Quantum circulant preconditioner for a linear system of equations

Changpeng Shao[*]
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

Hua Xiang[†]
School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

(Received 13 July 2018; revised manuscript received 18 November 2018; published 18 December 2018)

We consider the quantum linear solver for $Ax = b$ with the circulant preconditioner C. The main technique is the singular value estimation (SVE) introduced in [Kerenidis and Prakash, Quantum recommendation system, in ITCS (2017)]. However, the SVE should be modified to solve the preconditioned linear system $C^{-1}Ax = C^{-1}b$. Moreover, different from the preconditioned linear system considered in [Phys. Rev. Lett. **110**, 250504 (2013)], the circulant preconditioner is easy to construct and can be directly applied to general dense non-Hermitian cases. The time complexity depends on the condition numbers of C and $C^{-1}A$, as well as the Frobenius norm $\|A\|_F$.

DOI: 10.1103/PhysRevA.98.062321

I. INTRODUCTION

Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, the linear system of equations $Ax = b$ is a basic problem in scientific computing. The classical methods include direct methods and iterative methods. Gauss elimination with partial pivoting (GEPP) is the generally used direct method. The Jacobi method, Gauss–Seidel method, and the successive overrelaxation method (SOR) are typical classical iterative methods, and Krylov subspace methods, such as CG, GMRES, BiCGStab, etc. are the modern iterative methods [1–3]. For a general dense matrix, GEPP costs $O(n^3)$; and for a symmetric positive-definite matrix, the CG method runs with $O(ns\sqrt{\kappa}\log 1/\epsilon)$, where κ is the conditioner number, s and ϵ stand for the sparsity and the precision, respectively.

The first quantum algorithm to solve sparse linear system was proposed by Harrow, Hassidim, and Lloyd [4] in 2009, currently known as the HHL algorithm for short. It is exponentially faster than any classical method by calculating the quantum state of the solution, within a running time of $O((\log n)s^2\kappa^2/\epsilon)$. Subsequent works have improved the running time of the HHL algorithm to be linear in κ [5] and the precision dependence to be polynomial in $\log(1/\epsilon)$ [6]. Ambainis [5] reduced the condition number dependence from κ^2 to $\kappa\log^3\kappa$. Further work by Childs, Kothari, and Somm [6] reduced the precision number dependency of the algorithm from $O(\text{poly}(1/\epsilon))$ to $O(\text{poly}\log(1/\epsilon))$. The main idea of the HHL algorithm is the singular value decomposition (SVD) based on Hamiltonian simulation. In 2017, Kerenidis and Prakash [7] proposed a different method to achieve the SVD, named by the singular value estimation (SVE), with the introduction of a new data structure of quantum information that similar to the idea of qRAM [8]. Later, based on this work, Wossnig, Zhao, and Prakash presented the quantum algorithm [9] to general dense linear systems that takes time $O(\kappa^2\sqrt{n}\,\text{poly}(\log n)/\epsilon)$, a polynomial speedup for dense matrices. Wang and Wossnig [10] applied this method for dense Hamiltonian simulation. Some other applications of SVE are given in Refs. [7] and [11]. The HHL algorithm has wide applications, such as data processing [12], numerical calculation [13], artificial intelligence [14,15], neural networks [16], and so on. It is experimentally demonstrated with parametric down-converted single photons [17,18], liquid nuclear magnetic resonance [19], and a scalable superconducting quantum circuit [20].

We notice that the condition number κ of A plays an important role in the time complexity for both the classical and quantum algorithms. To reduce the dependence on condition number, one important technique is the preconditioning, and we need to solve a preconditioned linear system $MAx = Mb$ instead, where the preconditioner M is chosen such that $M \approx A^{-1}$. The iterative methods are successful only if there exists an effective preconditioner. For example, the classical CG on the typical second-order elliptic boundary value problems in three dimensions (3D), using the preconditioner can reduce the conditioner number from $O(n^{2/3})$ to $O(n^{1/3})$, and the time complexity of $O(n^{4/3})$ decreases to $O(n^{7/6})$. There exists many preconditioning techniques for the classical methods [1–3], including the algebraic multigrid method, the domain decomposition method, etc. To the best of our knowledge, there is only one work related to the quantum preconditioning [21]. To improve the efficiency of the quantum linear solver, Clader *et al.* [21] chose a sparse approximate inverse (SPAI) preconditioner M. It needs a unitary operator to calculate the elements of MA. The oracle for the matrix MA can be created by using the original oracle for A with only modest overhead of $O(s^3)$ in runtime and $O(s^2)$ in query complexity. Under the sparsity assumption of MA, this work improves the complexity of the HHL algorithm to $O(s^7\kappa(MA)(\log n)/\epsilon)$.

