A hybrid algorithm to solve linear systems of equations with limited qubit resources

Fang Gao\textsuperscript{1} · Guojian Wu\textsuperscript{1} · Mingyu Yang\textsuperscript{1} · Wei Cui\textsuperscript{2} · Feng Shuang\textsuperscript{1}\textsuperscript{a}

Received: 9 July 2021 / Accepted: 23 November 2021 / Published online: 3 March 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

The solution of linear systems of equations is a very frequent operation and thus important in many fields. The complexity using classical methods increases linearly with the size of equations. The HHL algorithm proposed by Harrow et al. achieves exponential acceleration compared with the best classical algorithm. However, it has a relatively high demand for qubit resources, and the solution $|x\rangle$ is in a normalized form. Assuming that the eigenvalues of the coefficient matrix of the linear systems of equations can be represented perfectly by finite binary number strings, the hybrid iterative phase estimation algorithm (HIPEA) is designed based on the iterative phase estimation algorithm in this paper. The complexity is transferred to the measurement operation in an iterative way, and thus, the demand of qubit resources is reduced in our hybrid algorithm. Moreover, the solution is stored in a classical register instead of a quantum register, so the exact unnormalized solution can be obtained. The required qubit resources in the three variants of HIPEA are different. The first variant only needs one single ancillary qubit. The number of ancillary qubits in the second variant

\textsuperscript{a} This work was supported by the National Natural Science Foundation of China under Grants 61773359, 61720106009 and 61873317.

\textsuperscript{1} Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, Guangxi University, Nanning 530004, China

\textsuperscript{2} School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China
is equal to the number of nondegenerate eigenvalues of the coefficient matrix of linear systems of equations. The third variant is designed with a flexible number of ancillary qubits. The HIPEA algorithm proposed in this paper broadens the application range of quantum computation in solving linear systems of equations by avoiding the problem that quantum programs may not be used to solve linear systems of equations due to the lack of qubit resources.

**Keywords** Linear systems of equations · Hybrid algorithm · Iterative phase estimation · Limited qubit resources

1 Introduction

The solution of linear systems of equations \( Ax = b \) is one basic problem in linear algebra, and the explosive growth of the problem scale drives us to develop faster and more efficient algorithms [1]. The quantum HHL algorithm proposed by Harrow, Hassidim and Lloyd [2] can achieve exponential speedup compared to the traditional conjugate gradient algorithm [3]. However, the number of qubits required in the HHL algorithm increases with the number of the linear systems of equations and the magnitude of eigenvalues of the coefficient matrix \( A \) [3–7]. In the Noisy Intermediate-Scale Quantum Era (NISQ Era), qubit resources are limited [8] (i.e., the number of qubits is between 50 and 100, and the quantum gates are not perfect). Therefore, the development of algorithms to solve linear systems of equations with lower requirements for qubit resources is of practical significance and application value.

In the exploration of reducing the quantum circuit depth, Lee et al. proposed a hybrid HHL algorithm [9] mainly composed of phase estimation [10–15], classical computing and reduced HHL algorithm, and its core is to process the information extracted from the phase estimation in a classical computer to design the quantum circuit of reduced HHL algorithm. When the coefficient matrix \( A \) meets certain conditions, the depth of quantum circuit can be reduced, which is favored due to imperfect quantum gates in the NISQ Era. In this work, a hybrid framework combining quantum and classical approaches is proposed to further reduce the required number of qubits when solving linear systems of equations.

Phase estimation is an important module in the HHL algorithm. Dobšíček et al. proposed an iterative phase estimation algorithm (IPEA), which improves the accuracy of phase estimation by increasing the number of iterations instead of that of ancillary qubits [16–19]. In this work, the hybrid iterative phase estimation algorithm (HIPEA) based on IPEA is presented to solve linear systems of equations, and the number of ancillary qubits can be chosen flexibly according to problem size and available qubit resources.

The standard quantum HHL circuit is adopted in this paper for discussion, and indeed the quantum circuit can also be designed in terms of linear combination of unitaries (LCU) [20, 21], which has become one major technique for quantum algorithm design [22] and is being employed extensively in many areas such as quantum eigensolver [23, 24] and optimization [25]. It is expected that our hybrid algorithm should also be valid in terms of LCU.
The rest of the paper is organized as follows: Sect. 2 introduces the theoretical basis of HIPEA; Then three variants of HIPEA are described detailly in Sect. 3 followed by the corresponding examples and error analysis in Sect. 4; The final conclusions are given in Sect. 5.

2 Theoretical basis

For linear systems of equations \( Ax = b \), it is assumed in this paper that the eigenvalues \( \phi_j \) of the matrix \( A \) can all be perfectly represented by finite binary number strings (i.e., \( \phi_j = 0.\phi_{j1}\phi_{j2}...\phi_{jm} \)), and \( b \) is a normalized vector that can be prepared as \( |b\rangle \) with QRAM [26, 27] and expressed as linear combinations of \( A \)'s eigenvectors. In the following, the general idea of IPEA and HIPEA will be introduced.

2.1 IPEA

In the following description of IPEA, it is assumed that the matrix \( A \) only has one eigenvalue \( \phi_1 \). The IPEA evaluates the eigenvalue digit by digit at different iteration steps. The quantum circuit of the general \( l \)-th iteration shown in Fig. 1 is employed to estimate the \( k \)-th bit (from high to low unless otherwise specified) of \( \phi_1 \) with \( k = m + 1 - l \).

The evolution of the quantum state in the \( l \)-th iteration is described in the following. The initial quantum state is \( |\psi_0\rangle = |0\rangle|b\rangle \). After the first Hadamard gate on the ancillary qubit of the top register, the state becomes \( |\psi_1\rangle = 2^{-1/2}(|0\rangle + |1\rangle)|b\rangle \). Then the controlled-U gate with \( U_l = U^{2^{m-l}} = e^{2\pi i A \times 2^{m-l}} = e^{2\pi i A \times 2^{m-1}} \) is used to perform the Hamiltonian simulation [28–33], and the resulted state is \( |\psi_2\rangle = 2^{-1/2}(|0\rangle + e^{2\pi i (2^{k-1}\phi_1)}|1\rangle)|b\rangle \). The Rz gate, with the angle \( \omega_l = \omega_{m+1-k} = 0.0\phi_{1(k+1)}\phi_{1(k+2)}...\phi_{1(m-1)}\phi_{1m} \) determined by the previous iterations, is imposed on the ancillary qubit leading to \( |\psi_3\rangle = 2^{-1/2} \times e^{-i\omega_l/2}(|0\rangle + e^{2\pi i (0.\phi_{1k})}|1\rangle)|b\rangle \). After the second Hadamard gate on the ancillary qubit, the state evolves to \( |\psi_4\rangle = 2^{-1/2} \times e^{-i\omega_l/2}[(|0\rangle + e^{2\pi i (0.\phi_{1k})})|0\rangle + (1 - e^{2\pi i (0.\phi_{1k})})|1\rangle]|b\rangle \). Therefore, the probability of measuring the top register qubit as \( |0\rangle \) is \( P_0 = \cos^2[\pi (0.\phi_{1k})] \) and that of measuring as \( |1\rangle \) is \( P_1 = \sin^2[\pi (0.\phi_{1k})] \), which means that the output state of the top
register qubit is definitely $|0\rangle$ when $\phi_{1k} = 0$ and $|1\rangle$ when $\phi_{1k} = 1$. So, the accurate value of $\phi_1 = 0.\phi_{11}\phi_{12}...\phi_{1(m-1)}\phi_{1m}$ can be evaluated by $m$ iteration steps with IPEA.

### 2.2 HIPEA

Given linear systems of equations $Ax = b$, the matrix $A$ can be decomposed in its eigenbasis as $A = \Sigma_j \phi_j u_j u_j^{-1}$ with $\phi_j$ as the eigenvalue and $u_j$ as the corresponding normalized eigenvector (i.e., $u_j$ can also be written as $|u_j\rangle$). The vector $b$ can be written as linear combinations of $u_j$ as $b = \Sigma_j \beta_j u_j$ with $\beta_j$ as the projection coefficients. Then the solution of $Ax = b$ is $x = \Sigma_j \beta_j u_j/\phi_j$. IPEA outputs a certain eigenvalue $\phi_j$ of the input matrix $A$ one bit by one bit, and other information in iterations has to be used to solve $Ax = b$. In HIPEA, $|\beta_j\rangle$ and $|u_{jp}\rangle$ ($u_j = [u_{j1}, u_{j2}, \ldots, u_{jp}, \ldots]^T$) can be obtained by quantum measurements in the iterative process with quantum circuits inspired by IPEA. According to Lemma 1, after $m$ iterations, the probability of obtaining the eigenvalue $\phi_j = 0.\phi_{j1}\phi_{j2}...\phi_{jm}$ in the classic register is $\beta_j^2$. Through the post-selection under partial measurement [34], when $\phi_j$ is obtained in the $m$ iterations, $u_{jp}$ can be obtained as the probability of the corresponding product state of the bottom register (i.e., $u_{j1}^2$ is the probability of $|00\cdots0\rangle$, $u_{j2}^2$ is the probability of $|00\cdots1\rangle$). Finally, the overall sign information of $\beta_j u_{jp}$ can be judged with the help of $b = \Sigma_j \beta_j u_j$, and thus, the solution $x = \Sigma_j \beta_j u_j/\phi_j$ can be obtained.

In the design of our hybrid algorithm, it is assumed that the matrix has $N$ different eigenvalues. Both IPEA and HIPEA require two quantum registers (top and bottom registers) to execute quantum operations and a classic register to store measurement results. It is worth mentioned the number of ancillary qubits in the top register can be different in our developed HIPEA algorithm, which is described in details in Sect. 3.

In the following, two lemmas concerning the information extraction and the accurate solution of $Ax = b$ are introduced together with proofs as the theoretical basis of HIPEA.

**Lemma 1** If the eigenvalue $\phi_{j'}$ differs from any other eigenvalue $\phi_j$, where $j$ is an integer ranging from 1 to $N$ except $j'$, the probability of obtaining the eigenvalue $\phi_{j''} = 0.\phi_{j1}\phi_{j2}...\phi_{jm}$ after $m$ iterations is $\beta_j^2$.

**Proof** The original IPEA algorithm only considers the case of one single eigenvalue. However, the general case is that the matrix $A$ with dimension $N$ has $N$ different eigenvalues $\phi_j$ with the integer $j$ ranging from 1 to $N$, and their corresponding normalized eigenvectors can be written as $|u_j\rangle$. The normalized vector $|b\rangle$ can be decomposed in the eigenspace of $A$ as.

$$|b\rangle = \Sigma_j \beta_j |u_j\rangle$$ (1)

In case of multiple eigenvalues, the system state $|\psi_4\rangle$ before measurement in Fig. 1 is:

$$|\psi_4\rangle = \sum_{j=1}^{N} \frac{e^{-i\omega j}}{2} \left[ \beta_j (1 - e^{2\pi i (0,\phi_{jk})}) |0\rangle + \beta_j (1 + e^{2\pi i (0,\phi_{jk})}) |1\rangle \right] |u_j\rangle$$ (2)

\( \copyright \) Springer
It is assumed in the following that the target eigenvalue to be estimated is $\phi_{j'} = 0.\phi_{j'1}\phi_{j'2} \cdots \phi_{j'(m-1)}\phi_{j'm}$ and its corresponding eigenvector is $|u_{j'}\rangle$, the projection coefficient of $|b\rangle$ on $|u_{j'}\rangle$ is $\beta_{j'}$. Same as in IPEA, the rotation parameter $\omega_l$ in the $l$-th iteration of HIPEA is designed to be $\omega_l = 0.0\phi_{j'(k+1)}\cdots\phi_{j'(m-1)}\phi_{j'm}$.

