calculate integrable functions with quantum gate circuits. Supposing the function $f(x)$ is integrable in the interval $[a, b]$, it can be written as $\int_a^b f(x) = F(b) - F(a) = I$ and I is a known constant. However, when functions are complex or cannot be formulated, they are usually approximated by the linear combination of function values under some discrete points in the interval.

$$\int_a^b f(x) dx \approx \sum_{k=0}^n A_k f(x_k) \tag{4}$$

The integration function can be quantized by quantum coding under discrete variables after selecting appropriate approximation methods.

The second problem is to get the distribution of points in the area S_1 . The point x that meets $f(x) \leq y$ is required, and the number is recorded as M . By constructing quantum gate circuits of computation and comparison, the distribution of these points in S_1 is acquired and stored in the amplitude of a superposition state.

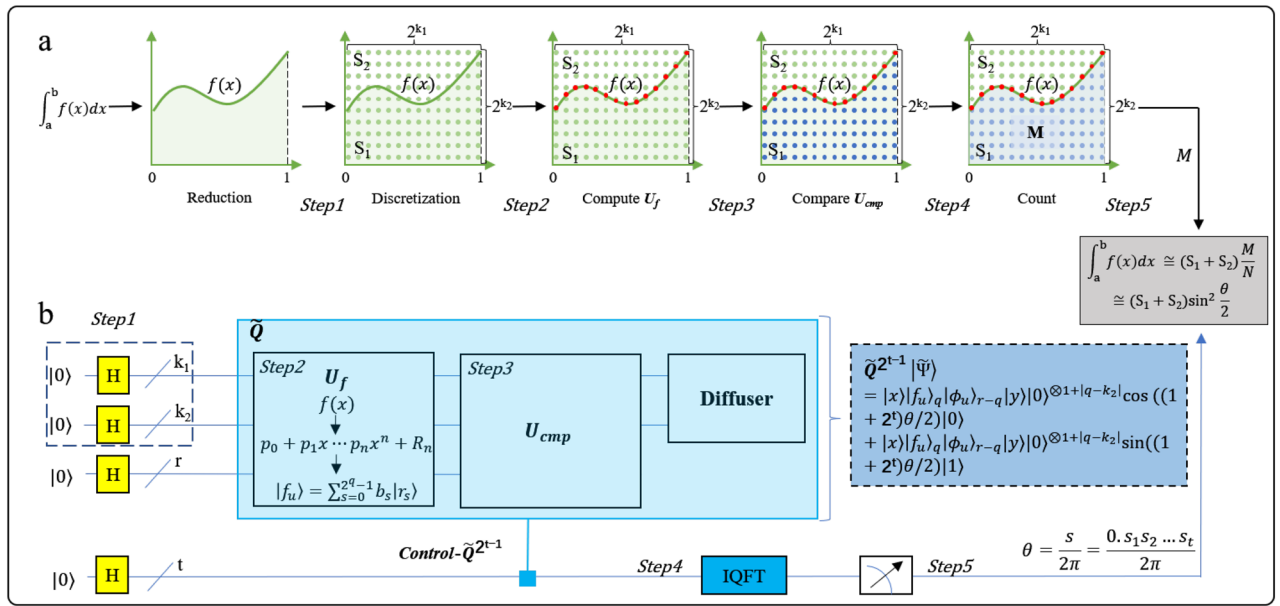


Figure 1. Diagram of quantum integration algorithm GQIA. (a) The classical monte carlo integration(MCI), where S_1 represents integration area, S_2 stands for non-integration area. M stands for the number of points in the area S_1 , and N is the total points in $S_1 + S_2$. (b) Quantum circuit diagram of GQIA, Step1–Step5 corresponds to quantization of the MCI.

The third problem is to get the ratio λ of distribution. There is no acceleration advantage to measure result from quantum circuit directly. Thus, by phase estimation circuit, we can acquire the phase θ with quadratic quantum advantage and the approximate value of the integration is $S \cdot \sin^2(\theta/2)$.