In this paper, we consider another kind of preconditioner, a circulant preconditioner C. Different from the SPAI used in

[*] cpshao@amss.ac.cn
[†] Corresponding author: hxiang@whu.edu.cn

Ref. [21], the circulant preconditioner is more general [22], suitable for the general dense linear systems. Moreover, the circulant preconditioner C contains a simple structure. It can be diagonalized by the Fourier transformation. In quantum computing, the quantum Fourier transformation can be implemented efficiently. Hence in some sense C can be viewed just as a diagonal matrix. Such a quantum preconditioner is easy to construct and suitable for quantum implementation. The only difficulty that lies in the construction of circulant preconditioner is that the eigenvalues of C is given by a summation. Direct calculation based on such a formula costs at least $O(n)$, which kills the exponential speedup of the quantum algorithm. Therefore we should find another efficient method to obtain them. The main technique that we will use to solve the preconditioned linear system $C^{-1}Ax = C^{-1}b$ is the SVE. However, the SVE given in Ref. [7] is not sufficient to our problem here, since C is not Hermitian and should be provided in a quantum state form. So we need to make some modifications about the SVE method introduced in Ref. [7]. Assuming the SVD of $A = \sum \sigma_i |u_i\rangle\langle v_i|$, then the SVE given in Ref. [7] achieves $\sum \alpha_i |v_i\rangle \mapsto \sum \alpha_i |v_i\rangle |\sigma_i\rangle$. However, it will be more helpful to us if we can achieve $\sum \alpha_i |v_i\rangle \mapsto \sum \alpha_i |u_i\rangle |\sigma_i\rangle$ or $\sum \alpha_i |u_i\rangle \mapsto \sum \alpha_i |v_i\rangle |\sigma_i\rangle$. This can be achieved by making some modifications to the SVE proposed in Ref. [7]. For Hermitian matrix A, we know that $|u_i\rangle = |v_i\rangle$. For the non-Hermitian matrix A, as introduced in the HHL algorithm [4], the SVE in Ref. [7] works on

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$$

instead. However, if A is given as quantum information, then it may not be easy to expand A into a Hermitian matrix. But our modified SVE method does not need such expansion and works well on the original non-Hermitian matrix, and hence it can solve the preconditioned linear system $C^{-1}Ax = C^{-1}b$ more efficiently.

The structure of this paper is as follows: In Sec. II, we briefly review the basic results of classical circulant preconditioner. Then in Sec. III, we introduce the modified SVE method and apply it to solve the preconditioned linear system. We also consider the physical implementation of our algorithm in a quantum circuit.

II. CIRCULANT PRECONDITIONER

In this section, we briefly review some basic known results about circulant preconditioner given by Strang [23], Chan [22], and Tyrtyshnikov [24]. A circulant preconditioner is defined by an n-by-n circulant matrix

$$C = (c_{ij})_{n\times n} = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \cdots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & \cdots & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix}, \quad (1)$$

where the entry $c_{ij} = c_{(i-j) \bmod n}$. Obviously, the matrix C is totally determined by its first column. Let Q be the following shift permutation matrix:

$$Q = \begin{pmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix},$$

then $C = \sum_{j=0}^{n-1} c_j Q^j$. In Ref. [25], Zhou and Wang applied this decomposition for the Hamiltonian simulation of C and solve the linear system $Cx = b$.

The circulant matrix can be diagonalized by the Fourier matrix $F = (\frac{1}{\sqrt{n}}\omega^{jk})_{n\times n}$, where $\omega = e^{-2\pi i/n}$. That is, there is a diagonal matrix $\Lambda = \text{diag}(\lambda_0, \ldots, \lambda_{n-1})$, which refers to the eigenvalues of C, such that

$$C = F^\dagger \Lambda F. \quad (2)$$

More precisely, if we set $e_0 = (1, 0, \ldots, 0)^\dagger$ and $e = (1, 1, \ldots, 1)^\dagger$, then $FCe_0 = \Lambda F e_0 = \frac{1}{\sqrt{n}}\Lambda e$. Note that $Ce_0 = (c_0, c_1, \ldots, c_{n-1})^\dagger$ and $\Lambda e = (\lambda_0, \lambda_1, \ldots, \lambda_{n-1})^\dagger$, so

$$\lambda_k = \sum_{j=0}^{n-1} c_j \omega^{jk}. \quad (3)$$

Let U is a unitary matrix. Define

$$\mathcal{M}_U := \{U^\dagger \Lambda_n U | \Lambda_n \text{ is an } n \times n \text{ diagonal matrix}\}.$$

Then $\mathcal{M}_F$ is the set of all circulant matrices.