In the first iteration (i.e., $l = 1$), the probability of measuring the ancillary qubit in the top register is

$$
P_{j',1} = \begin{cases} 
\sum_{j=1}^{N} \beta_{j'}^2 \cos^2[\pi (0.\phi_{jm})] , & \text{measured as } |0\rangle \\
\sum_{j=1}^{N} \beta_{j'}^2 \sin^2[\pi (0.\phi_{jm})] , & \text{measured as } |1\rangle 
\end{cases}
$$

Here in the subscript, $j'$ means that $\phi_{j'}$ is the target eigenvalue, and 1 denotes the first iteration.

In the second iteration, the probability of measuring the ancillary qubit is

$$
P_{j',2} = \begin{cases} 
\sum_{j=1}^{N} \beta_{j'}^2 \cos^2[\pi (0.\phi_{jm-1})\phi_{jm} - 0.0\phi_{j'm}] , & \text{measured as } |0\rangle \\
\sum_{j=1}^{N} \beta_{j'}^2 \sin^2[\pi (0.\phi_{jm-1})\phi_{jm} - 0.0\phi_{j'm}] , & \text{measured as } |1\rangle 
\end{cases}
$$

Generally, the probability of measuring the ancillary qubit in the $l$-th iteration is

$$
P_{j',l} = \begin{cases} 
\sum_{j=1}^{N} \beta_{j'}^2 \cos^2[\pi (0.\phi_{jm-l+1})\phi_{jm-l+2} \cdots \phi_{jm} - 0.0\phi_{j'm-l+1})\phi_{j'm-l+2} \cdots \phi_{j'm}] , & \text{measured as } |0\rangle \\
\sum_{j=1}^{N} \beta_{j'}^2 \sin^2[\pi (0.\phi_{jm-l+1})\phi_{jm-l+2} \cdots \phi_{jm} - 0.0\phi_{j'm-l+1})\phi_{j'm-l+2} \cdots \phi_{j'm}] , & \text{measured as } |1\rangle 
\end{cases}
$$

The projections of $P_{j', l}$ on $|u_{j'}\rangle$ is

$$
(P_{j', l})_{u_{j'}} = \begin{cases} 
\beta_{j'}^2 \cos^2[\pi (0.\phi_{jm-l+1})\phi_{jm-l+2} \cdots \phi_{jm} - 0.0\phi_{j'm-l+1})\phi_{j'm-l+2} \cdots \phi_{j'm}] , & \text{measured as } |0\rangle \\
\beta_{j'}^2 \sin^2[\pi (0.\phi_{jm-l+1})\phi_{jm-l+2} \cdots \phi_{jm} - 0.0\phi_{j'm-l+1})\phi_{j'm-l+2} \cdots \phi_{j'm}] , & \text{measured as } |1\rangle 
\end{cases}
$$

Then the total projection probability on $|u_{j'}\rangle$ of obtaining $\phi_{j'}$’s lowest $l$ bits in $l$ iterations in the classic register is multiplication of the corresponding probabilities in Eq. (6) in different iterative steps:

$$
(P_{j', l})^T_{u_{j'}} = \beta_{j'}^2 \prod_{l=1}^{m} \frac{(P_{j', l})_{u_{j'}}}{\beta_{j'}^2}
$$
where “$T$” means total. The probability of obtaining $\phi_{j(m-l+1)}\phi_{j(m-l+2)}\ldots\phi_{j(m-1)}\phi_{jm}$ in $l$ iterations in the classic register is

$$
(P_{j',l})^T = \sum_{j=1}^{N} (P_{j',l})_{u_j}^T
$$

Without losing generality, it is assumed that the matrix $A$ has eigenvalues $\phi_{j'}$ and $\phi_j$ with the low $m-d$ bits of $\phi_{j'}$ the same as those of $\phi_j$ and $\phi_{j'd} \neq \phi_{jd}$. Then in the $(m-d)$-th iteration,

$$
(P_{j',m-d})_{u_{j'}} = \beta_{j'}^2 \tag{9}
$$

$$
(P_{j',m-d})_{u_j}^T = \beta_j^2 \tag{10}
$$

In the $(m-d+1)$-th iteration, due to $(P_{j',m-d+1})_{u_{j'}} = \beta_{j'}^2$ and $(P_{j',m-d+1})_{u_j} = 0$, we have

$$
(P_{j',m-d+1})_{u_{j'}}^T = (P_{j',m-d})_{u_{j'}}^T \times (P_{j',m-d+1})_{u_{j'}}/\beta_{j'}^2 = \beta_{j'}^2 \tag{11}
$$

$$
(P_{j',m-d+1})_{u_j}^T = (P_{j',m-d})_{u_j}^T \times (P_{j',m-d+1})_{u_j}/\beta_j^2 = 0 \tag{12}
$$

The above relations indicate that when $\phi_j$ is different from $\phi_{j'}$, there must exist an iteration (i.e., $m-d+1$) making the probability

$$
(P_{j',m-d+1})_{u_j} = 0 \tag{13}
$$

It is obvious that the parameter $d$ is determined by bit location where $\phi_{j'}$ and $\phi_j$ differ at. When the eigenvalue $\phi_{j'}$ differs from any other eigenvalue $\phi_j$ for the low $m-d+1$ bits, where $j$ is an integer ranging from 1 to $N$ except $j'$, we can draw the following conclusion and complete the proof according to Eq. (8)

$$
(P_{j',m})^T = (P_{j',m-d+1})^T
$$

$$
= \sum_{j=1}^{j'-1} (P_{j',m-d+1})_{u_j}^T + (P_{j',m-d+1})_{u_{j'}}^T + \sum_{j=j'+1}^{N} (P_{j',m-d+1})_{u_j}^T
$$

$$
= 0 + \beta_{j'}^2 + 0 = \beta_{j'}^2 \tag{14}
$$

**Lemma 2** Different from the HHL algorithm, HIPEA can obtain the accurate solution $x$ instead of its normalized vector form $|x\rangle$.

**Proof** Fig. 2 shows the quantum circuit of the original HHL algorithm to extract $|x\rangle$, and it consists of the following steps: (1) The initial state of the system is prepared to be $|\psi_0\rangle = |0\rangle|0\rangle^{\otimes m}|b\rangle$. (2) After the phase estimation module, the state becomes
\[ |\psi_1\rangle = \sum_{j=1}^{N} \beta_j |\phi_j\rangle |u_j\rangle. \]  
(3) The controlled rotation operation makes the state evolve to 
\[ |\psi_2\rangle = \sum_{j=1}^{N} \beta_j (\sqrt{1 - C^2/\phi_j^2} |0\rangle + C/\phi_j |1\rangle) |\phi_j\rangle |u_j\rangle, \]  
where \( C \) is a normalizing constant chosen to ensure rotations are less than \( 2\pi \) [35]. (4) The inverse phase estimation makes the qubits in the middle register unentangled with those in the other two registers, and the output state is 
\[ |\psi_3\rangle = \sum_{j=1}^{N} \beta_j (\sqrt{1 - C^2/\phi_j^2} |0\rangle + C/\phi_j |1\rangle) |0\rangle^\otimes m |u_j\rangle. \] (5) When the ancillary qubit in the top register is measured as \( |1\rangle \), the qubits of the bottom register will collapse to the state 
\[ |x\rangle = C \sum_{j=1}^{N} (\beta_j/\phi_j) |u_j\rangle. \]  
As seen in the above procedures, a normalization constant \( C \) is introduced in order to normalize \( |x\rangle \), which means that an additional step is required to extract \( C \) to obtain the accurate solution 
\[ x = \sum_{j=1}^{N} \beta_j u_j/\phi_j. \]  
Different from HHL, the extracted information \( (\phi_j, |\beta_j|, |u_{jp}|) \) for solving the linear systems of equations is stored in the classic register without normalization constraint. Theoretically, the exact values of \( \phi_j \) and \( \beta_j u_j \) can be extracted within a certain error range, and thus, the accurate solution 
\[ x = \sum_{j=1}^{N} \beta_j u_j/\phi_j \] can be directly obtained. Therefore, the proof is completed.

### 3 Realization of HIPEA

HIPEA is a hybrid framework combining quantum and classical approaches, and the quantum part is performed on quantum circuits, whose design are constrained by available qubit resources (i.e., the number of qubits). In this section, three different variants of HIPEA will be given in details, and they can be chosen flexibly according to available qubit resources.

Without losing generality, the \( N \) different eigenvalues \( \phi_j = 0, \phi_j, \phi_j, \ldots, \phi_{jm} \) of the coefficient matrix \( A \) in \( Ax = b \) can be described by a binary tree shown in Fig. 3a, where \( \phi_j(m+1-l) \) represents the estimated \( (m+1-l) \)-th bit of \( \phi_j \) in the \( l \)-th iteration, \( \phi_{[1,2,\ldots,N]} \) represents the \( N \) eigenvalues (i.e., \( \phi_1, \phi_2, \ldots, \phi_N \)), and \( \phi_{[1,2,\ldots,N]} k \) represents their \( k \)-th bit.
If the low \( m - d_1 \) bits for all eigenvalues \( \phi_j \) are the same (i.e., \( \phi_j(d_1+1) \phi_j(d_1+2) \cdots \phi_jm \) are the same for all \( j \)) and the first divergence appears at the \( d_1 \)-th bit of \( \phi_j \), then in the \((m + 1 - d_1)\)-th iteration, the eigenvalues can be divided into two groups with the \( n_{d_1} \)-th eigenvalue \( \phi_j \) as the boundary

\[
\phi_{[1,2,..,n_{d_1}]} d_1 \neq \phi_{[n_{d_1}+1,n_{d_1}+2,..,N]} d_1
\]  

(15)

Similarly, after the \((t-1)\)-th divergence appears in the \((m + 1 - d_{t-1})\)-th iteration, the eigenvalues can be divided into \( t \) groups according to their low \( m + 1 - d_{t-1} \) bits. Until the \((N - 1)\)-th divergence appeared in the \((m + 1 - d_{N-1})\)-th iteration, all the \( N \) different eigenvalues are differentiated.

One specific example is given in Fig. 3b, where there are four different eigenvalues:

\[
\phi_1 = 0.101110000, \quad \phi_2 = 0.000010110 \\
\phi_3 = 0.011011011, \quad \phi_4 = 0.0001111011
\]  

(16)

The first divergence appears in the first iteration resulting to two groups \( \phi_{[1,2]} \) and \( \phi_{[3,4]} \); The second divergence appears in the second iteration, leading to \( \phi_1, \phi_2 \) and \( \phi_{[3,4]} \); The third divergence in the sixth iteration differentiates all the four eigenvalues.

The following three variants of HIPEA will be described based on the concept of "differentiation at different divergences", and they all include four steps:

1. Extract the eigenvalues \( \phi_j \) and the absolute values of the corresponding projection coefficients \( |\beta_j| \). The three variants of HIPEA mainly differ in this step, which will be shown in details from Sects. 3.1 to 3.3.
(2) Extract the absolute value $|u_{jp}|$ of each element of the normalized eigenvector. Post-selection is a very common processing method in the field of quantum computing. When the measuring results of the ancillary qubit(s) in the top register are $\phi_1 \phi_2 \cdots \phi_{j(m-1)} \phi_j m$ in the iterations, $u^2_{jp}$ can be measured as the probabilities of qubits in the bottom register being the Kronecker product basis states. The $|u_{jp}|$ can be calculated as the square root of $u^2_{jp}$.