The components of GQIA

Quantization of integration functions

Quantization of integration functions refers to the quantum representation of classical integration functions and the construction of the quantum circuits. The integrations over any interval $[a, b]$ can be transformed into interval $[0, 1]$, and the real number points (x, y) in the area S can be discretized by using k_1 and k_2 qubits, respectively. This means using bits of length k_1 to represent numbers between 0 and 1, similarly, to using bits of length k_2 to represent the value of the vertical axis y and these points evenly divide the integration interval S .

$$x, y) = \left\{ (x_i, y_{ij}) \mid i = 1, 2, \dots, 2^{k_1}; j = 1, 2, \dots, 2^{k_2} \right\} \tag{5}$$

Thus, the number of discrete points is $N = 2^{k_1+k_2}$ (Fig. 1a).

The researches on quantum integration algorithms are rare, and one of the obstacles is to represent integrable functions with quantum coding. In numerical analysis, continuous and bounded functions can be polynomials approximated, such as Chebyshev approximation, best square approximation⁸, etc. If the function $f(x)$ has n -order derivatives on interval $[a, b]$ including x_0 , function $f(x)$ can be approximated with a Taylor expansion at point x_0 , which the coefficients are the n -order derivatives of the function at x_0 . Otherwise, the polynomial coefficients can be determined by taking the function values from multiple points and using the method of undetermined coefficients, and the function is approximated by a simple polynomial, as shown in Eq. (6)

$$f(x) = a_0 + a_1 * x + a_2 * x^2 + \dots + a_n * x^n + R_n(x) \tag{6}$$

If the first $n + 1$ terms are used to approximate the function, the precision of approximation is $O(1/x^{n+1})$. Polynomial approximation is a linear combination of the power of variables and when the value of k_1 of variable discretization is determined, the power circuit is easy to construct, that is the reason for approximating integration functions with Eq. (6). For example, the quadratic power quantum circuit (Fig. 2) can be constructed with Table 2.

Construction of marking oracle

The area S is divided into S_1 and S_2 by the function curve, and only the points in S_1 are required. Here, a method is proposed to mark these points by constructing two quantum oracles. The first one is U_f computing $f(x)$, which $f(x) = \sum_{i=0}^n a_i x^i$ is the first n terms of the approximate polynomial.

Supposing r denotes the number of auxiliary qubits for this oracle, including q qubits representing the superposition output of $f(x)$ under the initial input $|f_u\rangle$. The oracle needs $k_1 + r$ qubits in that way, where k_1 stands for the number of qubits for variable x , and $|\phi_u\rangle$ is the output of the auxiliary qubits.

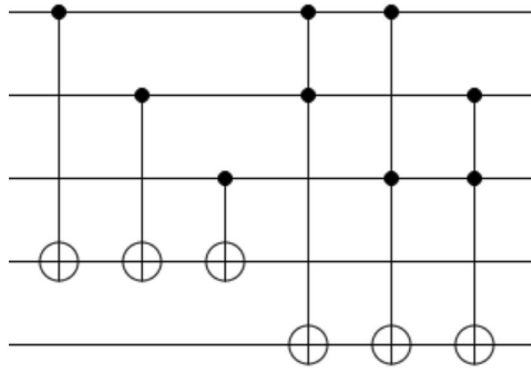


Figure 2. Quantum circuit of quadratic operation.

0	0	0	x_3	x_2	x_1
0	0	0	x_3	x_2	x_1
0	0	0	$x_1 * x_3$	$x_1 * x_2$	x_1
0	0	$x_2 * x_3$	x_2	$x_1 * x_2$	0
0	x_3	$x_2 * x_3$	$x_1 * x_3$	0	0
0	c_4	0	0	0	0
c_5	c_3	c_2	c_1	0	0
c_5	$c_3 + c_4 + x_3$	c_2	$c_1 + x_2$	0	x_1

Table 2. Binary multiplication table.