The Strang preconditioner is designed for the Toeplitz matrix

$$T = (t_{ij})_{n\times n} = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & \ddots & \cdots & t_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{n-2} & \cdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{pmatrix},$$

i.e., $t_{ij} = t_{i-j}$, determined by $2n-1$ entries. The matrix name arises from Toeplitz's work on bilinear forms associated with Laurant series. A circulant matrix is a special case of a Toeplitz matrix with $t_{-k} = t_{n-k}$ for $1 \leqslant k \leqslant n-1$. The Toeplitz linear system $Tx = b$ appears in a variety of applications, such as signal processing, control theory, networks, integral equations, etc. The quantum algorithm to the Toeplitz linear system $Tx = b$ has been considered in Ref. [26]. For such linear systems, Strang [23] proposed a circulant preconditioner $s_F(T)$ which satisfies [27]

$$s_F(T) = \arg\min_{C\in\mathcal{M}_F} \|T - C\|_1.$$

For simplicity, we assume that $n = 2m + 1$, while the case $n = 2m$ can be treated similarly. The Strang preconditioner $s_F(T)$ is a circulant matrix defined by its first column $s = [s_0, \ldots, s_{n-1}]^T$, where

$$s_k = \begin{cases} t_k, & 0 \leqslant k \leqslant m \\ t_{k-n}, & m \leqslant k \leqslant n-1. \end{cases}$$

Chan [22] proposed the optimal circulant preconditioner for solving Toeplitz systems and extended it for general matrices. For an arbitrary matrix A, it can prove that

$$c_U(A) := U^\dagger \mathrm{diag}(UAU^\dagger)U = \arg\min_{W\in\mathcal{M}_U} \|A-W\|_F, \quad (4)$$

where $\|\cdot\|_F$ is the Frobenius norm, and $\mathrm{diag}(A)$ is the diagonal matrix defined by the main diagonal entries of A. The special case $c_F(A)$ is called the optimal circulant preconditioner [22]. Then it is easy to see that

$$c_F(A) = \sum_{j=0}^{n-1}\left[\frac{1}{n}\sum_{p-q\equiv j (\mathrm{mod}\, n)} a_{pq}\right]Q^j. \quad (5)$$

In particular, when $A = T$ is a Toeplitz matrix, then the entries of circulant preconditioner are given by

$$c_k = [(n-k)t_k + kt_{k-n}]/n, \quad (0 \leqslant k \leqslant n-1).$$

Tyrtyshnikov [24] suggested a so-called superoptimal circulant preconditioner for an arbitrary matrix. We can prove that [24,28]

$$t_U(A) := c_U(AA^\dagger)[c_U(A^\dagger)]^{-1} = \arg\min_{W\in\mathcal{M}_U} \|I_n - W^{-1}A\|_F.$$

The special case $t_F(A)$ is called the superoptimal circulant matrix [24], where the construction of $t_F(T)$ needs $O(n\log n)$ operations.

To examine the efficiency of the circulant preconditioner, we are concerned about the spectra of the preconditioned matrix $C^{-1}A$, where C is a circulant preconditioner. The analysis for general case is difficult. Numerical tests shows that, in most cases, the circulant preconditioner can make the condition number of $C^{-1}A$ small. However, as for the Toeplitz matrix T_n with a positive generating function in the Wiener class, the circulant preconditioner $C = s_F(T), c_F(T)$, or $t_F(T)$ introduced above satisfies that, for all $\epsilon \geqslant 0$, there exist integers M and N such that, for all $n > N$, the matrix $C^{-1}T - I_n$ has at most M eigenvalues larger in absolute value than ϵ [29]. That is, for large n, the spectrum of the preconditioned matrix $C^{-1}T$ is clustered around 1. We can also prove that the smallest eigenvalue of the preconditioned matrix $C^{-1}T$ is uniformly bounded away from the origin. It follows that we can expect the superlinear convergence of the preconditioned CG method.

Although the circulant preconditioner $C = c_F(A)$ given in Eqs. (4) or (5) has an explicit formula, to compute all the entries of C will take about $O(n^2)$ in classical computing and at least $O(n)$ in quantum computing. So direct computation of C will bring no benefits in solving the linear system $C^{-1}Ax = C^{-1}b$. The SVE technique only requires the quantum state of C, and shows great advantages to solve the circulant preconditioned linear system, as we discuss in the next section.

III. PRECONDITIONED LINEAR SYSTEM

In this section, we consider the preconditioning technique for solving the linear system $Ax = b$. The circulant preconditioner C of this linear system can be constructed, for example, by Eq. (4).

Then, the preconditioned linear system reads

$$C^{-1}Ax = C^{-1}b. \quad (6)$$

The method we use to solve the linear system (6) is based on the singular value estimation (SVE) method introduced in Ref. [7]. In Sec. III A, we first introduce the SVE technique with some modifications. Then, in Sec. III B, we show how to solve Eq. (6) based on the modified SVE.

A. Singular value estimation

In Ref. [7], Kerenidis and Prakash introduced a data structure to efficiently store matrices in a quantum computer. Based on this data structure, a fast quantum algorithm to the SVE can be obtained. With this SVE technique, we can perform various scientific calculations with quantum computers, such as an implementation of a dense Hamiltonian simulation [10], solving dense linear system [9], as well as some other applications based on singular value decomposition [11].