(3) Extract the values of $\beta_j u_j$. The following equations can be derived from Eq. (1)

\[
(-1)^{n_{11}} \times |\beta_{1} u_{11}| + (-1)^{n_{21}} \times |\beta_{2} u_{21}| + \cdots + (-1)^{n_{N1}} \times |\beta_{N} u_{N1}| = b_1 \\
(-1)^{n_{12}} \times |\beta_{1} u_{12}| + (-1)^{n_{22}} \times |\beta_{2} u_{22}| + \cdots + (-1)^{n_{N2}} \times |\beta_{N} u_{N2}| = b_2 \\
\vdots \\
(-1)^{n_{1N}} \times |\beta_{1} u_{1N}| + (-1)^{n_{2N}} \times |\beta_{2} u_{2N}| + \cdots + (-1)^{n_{NN}} \times |\beta_{N} u_{NN}| = b_N
\]  
(17)

where $n_{jp}$ is an integer of 0 or 1 indicating the positive or negative sign of $\beta_j u_{jp}$. The information of signs can be determined by traversing all possible combinations of $n_{jp}$, and thus, $\beta_j u_j$ can be obtained.

(4) Obtain the solution of linear systems of equations with the above extracted information. With the extracted $\phi_j$ and $\beta_j u_j$, the solution of the linear systems of equations is obtained as $x = \Sigma_j \beta_j u_j / \phi_j$.

### 3.1 HIPEA with single ancillary qubit

Among the three variants, the least number of ancillary qubits is one. The eigenvalue extraction process of HIPEA with single ancillary qubit, which is labeled as the first variant, is almost the same as that of IPEA except the design of rotation parameters. As described in Sect. 2.1, IPEA only considers the case of one single eigenvalue, and it always collapse to $|0\rangle$ or $|1\rangle$ with a probability of 1 when the ancillary qubit in the top register is measured in each iteration. While HIPEA considers multiple eigenvalues, and the state of the ancillary qubit collapses to $|0\rangle$ or $|1\rangle$ with the probability no longer necessarily being 0 or 1 after measurement according to Eq. (5).

Taking the case in Fig. 3a as an example, Fig. 4 shows quantum circuits of HIPEA with the single ancillary qubit in different iterations, where the top register, the unitary matrix $U_l$ and rotation parameters $\omega_l$ are the same as in IPEA described in Sect. 2.1. Unless otherwise stated, the states of qubits in the top and bottom registers are, respectively, initialized to $|0\rangle$ and $|b\rangle$ after each iteration.

The first step of the first variant is performed as follows:

In the first iteration shown in Fig. 4, the rotation parameter of the Rz gate is $\omega_1 = -2\pi(0.0)$. The probability of the ancillary qubit being measured as $|\phi_{j\phi_{jm/\phi_{jm}}}|$ in the top register is $P_{j\phi_{jm/\phi_{jm}}}$. The rotation parameter $\omega_2$ in the second iteration is based on the measurement results of the first iteration. When no divergence appears in the first iteration, we have $P_{j\phi_{jm/\phi_{jm}}} = 1$ according to Eq. (3), and the rotation parameter in the second iteration $\omega_2 = -2\pi(0.0\phi_{jm/\phi_{jm}})$. The first divergence of different eigenvalues appears in the $d_1$-th bit, which means that after $l$ iterations, the measurement results
satisfy

\[
(P_j, l)^T = 1 \\
\text{s.t. } l < m + 1 - d_1
\]  

In the \((m + 1 - d_1)\)-th iteration, \(\phi_{1,2...,n_{d_1}}d_1 \neq \phi_{n_{d_1}+1,n_{d_1}+2,...,N}d_1\), according to Eq. (5), the measurement probability of being \(|0\rangle\) or \(|1\rangle\) is no longer 0 or 1:

\[
0 < (P_{1,...,n_{d_1},m+1-d_1})^T \times (P_{n_{d_1}+1,...,N},m+1-d_1)^T < 1
\]

After the first divergence appears in the \((m + 1 - d_1)\)-th iteration, two experiments are performed in the \((m + 2 - d_1)\)-th iteration: In experiment 1, the rotation parameters in the \((m + 2 - d_1)\)-th iteration are designed as \(\omega_{m+2-d_1} = -2\pi (0.0\phi_{1,...,n_{d_1}}d_1 \phi_{1,...,n_{d_1}}(d_{1}+1) \cdot \cdot \cdot \phi_{1,...,n_{d_1}}m)\). The probability of obtaining \(\phi_{1,...,n_{d_1}}d_1 \phi_{1,...,n_{d_1}}(d_{1}+1) \cdot \cdot \cdot \phi_{1,...,n_{d_1}}m\) as the measurement results after this iteration is \((P_{1,...,n_{d_1},1})^T\); In experiment 2, the rotation parameters in the \((m + 2 - d_1)\)-th iteration are designed as \(\omega_{m+2-d_1} = -2\pi (0.0\phi_{n_{d_1}+1,...,N}d_1 \phi_{n_{d_1}+1,...,N}(d_{1}+1) \cdot \cdot \cdot \phi_{n_{d_1}+1,...,N}m)\). The probability of obtaining \(\phi_{n_{d_1}+1,...,N}d_1 \phi_{n_{d_1}+1,...,N}(d_{1}+1) \cdot \cdot \cdot \phi_{n_{d_1}+1,...,N}m\) as the measurement results after this iteration is \((P_{n_{d_1}+1,...,N},1)^T\).

Generally, after \((t - 1)\)-th divergence appears in the \((m + 1 - d_{t-1})\)-th iteration, \(t\) different experiments are performed in the \((m + 2 - d_{t-1})\)-th iteration. The target eigenvalues in the \((m + 2 - d_{t-1})\)-th iteration of the \(t\) experiments are different, so the corresponding rotation parameters \(\omega_{m+2-d_{t-1}}\) are also different.
After the \((N - 1)\)-th divergence appears in the \((m + 1 - d_{N-1})\)-th iteration, \(N\) different experiments are performed in \((m + 2 - d_{N-1})\)-th iteration, with the experiment \(j\) taking \(\phi_j\) as the target eigenvalue. According to Lemma 1, after the \(m\)-th iteration, the probability of obtaining \(\phi_j\) is \(\beta_j^2\). Namely, all the eigenvalues \(\phi_j\) of \(A\) and the absolute values of the corresponding projection coefficients \(|\beta_j|\) can be extracted after \(m\) iterations.

To conclude, in order to obtain the solution of \(Ax = b\), we have to conduct different experiments with \(m\) iterations to differentiate the eigenvalues of matrix \(A\). The first variant requires the least number of ancillary qubits, that is, only one ancillary qubit is needed in the top register in each iteration. When the \(N\) eigenvalues are all nondegenerate, at most \(N\) different experiments have to be performed to differentiate these eigenvalues.

### 3.2 HIPEA with \(N\) ancillary qubits

HIPEA with \(N\) ancillary qubits, which is labeled as the second variant, achieves the same effect with \(N\) ancillary qubits instead of \(N\) experiments to differentiate \(N\) different eigenvalues. The top quantum register in the second variant contains \(N\) ancillary qubits, and the quantum circuits in different iterations are shown in Fig. 5.

The unitary matrices \(U_l\) and rotation parameters \(\omega_l^q\) are designed in the same way as in Sect. 2.1, where the superscript \(q\) means that the Rz gate with the rotation parameter \(\omega_l^q\) acts on the qubit top[\(q\)] in the top register.

The first step of the second variant is performed as follows:

In the first iteration shown in Fig. 5, only the qubit top[1] is used in the top register with the rotation parameter \(\omega_1^1 = -2\pi (0,0)\). Similar with the first variant, the measurement results are satisfied with Eq. (18) before the first divergence appears. It is

---

![Fig. 5 Quantum circuits of HIPEA with \(N\) ancillary qubits in different iterations](image-url)
worth noted that before the \((m + 2 - d_1)\)-th iteration \((l < m + 2 - d_1)\), no matter which eigenvalue is taken as the target eigenvalue, the corresponding rotation parameters are always the same. Therefore, only one qubit \(top[1]\) in the top register is used before the \((m + 2 - d_1)\)-th iteration.

The first divergence appears in the \((m + 1 - d_1)\)-th iteration, and the measurement results in this iteration obeys Eq. (19). The eigenvalues \(\phi_{[1,...,n_{d1}]1}\) and \(\phi_{[n_{d1}+1,...,N]}\) can be differentiated in the \((m + 2 - d_1)\)-th iteration with the rotation parameter \(\omega_{m+2-d_1}\) designed with \(\phi_{[1,...,n_{d1}]1}\) and \(\phi_{[n_{d1}+1,...,N]}\) as the target eigenvalues, respectively. In the second variant, the same effect can be achieved by increasing the number of qubits used in the top register instead of increasing the number of experiments in the first variant. Namely, in the \((m + 2 - d_1)\)-th iteration in Fig. 5, the rotation parameter \(\omega_{1m+2-d_1}\) of Rz gate in the qubit \(top[1]\) is designed with the target eigenvalue \(\phi_{[1,...,n_{d1}]1}\), and the rotation parameter \(\omega_{2m+2-d_1}\) of Rz gate in the qubit \(top[2]\) is designed with the target eigenvalue \(\phi_{[n_{d1}+1,...,N]}\):

\[
\omega_{1m+2-d_1} = -2\pi (0.0\phi_{[1,...,n_{d1}]1}\phi_{[1,...,n_{d1}]}(d_1+1) \cdots \phi_{[1,...,n_{d1}]m})
\]

\[
\omega_{2m+2-d_1} = -2\pi (0.0\phi_{[n_{d1}+1,...,N]}\phi_{[n_{d1}+1,...,N]}(d_1+1) \cdots \phi_{[n_{d1}+1,...,N]m})
\]

Generally, after the \((t - 1)\)-th divergence appears in the \((m + 1 - d_{t-1})\)-th iteration, the eigenvalues can be divided into \(t\) groups according to their low \(m + 1 - d_{t-1}\) bits. Accordingly, starting from the \((m + 2 - d_{t-1})\)-th iteration, the rotation parameters \(\omega_{m+2-d_{t-1}}\) have \(t\) different values, and \(t\) qubits in total in the top register are put into use:

\[
\omega_{1m+2-d_{t-1}} \neq \omega_{2m+2-d_{t-1}} \neq \cdots \neq \omega_{lm+2-d_{t-1}}
\]

where \(\omega_{lm+2-d_{t-1}}\) denotes the rotation parameter of the Rz gate on the qubit \(top[q]\).

After the \((N - 1)\)-th divergence appears in the \((m + 1 - d_{N-1})\)-th iteration, all the \(N\) eigenvalues can be differentiated. Accordingly, starting from the \((m + 2 - d_{N-1})\)-th iteration, all the \(N\) qubits in the top register are put into use, and the rotation parameter of Rz gate on the qubit \(top[q]\) are designed with \(\phi_q\) as the target eigenvalue. According to Lemma 1, all the eigenvalues \(\phi_j\) and the absolute value of the corresponding projection coefficients \(|\beta_j|\) can be extracted after \(m\) iterations.

In the second variant, the number of iterations is the same as in the first variant, but only one experiment has to be conducted instead of \(N\) experiments. The complexity just moves from “the number of experiments” to “the number of ancillary qubits”, and the number of ancillary qubits in the top register is \(N\) when all the eigenvalues of matrix \(A\) are not degenerate.

### 3.3 HIPEA with flexible multiple ancillary qubits

HIPEA with flexible multiple ancillary qubits, which is labeled as the third variant, tries to differentiate different eigenvalues with flexible multiple ancillary qubits using a traversing strategy. The quantum circuits in different iterations are shown in Fig. 6.
The number of ancillary qubits in the top register is $n_{\text{top}}$, and the $q$-th qubit is marked as $\text{top}[q]$. The meaning of the unitary matrices $U_l$ and rotation parameters $\omega_l$ are the same as in Sect. 2.1.