$$H^{\otimes k_1} |0\rangle^{\otimes k_1} |0\rangle^{\otimes r} \xrightarrow{U_f} \left(\frac{1}{\sqrt{2}}\right)^{k_1} \sum_{u=0}^{2^{k_1-1}} |u\rangle |f_u\rangle_q |\phi_u\rangle_{r-q} \tag{7}$$

The other oracle is to compare $f(x_i)$ with y_{ij} for each x_i . The point (x_i, y_{ij}) that meets $f(x_i) \leq y_{ij}$ needs to be marked, and a comparison circuit including n CGC gates and n ICGC gates can make it³², which CGC and ICGC gates are quantum circuits composed of CNOT and Toffoli gates, and the CGC gate and its inverse ICGC are constructed with two CNOT gates and one Toffoli gate in different orders respectively (Fig. 3). Thus, the number of qubits is $2n + 2$.

$$\begin{aligned} &U_{cmp} U_f H^{\otimes k_1+k_2} |0\rangle^{\otimes k_1+k_2} |0\rangle^{\otimes r+2+|q-k_2|} \\ &= U_{cmp} U_f |x\rangle |y\rangle |0\rangle^{\otimes r+2+|q-k_2|} \\ &= U_{cmp} |x\rangle |f_u\rangle_q |\phi_u\rangle_{r-q} |y\rangle |0\rangle^{\otimes 2+|q-k_2|} \\ &= |x\rangle |f_u\rangle_q |\phi_u\rangle_{r-q} |y\rangle |0\rangle^{\otimes 1+|q-k_2|} \sqrt{1-\lambda} |0\rangle \\ &\quad + |x\rangle |f_u\rangle_q |\phi_u\rangle_{r-q} |y\rangle |0\rangle^{\otimes 1+|q-k_2|} \sqrt{\lambda} |1\rangle \end{aligned} \tag{8}$$

where k_2 is the number of qubits required to represent $y, |y\rangle = \left(\frac{1}{\sqrt{2}}\right)^{k_2} \sum_{v=0}^{2^{k_2-1}} |v\rangle$, similarly, $|x\rangle = \left(\frac{1}{\sqrt{2}}\right)^{k_1} \sum_{u=0}^{2^{k_1-1}} |u\rangle$. If q is not equal to $k_2, 2 + |q - k_2|$ auxiliary qubits $|0\rangle^{\otimes 2+|q-k_2|}$ are needed for this

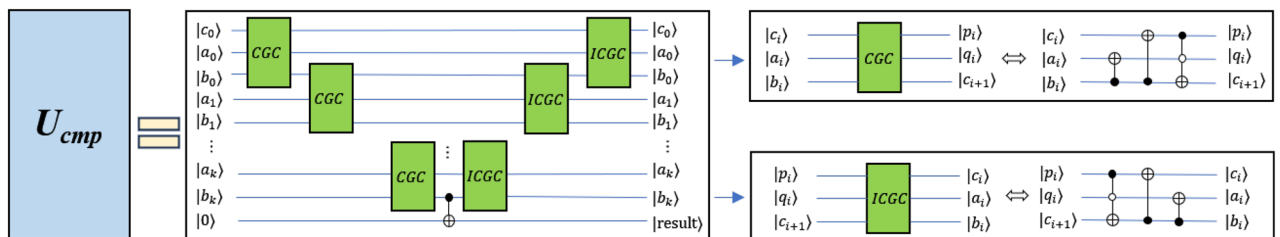


Figure 3. Structure and detailed circuits of comparator U_{cmp} .

comparator. λ is a real number between 0 and 1 evaluated by measurement, which representing the proportion of points that meet the comparison criteria and corresponding to amplitude of the auxiliary bits, when the output is 1. Thus, the comparison results are obtained by evaluating the value of λ , and the necessary qubit number of marking oracle is $k_1 + k_2 + r + 2 + |q - k_2|$.

To sum up, the marking oracle U of GQIA is made up with computation and comparison, which can be recorded as $U = U_{cmp}U_f$. Different from direct measurement, the GQIA obtains the results by phase estimation, which brings quadratic acceleration³³.