In their original paper [7], the authors used the rows of the given matrix A. Taking into account the preconditioning, here we slightly modify it and use the columns instead. Let $A = (A_{ij})_{n\times n}$ be an $n\times n$ matrix. For any $0 \leqslant j \leqslant n-1$, denote $\|A_j\|$ and $|A_j\rangle = \frac{1}{\|A_j\|}\sum_{i=0}^{n-1} A_{ij}|i\rangle$ as the two-norm and the quantum state of the jth column of A, and also define $\|A\|_F = (\sum_j \|A_j\|^2)^{1/2}$ as the Frobenius norm of A and $|A_F\rangle = \frac{1}{\|A\|_F}\sum_{j=0}^{n-1}\|A_j\||j\rangle$. With the similar analysis as Ref. [7], the quantum computer can perform the following mappings in $O(\mathrm{poly}(\log n))$ time:

$$\begin{aligned} U_{\mathcal{M}} &: |0\rangle|j\rangle \mapsto |A_j\rangle|j\rangle = \frac{1}{\|A_j\|}\sum_{i=0}^{n-1} A_{ij}|i,j\rangle, \\ U_{\mathcal{N}} &: |i\rangle|0\rangle \mapsto |i\rangle|A_F\rangle = \frac{1}{\|A\|_F}\sum_{j=0}^{n-1}\|A_j\||i,j\rangle. \end{aligned} \quad (7)$$

Define two degenerate operators $\mathcal{M}$ and $\mathcal{N}$ as

$$\mathcal{M} : |j\rangle \mapsto |A_j\rangle|j\rangle, \quad \mathcal{N} : |i\rangle \mapsto |i\rangle|A_F\rangle.$$

That is,

$$\mathcal{M} = \sum_{j=0}^{n-1}|A_j\rangle|j\rangle\langle j|, \quad \mathcal{N} = \sum_{i=0}^{n-1}|i\rangle|A_F\rangle\langle i|.$$

Then we can verify that

$$\begin{aligned} \mathcal{N}^\dagger\mathcal{M} &= \sum_{i,j=0}^{n-1}|i\rangle\langle i|A_j\rangle\langle A_F|j\rangle\langle j| = \sum_{i,j=0}^{n-1}\frac{A_{ij}}{\|A\|_F}|i\rangle\langle j| \\ &= \frac{A}{\|A\|_F}. \end{aligned}$$

It is also easy to check that $\mathcal{M}^\dagger\mathcal{M} = \mathcal{N}^\dagger\mathcal{N} = I_n$. The following unitary transformation:

$$\begin{aligned} 2\mathcal{M}\mathcal{M}^\dagger - I_{n^2} &= 2\sum_{j=0}^{n-1}|A_j\rangle|j\rangle\langle A_j|\langle j| - I_{n^2} \\ &= U_{\mathcal{M}}\left[2\sum_{j=0}^{n-1}|0\rangle|j\rangle\langle 0|\langle j| - I_{n^2}\right]U_{\mathcal{M}}^\dagger, \end{aligned}$$

can be efficiently implemented in time $O(\mathrm{poly}(\log n))$. Similarly, $2\mathcal{N}\mathcal{N}^\dagger - I_{n^2}$ can be efficiently implemented in time

$O(\text{poly}(\log n))$, too. Now denote

$$W = (2\mathcal{N}\mathcal{N}^\dagger - I_{n^2})(2\mathcal{M}\mathcal{M}^\dagger - I_{n^2}).$$

Let $A = \sum_{i=0}^{n-1} \sigma_i |u_i\rangle\langle v_i|$ be the singular value decomposition of A, then

$$\begin{aligned} W\mathcal{M}|v_i\rangle &= (2\mathcal{N}\mathcal{N}^\dagger - I_{n^2})(2\mathcal{M}\mathcal{M}^\dagger - I_{n^2})\mathcal{M}|v_i\rangle \\ &= (2\mathcal{N}\mathcal{N}^\dagger - I_{n^2})\mathcal{M}|v_i\rangle \\ &= \frac{2}{\|A\|_F}\mathcal{N}A|v_i\rangle - \mathcal{M}|v_i\rangle \\ &= \frac{2\sigma_i}{\|A\|_F}\mathcal{N}|u_i\rangle - \mathcal{M}|v_i\rangle, \end{aligned}$$