A set of experiments is performed in each iteration in the third variant to traverse all possible combinations of rotation parameters $\omega_l$, whereas $\omega_l$ in the first and second variants in each iteration are designed based on the measuring results of previous iterations. The operation on the qubit $\text{top}[q]$ in the $l$-th iteration in the third variant can be mapped to the operation on the ancillary qubit in the $((l-1) \times n_{\text{top}} + q)$-th iteration in the first variant, which means that the measuring results of the $l$ iterations in the third variant can be mapped to those of the $l \times n_{\text{top}}$ iterations in the first variant, namely $(P_{j'}^{T}, l \times n_{\text{top}})$.

Since "traversing" the rotation parameters is equivalent to "traversing" the target eigenvalues, we have $(P_{j'}^{T}, l \times n_{\text{top}})^T = 0$ in the experiments where the target phase $\phi_{j'}$ is not the eigenvalue of matrix $A$. Non-zero $(P_{j'}^{T}, l \times n_{\text{top}})^T$ in the third variant gives the measuring probability of the low $l \times n_{\text{top}}$ bits of the eigenvalue $\phi_{j'}$ of matrix $A$ after $l$ iterations. The number of non-zero $(P_{j'}^{T}, l \times n_{\text{top}})^T$ after $l$ iterations $n_{\text{iterl}}^{\text{nonzero}}$ can be understood as the number of divergences shown in Fig. 3a for the low $l \times n_{\text{top}}$ bits after $l$ iterations.

The rotation parameter of the Rz gate acting on the qubit $\text{top}[q]$ in the $l$-th iteration is denoted as $\omega_{((l-1) \times n_{\text{top}} + q)} = -2\pi (0,\text{bin}_{q}^{\text{iterl}})$, where $\text{bin}_{q}^{\text{iterl}}$ is a $(q + (l-1) \times n_{\text{top}})$-bit binary number and designed as follows: (1) Similar as the design principle of $\omega_l$ in Sec. 2.1, the highest bit of $\text{bin}_{q}^{\text{iterl}}$ is always 0; (2) All possible combinations of values of the next $q - 1$ bits (from high to low) of $\text{bin}_{q}^{\text{iterl}}$ are traversed; (3) The low
Table 1 \( \omega(l-1) \times n_{top} \times q \) for each experiment in the first iteration of the third variant

| Iteration 1 | Qubit \( |top[q]\) | Rotation parameter \( \omega_{l-1} \times n_{top} \times q \) | Experiment number |
|-------------|----------------|-------------------------------------------------|------------------|
| 1 \( l=1 \) | \( top[1] \) | \( \omega_1 \) | \( -2\pi(0.0) \) \( -2\pi(0.0) \) |
|             | \( top[2] \) | \( \omega_2 \) | \( -2\pi(0.00) \) \( -2\pi(0.01) \) |
|             | \( top[3] \) | \( \omega_3 \) | \( -2\pi(0.000) \) \( -2\pi(0.011) \) |
|             | \( top[n_{top}-2] \) | \( \omega_{n_{top}-2} \) | \( -2\pi(0.00\ldots0) \) \( -2\pi(0.01\ldots1) \) |
|             | \( top[n_{top}-1] \) | \( \omega_{n_{top}-1} \) | \( -2\pi(0.00\ldots0) \) \( -2\pi(0.01\ldots1) \) |
|             | \( top[n_{top}] \) | \( \omega_{n_{top}} \) | \( -2\pi(0.00\ldots0) \) \( -2\pi(0.01\ldots1) \) |

\((l-1) \times n_{top}\) bits of \( bin_{q}^{iterl} \) take the same values as the low \((l-1) \times n_{top}\) bits of \( \phi_j' \left( \left( P_{j', (l-1) \times n_{top}} \right)^T \neq 0 \right) \) extracted after \( l-1 \) iterations in the third variant.

The first step of the third variant is performed as follows:

The goal of the first iteration is to estimate the low \( n_{top} \) bits of the eigenvalues \( \phi_j \). The rotation parameters of each experiment in iteration 1 are listed in Table 1. Due to the traversal scheme of designing the rotation parameters \( \omega_q \), the number of experiments in this iteration is \( 2^{n_{top}-1} \), and all non-zero measurement results \( \left( P_{j', n_{top}} \right)^T \) can be obtained from these experiments.

Generally, the goal of the \( l \)-th iteration is to extract the information of the low \( l \times n_{top} \) bits of the eigenvalues \( \phi_j \). Taking the case in Fig. 3a as an example, we have \( n_{iterl-1}^{nonzero} = t \) when \( d_{l-1} \leq (l-1) \times n_{top} \leq d_l \), which means that \( t \) divergences in the eigenvalues can be differentiated after \( l-1 \) iterations. The high \( q \) bits of \( bin_{q}^{iterl} \) in the \( l \)-th iteration are designed in the same way as in Table 1. As mentioned above, the low \((l-1) \times n_{top}\) bits of \( bin_{q}^{iterl} \) in the \( l \)-th iteration take the same values as those of \( \phi_j' \left( \left( P_{j', (l-1) \times n_{top}} \right)^T \neq 0 \right) \) extracted after \( l-1 \) iterations in the third variant, and they have \( t \) possible values corresponding to \( t \) divergences. Therefore, there are \( t \times 2^{n_{top}-1} \) values to be traversed for \( bin_{n_{top}}^{iterl} \) in the \( l \)-th iteration, and the number of experiments to be conducted is also \( t \times 2^{n_{top}-1} \) in this iteration. In analogy with the first variant, measuring probabilities \( \left( P_{j', l \times n_{top}} \right)^T \) are obtained after the \( l \)-th iteration in the third variant, and the low \( l \times n_{top} \) bits of the eigenvalues \( \phi_j \) can be extracted accordingly.

According to Lemma 1, after \( \left[ m/n_{top} \right] \) (\( \left[ \right] \) means rounding up to an integer) iterations, all the eigenvalues \( \phi_j \) and the absolute values of the corresponding projection coefficients \( \left| \beta_j \right| \) can be extracted.

In the third variant, the number of ancillary qubits is flexible, and the number of iterations is reduced to \( \left[ m/n_{top} \right] \). As the cost of “flexible number of ancillary qubits”, the number of experiments in each iteration increases in order to traverse all possible combinations of rotation parameters. The divergences may appear in any iteration in the third variant, two extreme cases are considered here to analyze the total required number of experiments:
Case 1 $\text{iter}^{1}_{\text{nonzero}} = \text{iter}^{2}_{\text{nonzero}} = \cdots = \text{iter}^{\lceil m/n_{\text{top}} \rceil} = N$ ($N$ divergences can be differentiated or $N$ non-zero measuring possibilities are obtained in the first iteration): The number of experiments required is the largest, reaching $2^{n_{\text{top}}-1} + N \times (\lceil m/n_{\text{top}} \rceil - 1) \times 2^{n_{\text{top}}-1}$;

Case 2 $\text{iter}^{1}_{\text{nonzero}} = \text{iter}^{2}_{\text{nonzero}} = \cdots = \text{iter}^{\lceil m/n_{\text{top}} \rceil - 1} = 1$, $\text{iter}^{\lceil m/n_{\text{top}} \rceil} = N$ ($N$ divergences can be differentiated or $N$ non-zero measuring possibilities are obtained in the last iteration): The number of experiments required is the smallest, which is $\lceil m/n_{\text{top}} \rceil \times (2^{n_{\text{top}}-1} - 1)$.

In terms of qubit resource consumption, the number of qubits in the top register in the third variant has great flexibility and can be greater or less than $N$.

4 Results

In this section, HHL and the three variants of HIPEA are employed in solving linear $Ax = b$ with $A$ and $b$ being

$$A = 2^{-9} \times \begin{bmatrix} 224.5271 & -0.7218 & -0.0960 & -0.4480 \\ -0.7218 & 127.8316 & -0.8596 & -0.1073 \\ -0.0960 & -0.8596 & 125.9457 & -0.0794 \\ -0.4480 & -0.1073 & -0.0794 & 189.6956 \end{bmatrix}$$  \hspace{1cm} (23)

$$b = \begin{bmatrix} 0.3538 & -0.5054 & 0.0396 & 0.7860 \end{bmatrix}^T$$  \hspace{1cm} (24)

The eigenvalues of the Hermitian matrix $A$ are shown in Eq. (16) and Fig. 3b, and the corresponding eigenvectors $u_1, u_2, u_3$ and $u_4$ are

$$u_1 = \begin{bmatrix} u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \end{bmatrix} = \begin{bmatrix} -0.7444 \\ 0.1296 \\ -0.0496 \\ 0.6531 \end{bmatrix}, \quad u_2 = \begin{bmatrix} u_{21} \\ u_{22} \\ u_{23} \\ u_{24} \end{bmatrix} = \begin{bmatrix} 0.3976 \\ 0.6253 \\ 0.5593 \\ 0.3716 \end{bmatrix},$$

$$u_3 = \begin{bmatrix} u_{31} \\ u_{32} \\ u_{33} \\ u_{34} \end{bmatrix} = \begin{bmatrix} 0.5356 \\ -0.3225 \\ -0.4458 \\ 0.6406 \end{bmatrix}, \quad u_4 = \begin{bmatrix} u_{41} \\ u_{42} \\ u_{43} \\ u_{44} \end{bmatrix} = \begin{bmatrix} -0.0295 \\ -0.6987 \\ 0.6971 \\ 0.1581 \end{bmatrix}$$  \hspace{1cm} (25)

Decomposing the normalized column vector $b$ on the eigenspace of $A$ spanned by these eigenvectors, we can obtain the corresponding projection coefficients as $\beta_1 = 0.1825, \beta_2 = 0.1389, \beta_3 = 0.8384$ and $\beta_4 = 0.4945$. The exact solution $x$ obtained by the classic algorithm is taken as a benchmark in the analysis:

$$x = A^{-1}b = \begin{bmatrix} 4.9584 & -1.1007 & -0.6430 & 6.2099 \end{bmatrix}^T$$  \hspace{1cm} (26)
Table 2 Results of HHL and the three HIPEA variants

| Variant                      | Relative error | Qubits needed |
|------------------------------|----------------|--------------|
| HHL algorithm                | 0.0083         | 12           |
| The first HIPEA variant      | 0.0100         | 3            |
| The second HIPEA variant     | 0.0141         | 6            |
| The third HIPEA variant      | 0.0048         | 5            |

Meanwhile, the exact normalized solution is:

\[
|x\rangle = \frac{x}{\|x\|} = \begin{bmatrix} 0.6161 & -0.1368 & -0.0799 & 0.7716 \end{bmatrix}^T \tag{27}
\]

HHL and the three variants are implemented in PyQPanda, an open-source quantum computing software development framework. For clarity, \( \tilde{\beta}_j \), \( \tilde{u}_j \) and \( \tilde{x} \) are used instead in the following to indicate the results by the three variants of HIPEA.

In the following, the results are shown in details.

4.1 Results of HHL

The specific settings of quantum HHL circuit shown in Fig. 2 for this example are:
(1) Two qubits are allocated to the bottom register to map the 4-dimensional column vector \( b \); (2) Nine qubits are allocated to the middle register since each eigenvalue can be represented by one nine-bit binary number; (3) One ancillary qubit is allocated to the top register. The procedures are described in the proof of Lemma 2.