Extraction of results

For a quantum state in amplitude amplification (AA) algorithm²⁹,

$$|\Psi\rangle = \mathcal{A}|0\rangle_n|0\rangle = \sqrt{1-\lambda}|\psi_0\rangle_n|0\rangle + \sqrt{\lambda}|\psi_1\rangle_n|1\rangle \tag{9}$$

where $|\psi_1\rangle_n$ is the required target state, $|\psi_0\rangle_n$ is non-target state. Defining $\theta \in [0, \pi]$, so that $\sin^2\theta/2 = \lambda$ and a unitary operator Q ,

$$Q = -\mathcal{A}S_0\mathcal{A}^{-1}S_\chi \tag{10}$$

S_χ adds a negative phase before $|\psi_1\rangle_n|1\rangle$, $|\psi_0\rangle_n|0\rangle$ keeps unchanged.

$$|\Psi\rangle = \mathcal{A}|0\rangle_n|0\rangle = \cos\theta/2|\psi_0\rangle_n|0\rangle + \sin\theta/2|\psi_1\rangle_n|1\rangle \tag{11}$$

Using Q repeated j times for the quantum state $|\Psi\rangle$ gives

$$Q^j|\Psi\rangle = \cos((2j+1)\theta/2)|\psi_0\rangle_n|0\rangle + \sin((2j+1)\theta/2)|\psi_1\rangle_n|1\rangle \tag{12}$$

Similarly, from Eq. (8) we can get a quantum state $|\tilde{\Psi}\rangle$ and a unitary operator \tilde{Q} as

$$\begin{aligned} |\tilde{\Psi}\rangle &= \mathbf{U}H^{\otimes k_1+k_2}|0\rangle^{\otimes k_1+k_2}|0\rangle^{\otimes r+2+|q-k_2|} \\ &= |x\rangle|f_u\rangle_q|\phi_u\rangle_{r-q}|y\rangle|0\rangle^{\otimes 1+|q-k_2|\sqrt{1-\lambda}}|0\rangle \\ &\quad + |x\rangle|f_u\rangle_q|\phi_u\rangle_{r-q}|y\rangle|0\rangle^{\otimes 1+|q-k_2|\sqrt{\lambda}}|1\rangle \\ &= |a\rangle|0\rangle + |b\rangle|1\rangle \end{aligned} \tag{13}$$

$$\tilde{Q} = -\mathbf{U}H\mathbf{S}_0H^{-1}\mathbf{U}^{-1}\mathbf{S}_\chi \tag{14}$$

where

$$|a\rangle = |x\rangle|f_u\rangle_q|\phi_u\rangle_{r-q}|y\rangle|0\rangle^{\otimes 1+|q-k_2|\sqrt{1-\lambda}} = |x\rangle|f_u\rangle_q|\phi_u\rangle_{r-q}|y\rangle|0\rangle^{\otimes 1+|q-k_2|}\cos(\theta/2) \tag{15}$$

$$|b\rangle = |x\rangle|f_u\rangle_q|\phi_u\rangle_{r-q}|y\rangle|0\rangle^{\otimes 1+|q-k_2|\sqrt{\lambda}} = |x\rangle|f_u\rangle_q|\phi_u\rangle_{r-q}|y\rangle|0\rangle^{\otimes 1+|q-k_2|}\sin(\theta/2) \tag{16}$$

In the AA algorithm, the unitary operator \tilde{Q} is equivalent to rotate the superposition states by angle θ , which can be expressed as a 2*2 dimensional matrix in the single bit case

$$\tilde{Q} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{17}$$

It is easy to get the eigenvalues of \tilde{Q} from following formula

$$|\gamma I - \tilde{Q}| = \begin{vmatrix} \gamma - \cos\theta & \sin\theta \\ -\sin\theta & \gamma - \cos\theta \end{vmatrix} = (\gamma - \cos\theta)^2 + \sin^2\theta = 0 \tag{18}$$