and

$$\begin{aligned} W\mathcal{N}|u_i\rangle &= (2\mathcal{N}\mathcal{N}^\dagger - I_{n^2})(2\mathcal{M}\mathcal{M}^\dagger - I_{n^2})\mathcal{N}|u_i\rangle \\ &= (2\mathcal{N}\mathcal{N}^\dagger - I_{n^2})\left(\frac{2}{\|A\|_F}\mathcal{M}A^\dagger|u_i\rangle - \mathcal{N}|u_i\rangle\right) \\ &= (2\mathcal{N}\mathcal{N}^\dagger - I_{n^2})\left(\frac{2\sigma_i}{\|A\|_F}\mathcal{M}|v_i\rangle - \mathcal{N}|u_i\rangle\right) \\ &= \frac{4\sigma_i}{\|A\|_F^2}\mathcal{N}A|v_i\rangle - \frac{2\sigma_i}{\|A\|_F}\mathcal{M}|v_i\rangle - \mathcal{N}|u_i\rangle \\ &= \left(\frac{4\sigma_i^2}{\|A\|_F^2} - 1\right)\mathcal{N}|u_i\rangle - \frac{2\sigma_i}{\|A\|_F}\mathcal{M}|v_i\rangle. \end{aligned}$$

The subspace $\{\mathcal{M}|v_i\rangle, \mathcal{N}|u_i\rangle\}$ is invariant under W. Moreover, W is a rotation in the this subspace. The orthogonal basis of the subspace $\{\mathcal{M}|v_i\rangle, \mathcal{N}|u_i\rangle\}$ is

$$\begin{aligned} |e_{i1}\rangle &= \mathcal{M}|v_i\rangle, \\ |e_{i2}\rangle &= \frac{\mathcal{N}|u_i\rangle - \langle u_i|\mathcal{N}^\dagger\mathcal{M}|v_i\rangle|e_{i1}\rangle}{\|\mathcal{N}|u_i\rangle - \langle u_i|\mathcal{N}^\dagger\mathcal{M}|v_i\rangle|e_{i1}\rangle\|} \\ &= \frac{\mathcal{N}|u_i\rangle - \sigma_i\|A\|_F^{-1}|e_{i1}\rangle}{\sqrt{1-\sigma_i^2\|A\|_F^{-2}}}. \end{aligned}$$

Then

$$W|e_{i1}\rangle = \left[2\frac{\sigma_i^2}{\|A\|_F^2} - 1\right]|e_{i1}\rangle + 2\frac{\sigma_i}{\|A\|_F}\sqrt{1-\frac{\sigma_i^2}{\|A\|_F^2}}|e_{i2}\rangle.$$

Therefore, the eigenvalues $e^{\pm i\theta_i}$ of W satisfies $\cos\theta_i = 2\frac{\sigma_i^2}{\|A\|_F^2} - 1$; that is, $\cos(\theta_i/2) = \sigma_i/\|A\|_F$. The corresponding eigenvectors are $|x_\pm^{(i)}\rangle = \frac{1}{\sqrt{2}}(|e_{i1}\rangle \pm i|e_{i2}\rangle)$.

The vectors $\mathcal{M}|v_i\rangle$ and $\mathcal{N}|u_i\rangle$ can be reformulated by $|x_\pm^{(i)}\rangle$ as follows:

$$\begin{aligned} \mathcal{M}|v_i\rangle &= \sqrt{2}(|x_+^{(i)}\rangle + |x_-^{(i)}\rangle), \\ \mathcal{N}|u_i\rangle &= \sqrt{2}(e^{i\theta_i/2}|x_+^{(i)}\rangle + e^{-i\theta_i/2}|x_-^{(i)}\rangle). \end{aligned}$$

Given any state $|b\rangle = \sum_{i=0}^{n-1}\beta_i|v_i\rangle$, we have

$$U_{\mathcal{M}}|b\rangle = \sum_{i=0}^{n-1}\beta_i\mathcal{M}|v_i\rangle = \sum_{i=0}^{n-1}\sqrt{2}\beta_i(|x_+^{(i)}\rangle + |x_-^{(i)}\rangle).$$

Using the phase estimation algorithm and an oracle for computing $\sigma_i = \|A\|_F\cos(\theta_i/2)$, we have

$$\sum_{i=0}^{n-1}\sqrt{2}\beta_i[|x_+^{(i)}\rangle|\theta_i\rangle + |x_-^{(i)}\rangle|-\theta_i\rangle]|\sigma_i\rangle.$$

Using the phase rotation, the state is transformed into

$$\sum_{i=0}^{n-1}\sqrt{2}\beta_i[e^{i\theta_i/2}|x_+^{(i)}\rangle|\theta_i\rangle + e^{-i\theta_i/2}|x_-^{(i)}\rangle|-\theta_i\rangle]|\sigma_i\rangle.$$

Undo the phase estimation algorithm, we then obtain

$$\sum_{i=0}^{n-1}\sqrt{2}\beta_i(e^{i\theta_i/2}|x_+^{(i)}\rangle + e^{-i\theta_i/2}|x_-^{(i)}\rangle)|\sigma_i\rangle = \sum_{i=0}^{n-1}\beta_i\mathcal{N}|u_i\rangle|\sigma_i\rangle.$$

Finally, applying $U_{\mathcal{N}}^{-1}$, we have the state $\sum\beta_i|u_i\rangle|\sigma_i\rangle$. The procedure is summarized in the following lemma.