Theoretically, the exact normalized solution is stored in the bottom register. However, when the solution is extracted by measuring the qubits in the bottom register, an approximate solution with statistical error is obtained:

\[
|\tilde{x}\rangle = C \sum_{j=1}^{4} (\beta_j / \phi_j) |u_j\rangle \approx \begin{bmatrix} 0.6138 & -0.1306 & -0.0843 & 0.7740 \end{bmatrix}^T \tag{28}
\]

Comparing the obtained normalized solution \( |\tilde{x}\rangle \) in Eq. (28) with the exact normalized solution \( |x\rangle \) in Eq. (27), the relative error is

\[
\epsilon = \frac{\| |x\rangle - |\tilde{x}\rangle \|}{\| |x\rangle \|} = 0.0083 \tag{29}
\]
4.2 Results of the first HIPEA variant

According to the quantum circuits in Fig. 4, the specific settings of the first variant in this example are: (1) Two qubits are allocated to the bottom register to map the 4-dimensional column vector $b$; (2) Since each eigenvalue can be represented by one nine-bit binary number, nine iterations are needed to estimate these eigenvalues; (3) As shown in Fig. 3b, there are three divergences in total in the estimation process, so four experiments are required accordingly to differentiate the eigenvalues.

The first variant described in Sect. 3.1 is applied to the example, and results in each step are as follows:

(1) After nine iterations of the four experiments, the four eigenvalues in Eq. (16) are extracted from the measurement results of the top register. The probabilities of obtaining these four eigenvalues are, respectively:

\[
P_{10110000} = (P_{1,9})^T = \tilde{\beta}_1^2 = 0.03440, \quad P_{00010110} = (P_{2,9})^T = \tilde{\beta}_2^2 = 0.01975
\]

\[
P_{01011011} = (P_{3,9})^T = \tilde{\beta}_3^2 = 0.24440, \quad P_{00011101} = (P_{4,9})^T = \tilde{\beta}_4^2 = 0.70386
\]

leading to

\[
|\tilde{\beta}_1| = 0.1828, \quad |\tilde{\beta}_2| = 0.1405, \quad |\tilde{\beta}_3| = 0.4944, \quad |\tilde{\beta}_4| = 0.8390
\]  

(2) The measurement is performed on the bottom register when the measurement results of the top register are, respectively, $\phi_1, \phi_2, \phi_3$ and $\phi_4$, and the results are

\[
|\tilde{u}_1| = \begin{bmatrix} 0.7522 & 0.1352 & 0.0349 & 0.6440 \end{bmatrix}^T, \quad |\tilde{u}_2| = \begin{bmatrix} 0.4079 & 0.6213 & 0.5500 & 0.3809 \end{bmatrix}^T
\]

\[
|\tilde{u}_3| = \begin{bmatrix} 0.0202 & 0.6991 & 0.6975 & 0.1551 \end{bmatrix}^T, \quad |\tilde{u}_4| = \begin{bmatrix} 0.5365 & 0.3240 & 0.4435 & 0.6407 \end{bmatrix}^T
\]

(3) According to Eq. (17), the following values of $\tilde{\beta}_j\tilde{u}_j$ are obtained:

\[
\tilde{\beta}_1\tilde{u}_1 = \begin{bmatrix} -0.1375 & 0.0247 & -0.0064 & 0.1177 \end{bmatrix}^T, \quad \tilde{\beta}_2\tilde{u}_2 = \begin{bmatrix} 0.0573? & 0.0873 & 0.0773 & 0.0535 \end{bmatrix}^T
\]

\[
\tilde{\beta}_3\tilde{u}_3 = \begin{bmatrix} -0.0100 & -0.3456 & 0.3448 & 0.0767 \end{bmatrix}^T, \quad \tilde{\beta}_4\tilde{u}_4 = \begin{bmatrix} 0.4501 & -0.2718 & -0.3721 & 0.5375 \end{bmatrix}^T
\]

(4) With the information in Eqs. (16) and (33), the solution can be derived

\[
\tilde{x} = \sum_j \tilde{\beta}_j\tilde{u}_j / \phi_j = \begin{bmatrix} 5.0255 & -1.1005 & -0.6328 & 6.2534 \end{bmatrix}^T
\]

Comparing the obtained $\tilde{x}$ in Eq. (34) with the exact solution $x$ in Eq. (26), the relative error is

\[
\epsilon = \frac{\|x - \tilde{x}\|}{\|x\|} = 0.0100
\]
### Table 3 Rotation parameters and measuring probabilities of the first HIPEA variant

| Experiment 1 | Rotation parameters | Measuring probabilities |
|--------------|---------------------|-------------------------|
| Iteration 1  | $\omega_1 = -2\pi(0.0)$ | $P_0 = (P_{1.1})^T = 0.05257$ $P_1 = (P_{3.1})^T = 0.94742$ |
| Iteration 2  | $\omega_1 = -2\pi(0.0)$ | $P_{00} = (P_{1.2})^T = 0.03238$ $P_{10} = (P_{2.2})^T = 0.01898$ |
| Iteration 3  | $\omega_2 = -2\pi(0.0000)$ | $P_{000} = (P_{1.3})^T = 0.1012$ $P_{100} = (P_{0.3})^T = 0$ |
| Iteration 4  | $\omega_3 = -2\pi(0.0000)$ | $P_{0000} = (P_{1.4})^T = 0.03387$ $P_{1000} = (P_{0.4})^T = 0$ |
| Iteration 5  | $\omega_4 = -2\pi(0.00000)$ | $P_{00000} = (P_{0.5})^T = 0$ $P_{10000} = (P_{1.5})^T = 0.03372$ |
| Iteration 6  | $\omega_5 = -2\pi(0.010000)$ | $P_{010000} = (P_{0.6})^T = 0$ $P_{110000} = (P_{1.6})^T = 0.03234$ |
| Iteration 7  | $\omega_6 = -2\pi(0.0110000)$ | $P_{0110000} = (P_{0.7})^T = 0$ $P_{1110000} = (P_{1.7})^T = 0.03345$ |
| Iteration 8  | $\omega_7 = -2\pi(0.01110000)$ | $P_{01110000} = (P_{1.8})^T = 0.03245$ $P_{11110000} = (P_{0.8})^T = 0$ |
| Iteration 9  | $\omega_8 = -2\pi(0.001110000)$ | $P_{001110000} = (P_{0.9})^T = 0$ $P_{101110000} = (P_{1.9})^T = 0.0344$ |

| Experiment 2 | Rotation parameters | Measuring probabilities |
|--------------|---------------------|-------------------------|
| Iteration 1  | $\omega_1 = -2\pi(0.0)$ | $P_0 = (P_{1.1})^T = 0.05257$ $P_1 = (P_{3.1})^T = 0.94742$ |
| Iteration 2  | $\omega_2 = -2\pi(0.00)$ | $P_{00} = (P_{1.2})^T = 0.03238$ $P_{10} = (P_{2.2})^T = 0.01898$ |
| Iteration 3  | $\omega_3 = -2\pi(0.010)$ | $P_{010} = (P_{0.3})^T = 0$ $P_{110} = (P_{2.3})^T = 0.01943$ |
| Iteration 4  | $\omega_4 = -2\pi(0.0110)$ | $P_{0110} = (P_{2.4})^T = 0$ $P_{1110} = (P_{0.4})^T = 0$ |
| Iteration 5  | $\omega_5 = -2\pi(0.00110)$ | $P_{00110} = (P_{0.5})^T = 0$ $P_{10110} = (P_{2.5})^T = 0.01917$ |
| Iteration 6  | $\omega_6 = -2\pi(0.010110)$ | $P_{010110} = (P_{2.6})^T = 0.01919$ $P_{110110} = (P_{0.6})^T = 0$ |
| Experiment 2 | Rotation parameters | Measuring probabilities |
|-------------|---------------------|------------------------|
| Iteration 7 | $\omega_7 = -2\pi(0.0010110)$ | $P_{0010110} = (P_{2,7})^T = 0.01969$ $P_{1010110} = (P_{0,7})^T = 0$ |
| Iteration 8 | $\omega_8 = -2\pi(0.00010110)$ | $P_{00010110} = (P_{2,8})^T = 0.001933$ $P_{10010110} = (P_{0,8})^T = 0$ |
| Iteration 9 | $\omega_9 = -2\pi(0.000010110)$ | $P_{000010110} = (P_{2,9})^T = 0.01975$ $P_{100010110} = (P_{0,9})^T = 0$ |

| Experiment 3 | Rotation parameters | Measuring probabilities |
|-------------|---------------------|------------------------|
| Iteration 1 | $\omega_1 = -2\pi(0.0)$ | $P_0 = (P_{1,1})^T = 0.05257$ $P_1 = (P_{3,1})^T = 0.94742$ |
| Iteration 2 | $\omega_2 = -2\pi(0.01)$ | $P_{01} = (P_{0,2})^T = 0$ $P_{11} = (P_{3,2})^T = 0.94763$ |
| Iteration 3 | $\omega_3 = -2\pi(0.011)$ | $P_{011} = (P_{3,3})^T = 0.94703$ $P_{111} = (P_{3,3})^T = 0$ |
| Iteration 4 | $\omega_4 = -2\pi(0.0011)$ | $P_{0011} = (P_{3,4})^T = 0$ $P_{1011} = (P_{3,4})^T = 0.94680$ |
| Iteration 5 | $\omega_5 = -2\pi(0.0101)$ | $P_{01011} = (P_{3,5})^T = 0$ $P_{11011} = (P_{3,5})^T = 0.94644$ |
| Iteration 6 | $\omega_6 = -2\pi(0.01101)$ | $P_{011011} = (P_{3,6})^T = 0.24140$ $P_{111011} = (P_{3,6})^T = 0.70658$ |
| Iteration 7 | $\omega_7 = -2\pi(0.001101)$ | $P_{0011011} = (P_{3,7})^T = 0$ $P_{1011011} = (P_{3,7})^T = 0.24466$ |
| Iteration 8 | $\omega_8 = -2\pi(0.0011011)$ | $P_{01011011} = (P_{3,8})^T = 0$ $P_{11011011} = (P_{3,8})^T = 0.24543$ |
| Iteration 9 | $\omega_9 = -2\pi(0.01101101)$ | $P_{011011011} = (P_{3,9})^T = 0.24440$ $P_{111011011} = (P_{0,9})^T = 0$ |

| Experiment 4 | Rotation parameters | Measuring probabilities |
|-------------|---------------------|------------------------|
| Iteration 1 | $\omega_1 = -2\pi(0.0)$ | $P_0 = (P_{1,1})^T = 0.05257$ $P_1 = (P_{3,1})^T = 0.94742$ |
| Iteration 2 | $\omega_2 = -2\pi(0.01)$ | $P_{01} = (P_{0,2})^T = 0$ $P_{11} = (P_{3,2})^T = 0.94763$ |
| Iteration 3 | $\omega_3 = -2\pi(0.011)$ | $P_{01} = (P_{3,3})^T = 0.94703$ $P_{11} = (P_{0,3})^T = 0$ |
With the first variant to solve the example linear systems of equations, a total of four experiments are conducted, and there are nine iterations in each experiment. In terms of qubit resource consumption, a total of three qubits are used, with one ancillary qubit in the top register and two qubits in the bottom register.

### 4.3 Results of the second HIPEA variant

According to the quantum circuits in Fig. 5, the specific settings of the second variant in this example are: (1) Two qubits are allocated to the bottom register; (2) Nine iterations are used to estimate the eigenvalues; (3) As shown in Fig. 3b, the three divergences in the process of eigenvalue estimation appear in the first, second and sixth iterations. Accordingly, two, three and four qubits in the top register are, respectively, put into use from the second, third and seventh iterations.