The eigenvalues are $\gamma_1 = e^{i\theta}$ and $\gamma_2 = e^{i(2\pi-\theta)}$, either one is feasible because θ has a small value and $2\pi - \theta$ is large, making it easy to observe. In this article, it may be assumed that the value is θ and we take the case of γ_1 . Phase estimation is to get s in $\tilde{Q}^s|\psi\rangle = e^{2\pi is}|\psi\rangle$, where $s = \frac{\theta}{2\pi}$. Supposing that we denote s with t qubits, that is, $s = 0.s_1s_2 \dots s_t$. We can obtain the result of with inverse quantum Fourier transform(Eq. 19).

$$\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi isj} |j\rangle \rightarrow |s\rangle \tag{19}$$

Therefore, s and λ can be obtained by measuring the first t qubits in phase estimation circuit, and the integration can be approximated with $\lambda^2 \cdot N$.

$$\int_a^b f(x)dx \approx (S_1 + S_2) \cdot \lambda^2 = 2^{k_1+k_2} \cdot \sin^2\left(\frac{\pi s}{2^t}\right) \tag{20}$$

Experimental evaluations

To understand GQIA, we propose a simple study case $\int_{-1}^1 e^x dx$.

First, the integration interval $[-1, 1]$ can be transformed to interval $[0, 1]$, and the result is

$$\int_{-1}^1 e^x dx = 2 \int_0^1 e^{2z-1} dz \tag{21}$$

It is known that $e^x = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + o(x^n)$ with Taylor expansion, and e^x can be approximated with the first three terms of Taylor polynomial. For special functions that cannot be Taylor expanded, other polynomial approximation methods could be available as mentioned in Section “Quantization of integration functions”.

$$2 \int_0^1 e^{2z-1} dz \approx \int_0^1 4z^2 + 1 dz \tag{22}$$

Then, the oracle circuits that include calculation and comparison circuits are constructed to mark the points (Fig. 4b). We take the number of qubits for variable z and y , $k_1 = 3$ and $k_2 = 3$ respectively, thus the number of qubits for function $4z^2 + 1$ is $q = 7$ in Eq. (7), and the number of auxiliary qubit is $r = 7$ in Eq. (8).

Finally, the phase $s = \frac{\theta}{2\pi}$ contains the result M (Fig. 4a) is obtained by phase estimation and is estimated to $t = 5$ qubits accuracy in Eq. (19). Hence, the total number of qubits is $k_1 + k_2 + r + t + 2 + |k_2 - q| = 24$, and the experiments were implemented with IBM qiskit 32-bit simulator qasm—simulator.

The final result is $\theta = 4.52$, the number of the points in S is $M = N \cdot \sin^2 \frac{\theta}{2} = 2^6 \cdot \sin^2 \frac{\theta}{2} = 25.76$, and the approximated value of integration is $6.125 \cdot \frac{25.76}{2^6} = 2.465$. In contrast, $\int_{-1}^1 e^x dx = 2.333$, thus the obtained precision is 0.943. According to the precision analysis in Section “Proposed methods”, $\epsilon \leq 6.283$, the experimental precision satisfies the lower bounds of precision $\frac{\sqrt{N}-1}{\sqrt{N}(x^{l+1})\epsilon} = 0.557$. However, the precision of direct measurement fluctuates from 0.570 to 0.809 in the same situation, which demonstrates GQIA is more reliable and high-accuracy.

Discussion

The whole process of GQIA is divided into three steps. The first step achieves quantization of any integrable functions by polynomial approximation and quantum encoding, the second step constructs the oracle of marking and the third step gets the results by the phase estimation.