Lemma 1. Let A be an $n\times n$ matrix with the singular value decomposition $A = \sum_{i=0}^{n-1}\sigma_i|u_i\rangle\langle v_i|$. Then there is a quantum algorithm that runs in $O(\text{poly}(\log n)/\epsilon)$ and achieves $\sum_i\alpha_i|v_i\rangle|0\rangle \mapsto \sum_i\alpha_i|u_i\rangle|\tilde{\sigma}_i\rangle$, where $|\tilde{\sigma}_i - \sigma_i| \leqslant \epsilon\|A\|_F$ for all i with probability at least $1 - 1/\text{poly}(\log n)$.

In Ref. [7], they achieved $\sum_i\alpha_i|v_i\rangle|0\rangle \mapsto \sum_i\alpha_i|v_i\rangle|\tilde{\sigma}_i\rangle$. However, the result in Lemma 1 is the transformation $\sum_i\alpha_i|v_i\rangle|0\rangle \mapsto \sum_i\alpha_i|u_i\rangle|\tilde{\sigma}_i\rangle$. This procedure is quite suitable to perform matrix multiplication. Similarly, we can efficiently perform the transformation $\sum_i\alpha_i|u_i\rangle|0\rangle \mapsto \sum_i\alpha_i|v_i\rangle|\tilde{\sigma}_i\rangle$, and such a procedure benefits the inverse operation of a matrix.

Usually, when A is non-Hermitian, we need to expand it to a Hermitian matrix

$$\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix},$$

and so $|v_i\rangle = |u_i\rangle$. But for some cases where the matrix is given as quantum information, like the problem considered in this paper, such an expansion is not to easy to be realized. Our method given in Lemma 1 works for a non-Hermitian matrix and does not need such expansion.

Remark 1. Based on the data structure given in Ref. [7], similarly we can obtain

$$|A\rangle = \frac{1}{\|A\|_F}\sum_{i,j=0}^{n-1}A_{ij}|i,j\rangle = \frac{1}{\|A\|_F}\sum_{j=0}^{n-1}\|A_j\||A_j\rangle|j\rangle \tag{8}$$

in time $O(\text{poly}(\log n))$. The SVE in Lemma 1 is realized by using $U_{\mathcal{M}}$ and $U_{\mathcal{N}}$. However, by using the $U_{\mathcal{M}}$ in Eq. (7) and $|A\rangle$ in Eq. (8), we can also construct the SVE. The reason is that if we apply $U_{\mathcal{M}}^{-1}$ on $|A\rangle$, then we get $|A_F\rangle$, equivalently, we obtain $U_{\mathcal{N}}$. This is the main idea that will be used in the next section. We just need to focus on the construction of the quantum states of A and its columns.

For the linear system, we can choose $|b\rangle = \sum_i\beta_i|u_i\rangle$. The solution of the linear system $Ax = b$ can be obtained by Lemma 1 in the following way (a procedure similar to the

HHL algorithm):

$$
\begin{aligned}
&\sum_i \beta_i |u_i\rangle|0\rangle \\
\mapsto &\sum_i \beta_i |v_i\rangle|\tilde{\sigma}_i\rangle|0\rangle \\
\mapsto &\sum_i \beta_i |v_i\rangle|\tilde{\sigma}_i\rangle\big[Z\tilde{\sigma}_i^{-1}|0\rangle + \sqrt{1 - Z^2\tilde{\sigma}_i^{-2}}|1\rangle\big],
\end{aligned}
$$

for some parameter Z. The complexity to get the solution to accuracy ϵ is about $O(\kappa^2 \text{poly}(\log n)\|A\|_F/\epsilon)$. The analysis is the same as Theorem 3 of Ref. [9].

Lemma 2. For any matrix A and quantum state $|b\rangle$, the quantum state of $A^{-1}|b\rangle$ to the accuracy of order ϵ, can be obtained in time $O(\kappa(A)^2\text{poly}(\log n)\|A\|_F/\epsilon)$, where $\kappa(A)$ is the condition number of A.

B. Preconditioned linear solver

To design the quantum linear solver of the linear system (6), we want the SVE of $C^{-1}A$. Such SVE demands the quantum states of columns of $C^{-1}A$ and $C^{-1}A$ itself, which further needs the SVE of C.

We first consider the construction of the preconditioner C in a quantum state. Since $C = F^\dagger \Lambda F$ and F is Fourier transformation, we just need to focus on the diagonal matrix Λ. By Eqs. (3) and (5), the eigenvalues of C or the diagonals of Λ can be expressed by

$$\lambda_k = \frac{1}{n}\sum_{p,q} \omega^{(p-q)k} A_{p,q}. \tag{9}$$

By Eq. (9), we can get a bound about $|\lambda_k|$ if we have enough information of A. So we can perform a suitable scaling such that the singular values $|\lambda_k|$ of C are smaller than 1 and larger than $1/\kappa(C)$. Therefore, we assume that $\max_k |\lambda_k| = 1$ and $\min_k |\lambda_k| = 1/\kappa(C)$.