The second variant described in Sect. 3.2 is applied to the example, and results in each step are as follows:

(1) After nine iterations, the four eigenvalues in Eq. (16) are extracted from the measurement results of the top register. The probabilities of obtaining these four eigenvalues are, respectively:

\[ P_{101110000} = (P_{1.9})^T = \tilde{\beta}_1^2 = 0.03352, \quad P_{000010110} = (P_{2.9})^T = \tilde{\beta}_2^2 = 0.02007 \]
\[ P_{011011011} = (P_{3.9})^T = \tilde{\beta}_3^2 = 0.24276, \quad P_{000111101} = (P_{4.9})^T = \tilde{\beta}_4^2 = 0.70180 \]

leading to

\[ |\tilde{\beta}_1| = 0.1831, \quad |\tilde{\beta}_2| = 0.1417, \quad |\tilde{\beta}_3| = 0.4927, \quad |\tilde{\beta}_4| = 0.8377 \]
The rotation parameters and measuring probabilities in each iteration are shown in Table 4.

(2) The measurement is performed on the bottom register when the measurement results of the top register are, respectively, \(\phi_1, \phi_2, \phi_3\) and \(\phi_4\), and the results are

\[
|\tilde{u}_1| = \begin{bmatrix} 0.7427 & 0.1372 & 0.0424 & 0.6541 \end{bmatrix}^T, \quad |\tilde{u}_2| = \begin{bmatrix} 0.3973 & 0.6268 & 0.5579 & 0.3717 \end{bmatrix}^T
\]

\[
|\tilde{u}_3| = \begin{bmatrix} 0.0349 & 0.6974 & 0.6979 & 0.1595 \end{bmatrix}^T, \quad |\tilde{u}_4| = \begin{bmatrix} 0.5358 & 0.3165 & 0.4460 & 0.6434 \end{bmatrix}^T
\]

(38)

(3) According to Eq. (17), the following values of \(\tilde{\beta}_j\tilde{u}_j\) are obtained:

\[
\tilde{\beta}_1\tilde{u}_1 = \begin{bmatrix} -0.1360 & 0.0251 & -0.0078 & 0.1198 \end{bmatrix}^T, \quad \tilde{\beta}_2\tilde{u}_2 = \begin{bmatrix} 0.0563 & 0.0888 & 0.0790 & 0.0527 \end{bmatrix}^T
\]

\[
\tilde{\beta}_3\tilde{u}_3 = \begin{bmatrix} -0.0172 & -0.3436 & 0.3439 & 0.0786 \end{bmatrix}^T, \quad \tilde{\beta}_4\tilde{u}_4 = \begin{bmatrix} -0.4489 & -0.2651 & -0.3736 & 0.5380 \end{bmatrix}^T
\]

(39)

(4) With the information in Eqs. (16) and (39), the solution is derived as

\[
\tilde{x} = \Sigma_j \tilde{\beta}_j\tilde{u}_j / \phi_j = \begin{bmatrix} 4.9757 & -1.0027 & -0.6098 & 6.2533 \end{bmatrix}^T
\]

(40)

Comparing the obtained solution \(\tilde{x}\) in Eq. (40) with the exact solution \(x\) in Eq. (26), the relative error is

\[
\epsilon = \frac{\|x - \tilde{x}\|}{\|x\|} = 0.0141
\]

(41)

With the second variant to solve this example linear systems of equations, only one experiment with nine iterations is conducted. In terms of qubit resource consumption, a total of six qubits are used, with four ancillary qubits in the top register and two qubits in the bottom register.

4.4 Results of the third HIPEA variant

According to the quantum circuits in Fig. 6, the specific settings of the third variant in this example are: (1) Two qubits are allocated to the bottom register; (2) The number of qubits in the top register in the third variant is flexible, which is set to be \(n_{top} = 3\) here for demonstration; (3) Since the eigenvalues can be represented by \(m = 9\) digit binary numbers, \(\lceil m/n_{top} \rceil = 3\) iterations are needed for eigenvalue estimation.

The third variant described in Sect. 3.3 is applied to the example, and results in each step are as follows:

(1) After three iterations, the four eigenvalues in Eq. (16) are extracted from the measurement results of the top register. The probabilities of obtaining these four eigenvalues are, respectively:

\[
P_{010110000} = (P_{1,9})^T = \tilde{\beta}_1^2 = 0.03361, \quad P_{000010110} = (P_{2,9})^T = \tilde{\beta}_2^2 = 0.01892
\]

\[
P_{011011011} = (P_{3,9})^T = \tilde{\beta}_3^2 = 0.24549, \quad P_{000111011} = (P_{4,9})^T = \tilde{\beta}_4^2 = 0.70108
\]

(42)
Table 4 Rotation parameters and measuring probabilities of the second HIPEA variant. Blank indicates that the corresponding qubit in the top register is not in use in this iteration.

| Top [1] | Rotation parameters       | Measuring Probabilities                  |
|---------|---------------------------|------------------------------------------|
| Iteration 1 | $\omega_1 = -2\pi(0.0)$ | $P_0 = (P_{1,1})^T = 0.05258$             |
| Iteration 2 | $\omega_2 = -2\pi(0.00)$ | $P_{00} = (P_{1,2})^T = 0.03379$          |
| Iteration 3 | $\omega_3 = -2\pi(0.000)$ | $P_{000} = (P_{1,3})^T = 0.0335$          |
| Iteration 4 | $\omega_4 = -2\pi(0.0000)$ | $P_{0000} = (P_{1,4})^T = 0.03270$        |
| Iteration 5 | $\omega_5 = -2\pi(0.00000)$ | $P_{00000} = (P_{1,5})^T = 0.03371$       |
| Iteration 6 | $\omega_6 = -2\pi(0.010000)$ | $P_{0100000} = (P_{1,6})^T = 0.03274$     |
| Iteration 7 | $\omega_7 = -2\pi(0.0110000)$ | $P_{0110000} = (P_{1,7})^T = 0.03345$     |
| Iteration 8 | $\omega_8 = -2\pi(0.01110000)$ | $P_{01110000} = (P_{1,8})^T = 0.03306$    |
| Iteration 9 | $\omega_9 = -2\pi(0.001110000)$ | $P_{011110000} = (P_{1,9})^T = 0.03352$   |

| Top [2] | Rotation parameters       | Measuring probilities                  |
|---------|---------------------------|------------------------------------------|
| Iteration 1 |                               |                                          |
| Iteration 2 | $\omega_2 = -2\pi(0.01)$ | $P_{01} = (P_{0,2})^T = 0$              |
| Iteration 3 | $\omega_3 = -2\pi(0.011)$ | $P_{011} = (P_{0,3})^T = 0.94797$       |
| Iteration 4 | $\omega_4 = -2\pi(0.0011)$ | $P_{0101} = (P_{0,4})^T = 0$            |
| Iteration 5 | $\omega_5 = -2\pi(0.01011)$ | $P_{01011} = (P_{0,5})^T = 0$           |
| Iteration 6 | $\omega_6 = -2\pi(0.011011)$ | $P_{011011} = (P_{0,6})^T = 0.24589$    |
| Iteration 7 | $\omega_7 = -2\pi(0.0011011)$ | $P_{01011011} = (P_{0,7})^T = 0$        |
| Iteration 8 | $\omega_8 = -2\pi(0.01011011)$ | $P_{01011011} = (P_{0,8})^T = 0$        |
Table 4 (continued)

| Top [2] | Rotation parameters | Measuring probabilities |
|---------|---------------------|-------------------------|
| Iteration 9 | $\omega_9^2 = -2\pi(0.011011011)$ | $P_{011011011} = (P_{3,9})^T = 0.24276$, $P_{111011011} = (P_{0,9})^T = 0$ |

| Top [3] | Rotation parameters | Measuring probabilities |
|---------|---------------------|-------------------------|
| Iteration 1 | | |
| Iteration 2 | | |
| Iteration 3 | $\omega_3^3 = -2\pi(0.010)$ | $P_{010} = (P_{0,3})^T = 0$, $P_{110} = (P_{2,3})^T = 0.01907$ |
| Iteration 4 | $\omega_4^3 = -2\pi(0.01110)$ | $P_{0110} = (P_{2,4})^T = 0.01906$, $P_{1110} = (P_{0,4})^T = 0$ |
| Iteration 5 | $\omega_5^3 = -2\pi(0.00110)$ | $P_{00110} = (P_{0,5})^T = 0$, $P_{10110} = (P_{2,5})^T = 0.01954$ |
| Iteration 6 | $\omega_6^3 = -2\pi(0.010110)$ | $P_{010110} = (P_{2,6})^T = 0.01957$, $P_{110110} = (P_{0,6})^T = 0$ |
| Iteration 7 | $\omega_7^3 = -2\pi(0.0010110)$ | $P_{0010110} = (P_{4,7})^T = 0.01965$, $P_{1010110} = (P_{0,7})^T = 0$ |
| Iteration 8 | $\omega_8^3 = -2\pi(0.00010110)$ | $P_{00010110} = (P_{2,8})^T = 0.70406$, $P_{10010110} = (P_{0,8})^T = 0$ |
| Iteration 9 | $\omega_0^3 = -2\pi(0.00001100)$ | $P_{00001100} = (P_{2,9})^T = 0.02007$, $P_{10001100} = (P_{0,9})^T = 0$ |

| Top [4] | Rotation parameters | Measuring Probabilities |
|---------|---------------------|-------------------------|
| Iteration 1 | | |
| Iteration 2 | | |
| Iteration 3 | | |
| Iteration 4 | | |
| Iteration 5 | | |
| Iteration 6 | | |
| Iteration 7 | $\omega_9^4 = -2\pi(0.0111011)$ | $P_{0111011} = (P_{4,7})^T = 0.70208$, $P_{1111011} = (P_{0,7})^T = 0$ |
| Iteration 8 | $\omega_8^4 = -2\pi(0.00111011)$ | $P_{00111011} = (P_{4,8})^T = 0.7039$, $P_{10111011} = (P_{0,8})^T = 0$ |
| Iteration 9 | $\omega_9^4 = -2\pi(0.000111011)$ | $P_{000111011} = (P_{4,9})^T = 0.7018$, $P_{100111011} = (P_{0,9})^T = 0$ |
leading to

\[
\begin{align*}
|\tilde{\beta}_1| &= 0.1833, \\
|\tilde{\beta}_2| &= 0.1375, \\
|\tilde{\beta}_3| &= 0.4955, \\
|\tilde{\beta}_4| &= 0.8373
\end{align*}
\]  
(43)

The rotation parameters and measuring probabilities in each iteration are shown in Table 5.

(2) The measurement is performed on the bottom register when the measurement results of the top register are, respectively, \(\phi_1, \phi_2, \phi_3\) and \(\phi_4\), and the results are

\[
\begin{align*}
|\tilde{u}_1| &= \begin{bmatrix} 0.7339 & 0.1295 & 0.0629 & 0.6639 \end{bmatrix}^T, \\
|\tilde{u}_2| &= \begin{bmatrix} 0.4018 & 0.6263 & 0.5617 & 0.3618 \end{bmatrix}^T, \\
|\tilde{u}_3| &= \begin{bmatrix} 0.0287 & 0.6984 & 0.6977 & 0.1568 \end{bmatrix}^T, \\
|\tilde{u}_4| &= \begin{bmatrix} 0.5355 & 0.3227 & 0.4433 & 0.6423 \end{bmatrix}^T
\end{align*}
\]  
(44)

(3) According to Eq. (17), the following values of \(\tilde{\beta}_j\tilde{u}_j\) are obtained:

\[
\begin{align*}
\tilde{\beta}_1\tilde{u}_1 &= \begin{bmatrix} -0.1345 & 0.0237 & -0.0115 & 0.1217 \end{bmatrix}^T, \\
\tilde{\beta}_2\tilde{u}_2 &= \begin{bmatrix} 0.0553 & 0.0861 & 0.0773 & 0.0498 \end{bmatrix}^T, \\
\tilde{\beta}_3\tilde{u}_3 &= \begin{bmatrix} -0.0142 & -0.3460 & 0.3457 & 0.0777 \end{bmatrix}^T, \\
\tilde{\beta}_3\tilde{u}_3 &= \begin{bmatrix} 0.4484 & -0.2702 & -0.3712 & 0.5378 \end{bmatrix}^T
\end{align*}
\]  
(45)

(4) With the information in Eqs. (45) and (26), the solution is derived as

\[
\tilde{x} = \Sigma_j \tilde{\beta}_j\tilde{u}_j / \phi_j = \begin{bmatrix} 4.9568 & -1.1158 & -0.6308 & 6.1782 \end{bmatrix}^T
\]  
(46)

Comparing the obtained solution \(\tilde{x}\) in Eq. (46) with the exact solution \(x\) in Eq. (26), the relative error is

\[
\epsilon = \frac{\| x - \tilde{x} \|}{\| x \|} = 0.0048
\]  
(47)

With the third variant to solve this example linear systems of equations, only three iterations are required, and a total of 32 experiments are conducted. In terms of qubit resource consumption, a total of five qubits are used, with three ancillary qubits in the top register and two qubits in the bottom register.