The first step approximates integration function with the first $l + 1$ terms of a simple polynomial. The number of qubits to compute the value of function $f(x)$ is $k_1 + r$, where k_1 denotes the number of qubits for variable x ,

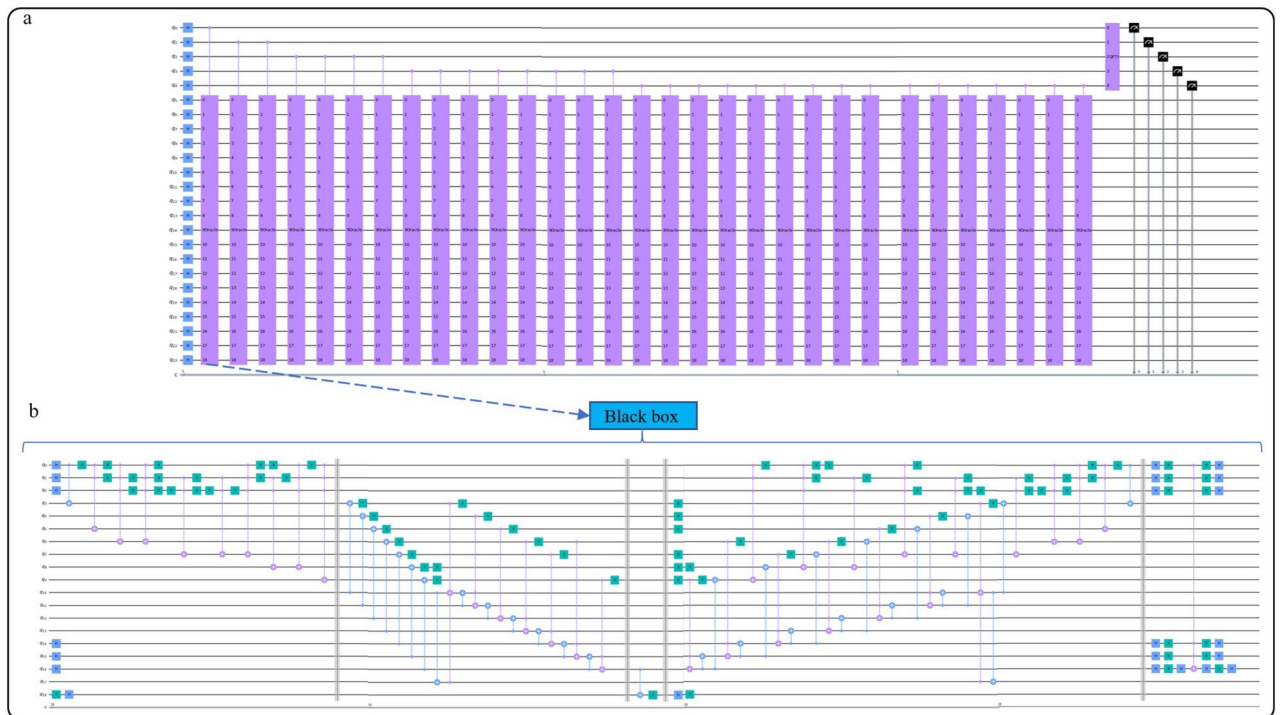


Figure 4. Diagram of $\int_{-1}^1 e^x dx$ with GQIA. (a) The circuit of GQIA for the study case. (b) The black box (oracle) for the study case of GQIA.

and r is the number of auxiliary qubits including q qubits to represent result $|f_u\rangle$, where $q < r$. The computational precision is $\mathcal{O}(1/x^{l+1})$.

In the second step, k_2 is the number of qubits for variable y . When k_2 is not equal to q , additional $|k_2 - q| + 2$ auxiliary qubits are required for the comparator circuit. Thus, the number of qubits in the first two steps is $k_1 + k_2 + r + |k_2 - q| + 2$, and the precision keeps unchanged, $\mathcal{O}(1/x^{l+1})$.