In the following we form the state $|\lambda\rangle = \frac{1}{\|C\|_F}\sum_{k=0}^{n-1}\lambda_k|k\rangle$, where $\|C\|_F = (\sum_{k=0}^{n-1}\lambda_k^2)^{1/2}$. From the quantum state of $|A\rangle$, we can get

$$
\begin{aligned}
&\frac{1}{\|A\|_F}\sum_{p,q=0}^{n-1} A_{p,q}|p,q\rangle \\
\mapsto &\frac{1}{n\|A\|_F}\sum_{p,q,u,v=0}^{n-1} A_{p,q}\omega^{pu-qv}|u,v\rangle|u-v\rangle \\
= &\frac{1}{n\|A\|_F}\sum_{p,q,k=0}^{n-1} A_{p,q}\omega^{(p-q)k}|k,k\rangle|0\rangle + |0\rangle^\perp \\
= &\frac{1}{\|A\|_F}\sum_{k=0}^{n-1}\lambda_k|k,k\rangle|0\rangle + |0\rangle^\perp \\
\mapsto &\frac{1}{\|A\|_F}\sum_{k=0}^{n-1}\lambda_k|k\rangle|0\rangle|0\rangle + |0,0\rangle^\perp.
\end{aligned}
$$

The probability to get $|\lambda\rangle$ is $\|C\|_F/\|A\|_F$. Performing measurements, we can get the state $|\lambda\rangle$ in time

$$O(\|A\|_F\text{poly}(\log n)/\|C\|_F) = \widetilde{O}(\|A\|_F/\|C\|_F). \tag{10}$$

Therefore, $U_{\mathcal{N}}$ for Λ can be implemented in time $\widetilde{O}(\|A\|_F/\|C\|_F)$, while $U_{\mathcal{M}}$ for Λ is trivial. Thus we have the SVE of Λ, so equivalently the SVE of C. Note that Λ is diagonal, so the SVD of Λ is completely trivial if we know its diagonals explicitly. However, a direct calculation according to the formula (9) will cost at least $O(n^2)$ to get Λ. In the quantum procedure above, we use a different method to construct the quantum state of the diagonal of Λ within the time complexity as given in Eq. (10).

Next, we consider how to form the quantum state $|C^{-1}A\rangle$. The basic idea is computing the inverse of C based on its SVE. As shown in the HHL algorithm, such a procedure depends on the condition number of C. By Lemma III A, the quantum state $|C^{-1}A_j\rangle$ of the jth column of $C^{-1}A$, which is proportional to $C^{-1}|A_j\rangle$, can be prepared in time

$$\widetilde{O}(\kappa(C)^2\|C\|_F\|A\|_F/\|C\|_F\epsilon) = \widetilde{O}(\|A\|_F\kappa(C)^2/\epsilon). \tag{11}$$

This is the complexity to generate $U_{\mathcal{M}}$ for $C^{-1}A$.

Note that the quantum state of $|C^{-1}A\rangle$ equals

$$|C^{-1}A\rangle = \frac{1}{\|C^{-1}A\|_F}\sum_{j=0}^{n-1}\|A_j\|\|C^{-1}|A_j\rangle\||C^{-1}A_j\rangle|j\rangle.$$

Due to the parallelism of the quantum computer, $|C^{-1}A\rangle$ can also be obtained in time (11). The error of obtaining $|C^{-1}A_j\rangle$ is bounded by ϵ; however, the error of $|C^{-1}A\rangle$ will be enlarged by the summation. To analyze this error, we need the following lemma to check the accuracy of the generated state $|\phi\rangle$, compared with the exact one $|\psi\rangle$.

Lemma 3. Assume that

$$|\phi\rangle = \frac{1}{\sqrt{Z}}\sum_{j=0}^{n-1} a_j\mathbf{u}_j, \quad |\psi\rangle = \frac{1}{\sqrt{W}}\sum_{j=0}^{n-1} b_j\mathbf{v}_j,$$

where $\{\mathbf{u}_j : j = 0, \dots, n-1\}$ and $\{\mathbf{v}_j : j = 0, \dots, n-1\}$ are orthogonal basses, not necessarily unit. We assume that $|a_j - b_j| \leqslant \eta_0$, $\|\mathbf{u}_j - \mathbf{v}_j\| \leqslant \eta_1$ for all j, $|Z - W| \leqslant \eta_2$, $\max_j \|\mathbf{v}_j\| = \eta_3$, and $1/\min_j \|\mathbf{u}_j\| = \eta_4$. Then the error estimate reads

$$\||\phi\rangle - |\psi\rangle\|^2 \leqslant 3\eta_1^2\eta_4^2 + \frac{3\eta_2^2\eta_3^2\eta_4^2}{W^2} + \frac{3n\eta_0^2\eta_3^2}{W}. \tag{12}$$