5 Conclusions

In this paper, one hybrid classical-quantum algorithm with three variants is designed to solve linear systems of equations \(Ax = b\) based on the iterative phase estimation algorithm, and less qubit resources are required in this hybrid algorithm than in quantum HHL algorithm. The hybrid algorithm stores the measurement results of each iteration of each experiment in the classic register, and the non-zero measuring probabilities of states of the ancillary qubits in the top register are used to extract the eigenvalues of \(A\) and the corresponding projection coefficients \(\beta_j\) in \(b = \Sigma_j \beta_ju_j\). When measuring
### Table 5 Rotation parameters and measuring probabilities of the third HIPEA variant

(a) The first iteration

| Experiment   | Iteration 1 | Measuring Probabilities |
|--------------|-------------|-------------------------|
|              | $\omega_1 = -2\pi(0.0)$ | $P_{000} = (P_{1.3})^T = 0.03362$ |
|              | $\omega_2 = -2\pi(0.00)$ | $P_{100} = (P_{0.3})^T = 0$ |
|              | $\omega_3 = -2\pi(0.000)$ | $P_{001} = (P_{0.3})^T = 0$ |
|              |             | $P_{101} = (P_{0.3})^T = 0$ |
|              |             | $P_{110} = (P_{2.3})^T = 0.01931$ |
|              |             | $P_{011} = (P_{3.3})^T = 0.94844$ |
|              |             | $P_{111} = (P_{0.3})^T = 0$ |

(b) The second iteration

| Experiment   | Iteration 2 | Measuring Probabilities |
|--------------|-------------|-------------------------|
|              |             | $P_{00000} = (P_{0.6})^T = 0$ |
|              |             | $P_{10000} = (P_{0.6})^T = 0$ |
|              |             | $P_{00100} = (P_{0.6})^T = 0$ |
|              |             | $P_{10100} = (P_{0.6})^T = 0$ |
|              |             | $P_{01100} = (P_{0.6})^T = 0$ |
|              |             | $P_{11100} = (P_{0.6})^T = 0$ |
Table 5 (continued)

(b) The second iteration

| Iteration 2 |  $\omega_4 = -2\pi(0.01110)$ |  $\omega_4 = -2\pi(0.01110)$ |
|--------------|-------------------------------|-------------------------------|
|              |  $\omega_5 = -2\pi(0.00110)$ |  $\omega_5 = -2\pi(0.00110)$ |
|              |  $\omega_6 = -2\pi(0.000111)$ |  $\omega_6 = -2\pi(0.000111)$ |

Measuring Probabilities

| Experiment 11 | $P_{000100} = (P_{0,6})^T = 0$ | $P_{00110} = (P_{0,6})^T = 0$ |
|---------------|--------------------------------|--------------------------------|
| Experiment 12 | $P_{001110} = (P_{0,6})^T = 0$ | $P_{10110} = (P_{0,6})^T = 0$ |
|               | $P_{010110} = (P_{2,6})^T = 0.0201$ | |
| Experiment 13 | $P_{10110} = (P_{0,6})^T = 0$ | $P_{206780} = (P_{0,6})^T = 0$ |
|               | $P_{2010110} = (P_{0,6})^T = 0$ | $P_{110110} = (P_{0,6})^T = 0$ |
|               | $P_{010110} = (P_{2,6})^T = 0$ | $P_{346789} = (P_{0,6})^T = 0$ |
|               | $P_{110110} = (P_{0,6})^T = 0$ | $P_{110110} = (P_{0,6})^T = 0$ |

| Experiment 14 | $P_{001110} = (P_{2,6})^T = 0$ | $P_{346789} = (P_{0,6})^T = 0$ |
|               | $P_{010110} = (P_{2,6})^T = 0$ | $P_{346789} = (P_{0,6})^T = 0$ |
|               | $P_{010110} = (P_{2,6})^T = 0$ | $P_{346789} = (P_{0,6})^T = 0$ |

| Experiment 15 | $P_{011110} = (P_{0,6})^T = 0$ | $P_{110110} = (P_{0,6})^T = 0$ |
|               | $P_{111110} = (P_{0,6})^T = 0$ | $P_{010111} = (P_{3,6})^T = 0.24537$ |
|               | $P_{111110} = (P_{0,6})^T = 0$ | $P_{110111} = (P_{4,6})^T = 0.70272$ |
| Experiment 16 | $P_{011110} = (P_{0,6})^T = 0$ | $P_{110111} = (P_{4,6})^T = 0.70272$ |
|               | $P_{011110} = (P_{0,6})^T = 0$ | $P_{110111} = (P_{4,6})^T = 0.70272$ |


Table 5 (continued)

(c) The third iteration

| Experiment 17 | Experiment 18 |
|---------------|---------------|
| $\omega_7 = -2\pi(0.0110000)$ | $\omega_7 = -2\pi(0.0110000)$ |
| $\omega_8 = -2\pi(0.00110000)$ | $\omega_8 = -2\pi(0.00110000)$ |
| $\omega_9 = -2\pi(0.000110000)$ | $\omega_9 = -2\pi(0.000110000)$ |
| Measuring Probabilities | Measuring Probabilities |
| $P_{000110000} = (P_{0,9})^T = 0$ | $P_{100110000} = (P_{0,9})^T = 0$ |
| $P_{101100000} = (P_{0,9})^T = 0$ | $P_{011100000} = (P_{0,9})^T = 0$ |
| $P_{010110000} = (P_{0,9})^T = 0$ | $P_{110110000} = (P_{0,9})^T = 0$ |
| Experiment 19 | Experiment 20 |
| Meetin Probabilities | Meetin Probabilities |
| $P_{001011000} = (P_{2,9})^T = 0.01892$ | $P_{100101100} = (P_{0,9})^T = 0$ |
| $P_{101010110} = (P_{0,9})^T = 0$ | $P_{010101100} = (P_{0,9})^T = 0$ |
| Experiment 21 | Experiment 22 |
| $\omega_7 = -2\pi(0.010110)$ | $\omega_7 = -2\pi(0.010110)$ |
| $\omega_8 = -2\pi(0.00010110)$ | $\omega_8 = -2\pi(0.0010110)$ |
| $\omega_9 = -2\pi(0.00010110)$ | $\omega_9 = -2\pi(0.0010110)$ |
| Measuring Probabilities | Measuring Probabilities |
| $P_{000010110} = (P_{0,9})^T = 0.04291$ | $P_{100010110} = (P_{0,9})^T = 0$ |
| $P_{101010110} = (P_{0,9})^T = 0$ | $P_{010010110} = (P_{0,9})^T = 0$ |
| Experiment 23 | Experiment 24 |
| $\omega_7 = -2\pi(0.0010110)$ | $\omega_7 = -2\pi(0.0010110)$ |
| $\omega_8 = -2\pi(0.00010110)$ | $\omega_8 = -2\pi(0.01010110)$ |
| $\omega_9 = -2\pi(0.01010110)$ | $\omega_9 = -2\pi(0.011010110)$ |
### Table 5 (continued)

#### (c) The third iteration

| Measuring Probabilities | Experiment 25 | Experiment 26 | Experiment 27 | Experiment 28 | Experiment 29 | Experiment 30 | Experiment 31 | Experiment 32 |
|-------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| $P_{00101110} = (P_{0,9})^T = 0$ | $P_{10010110} = (P_{0,9})^T = 0$ | $P_{01101010} = (P_{0,9})^T = 0$ | $P_{11101010} = (P_{0,9})^T = 0$ | $P_{10011011} = (P_{0,9})^T = 0$ | $P_{11011011} = (P_{0,9})^T = 0$ | $P_{11101011} = (P_{0,9})^T = 0$ | $P_{01111101} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ |
| Iteration 3             | $\omega_7 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0111011)$ | $\omega_7 = -2\pi(0.0111011)$ |
| $P_{00011011} = (P_{0,9})^T = 0$ | $P_{10011011} = (P_{0,9})^T = 0$ | $P_{01101111} = (P_{0,9})^T = 0$ | $P_{11010111} = (P_{0,9})^T = 0$ | $P_{10101011} = (P_{0,9})^T = 0$ | $P_{11111011} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ | $P_{01111011} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ |
| Iteration 3             | $\omega_7 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0111011)$ | $\omega_7 = -2\pi(0.0111011)$ |
| $P_{01001101} = (P_{0,9})^T = 0$ | $P_{11001101} = (P_{0,9})^T = 0$ | $P_{01101111} = (P_{0,9})^T = 0$ | $P_{10101111} = (P_{0,9})^T = 0$ | $P_{10110111} = (P_{0,9})^T = 0$ | $P_{11110111} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ | $P_{01111011} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ |
| Measuring Probabilities | Experiment 26 | Experiment 27 | Experiment 28 | Experiment 29 | Experiment 30 | Experiment 31 | Experiment 32 |
| $P_{00011101} = (P_{0,9})^T = 0.70108$ | $P_{10011101} = (P_{0,9})^T = 0.24549$ | $P_{01101111} = (P_{0,9})^T = 0$ | $P_{11011111} = (P_{0,9})^T = 0$ | $P_{10101111} = (P_{0,9})^T = 0$ | $P_{11111111} = (P_{0,9})^T = 0$ | $P_{11111111} = (P_{0,9})^T = 0$ | $P_{01111111} = (P_{0,9})^T = 0$ | $P_{11111111} = (P_{0,9})^T = 0$ |
| Iteration 3             | $\omega_7 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0111011)$ | $\omega_7 = -2\pi(0.0111011)$ |
| $P_{00111101} = (P_{0,9})^T = 0$ | $P_{10011101} = (P_{0,9})^T = 0$ | $P_{01101111} = (P_{0,9})^T = 0$ | $P_{11011111} = (P_{0,9})^T = 0$ | $P_{10101111} = (P_{0,9})^T = 0$ | $P_{11111111} = (P_{0,9})^T = 0$ | $P_{11111111} = (P_{0,9})^T = 0$ | $P_{01111111} = (P_{0,9})^T = 0$ | $P_{11111111} = (P_{0,9})^T = 0$ |
| Measuring Probabilities | Experiment 30 | Experiment 31 | Experiment 32 | Experiment 31 | Experiment 32 | Experiment 32 | Experiment 32 | Experiment 32 |
| $P_{00111011} = (P_{0,9})^T = 0$ | $P_{10011011} = (P_{0,9})^T = 0$ | $P_{01101011} = (P_{0,9})^T = 0$ | $P_{11010101} = (P_{0,9})^T = 0$ | $P_{10101011} = (P_{0,9})^T = 0$ | $P_{11101011} = (P_{0,9})^T = 0$ | $P_{11110101} = (P_{0,9})^T = 0$ | $P_{01111011} = (P_{0,9})^T = 0$ | $P_{11111011} = (P_{0,9})^T = 0$ |
| Iteration 3             | $\omega_7 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_8 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_9 = -2\pi(0.0011011)$ | $\omega_7 = -2\pi(0.0111011)$ | $\omega_7 = -2\pi(0.0111011)$ |
| $P_{00101011} = (P_{0,9})^T = 0$ | $P_{10010111} = (P_{0,9})^T = 0$ | $P_{01101101} = (P_{0,9})^T = 0$ | $P_{11011101} = (P_{0,9})^T = 0$ | $P_{10101101} = (P_{0,9})^T = 0$ | $P_{11101101} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ | $P_{01111101} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ |
| Measuring Probabilities | Experiment 31 | Experiment 32 | Experiment 32 | Experiment 32 | Experiment 32 | Experiment 32 | Experiment 32 | Experiment 32 |
| $P_{00101011} = (P_{0,9})^T = 0$ | $P_{10010111} = (P_{0,9})^T = 0$ | $P_{01101101} = (P_{0,9})^T = 0$ | $P_{11011101} = (P_{0,9})^T = 0$ | $P_{10101101} = (P_{0,9})^T = 0$ | $P_{11101101} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ | $P_{01111101} = (P_{0,9})^T = 0$ | $P_{11111101} = (P_{0,9})^T = 0$ |
with post-selection, the absolute value $|u_{jp}|$ of each element of the normalized eigenvector $u_j$ of $A$ can be obtained. The final solution will then be derived after the values of $\beta_j u_j$ in $b = \Sigma_j \beta_j u_j$ are determined.