In the third step, t qubits are required to estimate θ to m qubits accuracy, where $t = m + \lceil \log(2 + 1/(2\epsilon)) \rceil$. Thus, the number of required qubits is $k_1 + k_2 + r + |q - k_2| + 2 + t$. For getting the accurate value of θ with high probability $1 - \epsilon$, \tilde{Q} operator need to be implemented $\frac{\pi}{4} \sqrt{2^{k_1+k_2}/M}$ times, and the maximum execution number of \tilde{Q} does not exceed 2^{t-1} times. Hence, $t = \log\left(\frac{\pi}{4} \sqrt{2^{k_1+k_2}/M}\right) + c$, where c is a constant related to error ϵ and the upper bound of error ϵ is $\frac{2\pi}{2^t} \sqrt{M(N-M)} + \frac{\pi^2}{2^{2t}} |N - 2M|$. As a result, the complexity of GQIA is $\mathcal{O}\left(\frac{1}{\epsilon} \sqrt{2^{k_1+k_2}/M}\right)$ that is recorded as $\mathcal{O}\left(\frac{1}{\epsilon} \sqrt{N/M}\right)^{33,34}$, and the precision is $\mathcal{O}\left(\frac{\sqrt{N-1}}{\sqrt{N}(x^{l+1})\epsilon}\right)$.

Conclusion

In conclusion, the GQIA we proposed for solving numerical integration, showing superiority of quantum algorithm in numerical problems. Quantum encoding of any integrable functions presents strong generality by approximating the functions with polynomials. Furthermore, constructing oracle and converting the results to phase exhibit the advantages of quadratic acceleration. The GQIA provides a quantum integration algorithm framework based on the MCI idea, where methods such as polynomial approximation, spatial discretization, and oracle construction are not unique and can be further optimized and studied.

The future work worth exploring about the algorithm of this article including the following points. The GQIA's circuit is relatively deep that poses challenges in running on current quantum computers, the depth of the circuit is mainly caused by phase estimation, and improvements or alternative algorithms can be studied. In addition, the polynomial approximation methods of the integration function may affect the complexity of GQIA's oracle circuits. For example, GQIA mainly applies the truth table method to construct circuits, and further research can be conducted on polynomial circuit construction methods.

Data availability

All relevant data supporting the main conclusions and figures of the document are available on request. Please refer to Guoqiang Shu at sstronger21@163.com.

Code availability

The code used for obtaining the presented numerical results as well as for generating the plots is available from the corresponding author on reasonable request.

Received: 2 November 2023; Accepted: 30 April 2024

Published online: 07 May 2024

References

- An, D. *et al.* Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance. *Quantum* **5**, 481. <https://doi.org/10.22331/q-2021-06-24-481> (2021).
- Griffin, P. & Sampat, R. Quantum computing for supply chain nance. In *2021 IEEE International Conference on Services Computing (SCC)* 456–459 (IEEE, 2021).
- Miyamoto, K. Quantum algorithms for monte carlo integration using pseudo-random numbers. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* 454–455 (IEEE, 2021).
- Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **41**(2), 303332 (1999).
- Grover, L. K. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symposium on Theory of Computing* (1996).
- Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**(15), 150502 (2009).
- Dehghan, M., Masjed-Jamei, M. & Eslahchi, M. On numerical improvement of closed newton cotes quadrature rules. *Appl. Math. Comput.* **165**(2), 251260 (2005).
- Linz, P. *Theoretical Numerical Analysis* (Courier Dover Publications, 2019).
- Burden, R. L., Faires, J. D. & Burden, A. M. *Numerical Analysis* (Cengage Learning, 2015).
- Leobacher, G. & Pillichshammer, F. *Introduction to Quasi-Monte Carlo Integration and Applications* (Springer, 2014).
- Tokdar, S. T. & Kass, R. E. Importance sampling: A review. *Wiley Interdiscip. Rev. Comput. Stat.* **2**(1), 5460 (2010).
- Sharma, G. Pros and cons of different sampling techniques. *Int. J. Appl. Res.* **3**(7), 749752 (2017).
- Kanwal, R. & Liu, K. A Taylor expansion integral equations. *Int. J. Math. Educ. Sci. Technol.* **20**(3), 411414 (1989).
- Yalçınbaş, S. Taylor polynomial solutions of nonlinear Volterra-Fredholm integral equations. *Appl. Math. Comput.* **127**(2–3), 195206 (2002).
- Acioli, P. H. Review of quantum Monte Carlo methods and their applications. *J. Mol. Struct. THEOCHEM* **394**(2–3), 7585 (1997).
- Foulkes, W., Mitas, L., Needs, R. & Rajagopal, G. Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.* **73**(1), 33 (2001).
- Herbert, S. Quantum monte-carlo integration: The full advantage in minimal circuit depth. Preprint at <https://arxiv.org/abs/2105.09100> (2021).
- Abrams, D. S. & Williams, C. P. Fast quantum algorithms for numerical integrals and stochastic processes. arXiv <https://arxiv.org/abs/quant-ph/9908083> (1999).
- Johnston, E. R. Quantum supersampling. ACM SIGGRAPH 2016 Talks (2016).
- Shimada, N. H. & Hachisuka, T. Quantum coin method for numerical integration. *Comput. Graph. Forum* **39**(2436), 257 (2020).
- Heinrich, S. Quantum integration in Sobolev classes. *J. Complex.* **19**(1), 1942 (2003).

22. DeWitt-Morette, C., Maheshwari, A. & Nelson, B. Path integration in non-relativistic quantum mechanics. *Phys. Rep.* **50**(5), 255372 (1979).
23. Heinrich, S. Quantum summation with an application to integration. *J. Complex.* **18**(1), 150 (2002).
24. Rebentrost, P., Gupt, B. & Bromley, T. R. Photonic quantum algorithm for monte carlo integration. Preprint at <https://arxiv.org/abs/1809.02579> (2018).
25. Suzuki, Y. *et al.* Amplitude estimation without phase estimation. *Quantum Inf. Process.* **19**(2), 117 (2020).
26. Aaronson, S. & Rall, P. Quantum approximate counting, simplified. In *Symposium on Simplicity in Algorithms (SOSA)* (SIAM, 2020).
27. Prakash, A. & Kerenidis, I. Method for amplitude estimation with noisy intermediate-scale quantum computers. US Patent App. 16/892,229 (2021).
28. Kaneko, K., Miyamoto, K., Takeda, N. & Yoshino, K. Quantum speedup of monte carlo integration with respect to the number of dimensions and its application to finance. *Quantum Inf. Process.* **20**(5), 124 (2021).
29. Brassard, G., Hoyer, P., Mosca, M. & Tapp, A. Quantum amplitude amplification and estimation. *Contemp. Math.* **305**, 5374 (2002).
30. Grinko, D., Gacon, J., Zoufal, C. & Woerner, S. Iterative quantum amplitude estimation. *npj Quantum Inf.* **7**(1), 16 (2021).
31. Novak, E. Quantum complexity of integration. *J. Complex.* **17**(1), 216 (2001).
32. Xia, H., Li, H., Zhang, H., Liang, Y. & Xin, J. An efficient design of reversible multi-bit quantum comparator via only a single ancillary bit. *Int. J. Theor. Phys.* **57**(12), 37273744 (2018).
33. Brassard, G., Hyer, P. & Tapp, A. Quantum counting. In *International Colloquium on Automata, Languages, and Programming* (Springer, 1998).
34. Diao, Z., Huang, C. & Wang, K. Quantum counting: Algorithm and error distribution. *Acta Appl. Math.* **118**(1), 147159 (2012).

Acknowledgements

This work was supported by Major Science and Technology Projects in Henan Province, China, Grant No.: 221100210600. We thank, Hui Hui Sun, and Cong Cong Feng for insightful discussions as well as Bo Zhao for technical support. We also acknowledge useful discussions and the feedback on the manuscript from Shi Qin Di.

Author contributions

All authors developed the idea for the experiment; Z.S. performed the measurements and analyzed the data; Z.S. and G.Q.S. implemented the control methods. G.Q.S. and S.Y.W. carried out the numerical simulations. G.Q.S., S.Y.W., J.Z. and J.C.X. wrote the manuscript. All authors contributed to the discussions and interpretations of the results.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to G.S.

Reprints and permissions information is available at www.nature.com/reprints.

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International

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

© The Author(s) 2024