The number of qubits in the bottom register is the same in our three variants of HIPEA, which is determined by the dimension of the column vector $b$, while the number of ancillary qubits in the top register is different. The first variant requires the least number of ancillary qubits (i.e., 1), and the eigenvalues of matrix are differentiated by conducting different experiments with $N$ iterations. The second variant achieves the same effect with multiple ancillary qubits (i.e., $N$) instead of multiple experiments, and the complexity just moves from “the number of experiments” to “the number of ancillary qubits”. The third variant differentiates different eigenvalues with flexible multiple ancillary qubits using a traversing strategy, and a set of experiments is performed in each iteration to traverse one bit by one bit all possible combinations of rotation parameters and thus all possible binary number strings of eigenvalues. The hybrid algorithm is implemented in PyQPanda, and one specific example is employed to prove the feasibility of this algorithm when the eigenvalues of $A$ can be perfectly represented by finite binary number strings. The developed HIPEA algorithm with limited qubit resources broadens the application range of quantum computation in solving linear systems of equations.

In future studies, it is worthy to modify HIPEA algorithm or develop new hybrid algorithms to solve $Ax = b$ when the eigenvalues of $A$ cannot be perfectly represented by finite binary number strings. Moreover, how to calibrate the signs of $\beta_j u_{jp}$ with a more universal and rigorous scheme is also a question worth considering. In case that the eigenvalues and eigenvectors of the coefficient matrix $A$ are known in advance, we do not need to design redundant rotation parameters, and only have to focus on how to extract the coefficients $\beta_j$ efficiently. Since modules of initial state preparation, Hamiltonian simulation and accurate measurement are included in HIPEA, the improvement schemes of these modules may possibly help to improve the performance of HIPEA.

**Funding** National natural science foundation of china,61773359,Fang Gao,61720106009,Feng Shuang,61873317,Wei Cui

**References**

1. Cai, X.D., Weedbrook, C., Su, Z.E., Chen, M.C., Gu, M., Zhu, M.J., Li, L., Le Liu, N., Lu, C.Y., Pan, J.W.: Experimental quantum computing to solve systems of linear equations. Phys. Rev. Lett. 110, 1–5 (2013). https://doi.org/10.1103/PhysRevLett.110.230501
2. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Phys. Rev. Lett. (2009). https://doi.org/10.1103/PhysRevLett.103.150502
3. Schleich, P.: How to solve a linear system of equations using a quantum computer. Semin. Proj. (2019) 1–35. www.mathcces.rwth-aachen.de/_media/3teaching/00projects/schleich.pdf
4. Shao, C.: Reconsider hhl algorithm and its related quantum machine learning algorithms. arXiv preprint arXiv:1803.01486 (2018)
5. Dickens, J.: Quantum Computing Algorithms for Applied Linear Algebra (2019)
6. Carrera Vázquez, A., Wörner, S., Hiptmair, R.: Quantum algorithm for solving tri-diagonal linear systems of equations, (2018) 1–24
7. Duan, B., Yuan, J., Yu, C.H., Huang, J., Hsieh, C.Y.: A survey on HHL algorithm: from theory to application in quantum machine learning. Phys. Lett. Sect. A Gen. At. Solid State Phys. 384, 126595 (2020) https://doi.org/10.1016/j.physleta.2020.126595
8. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum 2, 1–20 (2018). https://doi.org/10.22331/q-2018-08-06-79
9. Lee, Y., Joo, J., Lee, S.: Hybrid quantum linear equation algorithm and its experimental test on IBM Quantum Experience. Sci. Rep. 9, 1–12 (2019). https://doi.org/10.1038/s41598-019-41324-9
10. Bužek, V., Derka, R., Massar, S.: Optimal quantum clocks. Asymptot. Theory Quantum Stat. Inference Sel. Pap. (2005). https://doi.org/10.1142/9789812563071_0032
11. Svore, K.M., Hastings, M.B., Freedman, M.: Faster phase estimation. Quantum Inf. Comput. 14, 306–328 (2014). https://doi.org/10.26421/QIC14.3-4-7
12. Cleve, R., Ekert, A., Macchiavello, C., Mosca, M.: Quantum algorithms revisited, Proc. R. Soc. A Math. Phys. Eng. Sci. 454, 339–354 (1998). https://doi.org/10.1098/rspa.1998.0164
13. Zhou, X.Q., Kalasuwan, P., Ralph, T.C., O’Brien, J.L.: Calculating unknown eigenvalues with a quantum algorithm. Nat. Photonics. 7, 223–228 (2013). https://doi.org/10.1038/nphoton.2012.360
14. Parasa, V., Perkowski, M.: Quantum phase estimation using multivalued logic. In: Proceedings of the 2011 41st IEEE International Symposium on Multiple-valued logic, ISMVL 2011. (2011) 224–229. https://doi.org/10.1109/ISMVL.2011.47
15. O’Brien, T.E., Tarasinski, B., Terhal, B.M.: Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments. New J. Phys. (2019). https://doi.org/10.1088/1367-2630/aafb8e
16. Wiebe, N., Granade, C.: Efficient Bayesian phase estimation. Phys. Rev. Lett. 117(1), 010503 (2016). https://doi.org/10.1103/PhysRevLett.117.010503
17. O’Loan, C.J.: Iterative phase estimation. J. Phys. A Math. Theor. (2010). https://doi.org/10.1088/1751-8113/43/1/015301
18. Dobšíček, M., Johansson, G., Shumeiko, V., Wendin, G.: Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark, Phys. Rev. A - At. Mol. Opt. Phys. 76, 1–4 (2007). https://doi.org/10.1103/PhysRevA.76.030306
19. Liu, X.M., Luo, J., Sun, X.P.: Experimental realization of arbitrary accuracy iterative phase estimation algorithms on ensemble quantum computers. Chin. Phys. Lett. 24, 3316–3319 (2007). https://doi.org/10.1088/0256-307X/24/12/007
20. Long, G.-L.: General quantum interference principle and duality computer. Commun. Theor. Phys. 45, 825 (2006). https://doi.org/10.1088/0253-6102/45/5/013
21. Long, G.L.: Duality Quantum Computing and Duality Quantum Information Processing. Int. J. Theor. Phys. 50, 1305–1318 (2011). https://doi.org/10.1007/s10773-010-0603-z
22. Shao, C., Li, Y., Li, H.: Quantum Algorithm Design: Techniques and Applications. J. Syst. Sci. Complex. 32, 375–452 (2019). https://doi.org/10.1007/s11424-019-9008-0
23. Wei, S., Li, H., Long, G.: A Full Quantum Eigensolver for Quantum Chemistry Simulations. Res. 2020, 1486935 (2020). https://doi.org/10.34133/2020/1486935
24. Jin, S., Wu, S., Zhou, G., Li, Y., Li, L., Li, B., Wang, X.: A query-based quantum eigensolver. Quantum Eng. 2, e49 (2020). https://doi.org/10.1002/que2.49
25. Gao, P., Li, K., Wei, S., Long, G.L.: Quantum second-order optimization algorithm for general polynomials. Sci. China Physics Mech. Astron. 64, 100311 (2021). https://doi.org/10.1007/s11433-021-1725-9
26. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum random access memory. Phys. Rev. Lett. 100, 1–4 (2008). https://doi.org/10.1103/PhysRevLett.100.160501
27. Giovannetti, V., Lloyd, S., Maccone, L.: Architectures for a quantum random access memory, Phys. Rev. A At. Mol. Opt. Phys. 78, 1–9 (2008). https://doi.org/10.1103/PhysRevA.78.052310
28. Childs, A.M., Wiebe, N.: Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. 12, 901–924 (2012). https://doi.org/10.26421/qic12.11-12-1
29. Berry, D.W., Childs, A.M.: Black-box hamiltonian simulation and unitary implementation. Quantum Inf. Comput. 12, 29–62 (2012). https://doi.org/10.26421/QIC12.1-2
30. Nielsen, M.A., Bremmer, M.J., Dodd, J.L., Childs, A.M., Dawson, C.M.: Universal simulation of Hamiltonian dynamics for quantum systems with finite-dimensional state spaces, Phys. Rev. A - At. Mol. Opt. Phys. 66, 1–12 (2002). https://doi.org/10.1103/PhysRevA.66.022317
31. Low, G.H., Chuang, I.L.: Optimal Hamiltonian Simulation by Quantum Signal Processing. Phys. Rev. Lett. 118, 1–5 (2017). https://doi.org/10.1103/PhysRevLett.118.010501
32. Santagati, R., Wang, J., Gentile, A.A., Paesani, S., Wiebe, N., McClean, J.R., Morley-Short, S., Shadbolt, P.J., Bonneau, D., Silverstone, J.W., Tew, D.P., Zhou, X., O’Brien, J.L., Thompson, M.G.: Witnessing eigenstates for quantum simulation of Hamiltonian spectra. Sci. Adv. 4, 1–12 (2018). https://doi.org/10.1126/sciadv.aap9646

33. Berry, D.W., Ahokas, G., Cleve, R., Sanders, B.C.: Efficient quantum algorithms for simulating sparse hamiltonians. Commun. Math. Phys. 270, 359–371 (2007). https://doi.org/10.1007/s00220-006-0150-x

34. Long, G.-L.: Collapse-in and Collapse-out in Partial Measurement in Quantum Mechanics and its WISE Interpretation. Sci. China Physics Mech. Astron. 64, 280321 (2021). https://doi.org/10.1007/s11433-021-1716-y

35. Clader, B.D., Jacobs, B.C., Sprouse, C.R.: Preconditioned quantum linear system algorithm. Phys. Rev. Lett. 110, 1–5 (2013). https://doi.org/10.1103/PhysRevLett.110.250504

36. Dervovic, D., Herbster, M., Mountney, P., Severini, S., Usher, N., Wossnig, L.: Quantum linear systems algorithms: a primer. arXiv preprint arXiv:1802.08227 (2018)

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