Proof. The estimate of the error bound between $|\phi\rangle$ and $|\psi\rangle$ can be derived as follows:

$$
\begin{aligned}
\||\phi\rangle - |\psi\rangle\|^2 &= \frac{1}{ZW}\sum_{j=0}^{n-1}\big\|\sqrt{W}a_j\mathbf{u}_j - \sqrt{Z}b_j\mathbf{v}_j\big\|^2 \\
&\leqslant \frac{3}{ZW}\sum_{j=0}^{n-1}\big[W|a_j|^2\|\mathbf{u}_j - \mathbf{v}_j\|^2 + Z|a_j - b_j|^2 \\
&\quad \times \|\mathbf{v}_j\|^2 + |\sqrt{W} - \sqrt{Z}|^2|a_j|^2\|\mathbf{v}_j\|^2\big] \\
&\leqslant \frac{3}{ZW}\sum_{j=0}^{n-1}\big[W|a_j|^2\eta_1^2 + |\sqrt{W} - \sqrt{Z}|^2
\end{aligned}
$$

$$\times |a_j|^2\eta_3^2 + Z\eta_0^2\eta_3^2]$$
$$= \frac{3\eta_1^2}{Z}\sum_{j=0}^{n-1}|a_j|^2 + \frac{3|W-Z|^2\eta_3^2}{ZW(\sqrt{W}+\sqrt{Z})^2}$$
$$\times \sum_{j=0}^{n-1}|a_j|^2 + \frac{3n\eta_0^2\eta_3^2}{W}.$$

By using the fact that $\sum_j |a_j|^2/Z \leqslant \eta_4^2$, we then obtain the estimate (14). ■

To estimate the error in generating the state $|C^{-1}A\rangle$, we need to estimate the errors in states $|C^{-1}A_j\rangle$ and the norms $\|C^{-1}|A_j\rangle\|$ and $\|C^{-1}A\|_F$, due to Lemma 3. If we set $|A_j\rangle = \sum_k \alpha_{jk}F^\dagger|k\rangle$, then $C^{-1}|A_j\rangle = \sum_k \alpha_{jk}\lambda_k^{-1}F^\dagger|k\rangle$. So

$$1 = 1/\max_k|\lambda_k| \leqslant \|C^{-1}|A_j\rangle\| \leqslant 1/\min_k|\lambda_k| = \kappa(C). \quad (13)$$

(i) By Lemma 2, we obtain an approximation of $C^{-1}|A_j\rangle$ in the form $\Phi_j = \sum_k \alpha_{jk}\tilde{\lambda}_k^{-1}F^\dagger|k\rangle$, where $|\tilde{\lambda}_k^{-1} - \lambda_k^{-1}| \leqslant \epsilon$. Thus,

$$\|C^{-1}|A_j\rangle - \Phi_j\| = \sqrt{\sum_k |\alpha_{jk}|^2\big|\tilde{\lambda}_k^{-1} - \lambda_k^{-1}\big|^2} \leqslant \epsilon, \quad (14)$$

since $\sum_k |\alpha_{jk}|^2 = 1$.

(ii) Since $\|C^{-1}|A_j\rangle\|^2 = \sum_k |\alpha_{jk}\lambda_k^{-1}|^2$ and $\|\Phi_j\|^2 = \sum_k |\alpha_{jk}\tilde{\lambda}_k^{-1}|^2$, we have $|\|C^{-1}|A_j\rangle\|^2 - \|\Phi_j\|^2| \leqslant \sum_k |\alpha_{jk}|^2|\lambda_k^{-2} - \tilde{\lambda}_k^{-2}| = \sum_k |\alpha_{jk}|^2|\lambda_k^{-1} - \tilde{\lambda}_k^{-1}||\lambda_k^{-1} + \tilde{\lambda}_k^{-1}| \leqslant 2\epsilon/\min_k|\lambda_k| = 2\kappa(C)\epsilon$.

(iii) Since $\|A_j\|$ is given in advance, it contains no error. Note that $\|C^{-1}A\|_F^2 = \sum_j \|A_j\|^2\|C^{-1}|A_j\rangle\|^2$, therefore we have

$$\Big|\|C^{-1}A\|_F^2 - \sum_j \|A_j\|^2\|\Phi_j\|^2\Big|$$
$$= \left|\sum_j \|A_j\|^2(\|C^{-1}|A_j\rangle\|^2 - \|\Phi_j\|^2)\right|$$
$$\leqslant 2\|A\|_F^2\kappa(C)\epsilon. \quad (15)$$

Applying Lemma 3 with the parameters

$\eta_0 = 0$,	since no error in $\|A_j\|$,
$\eta_1 = \epsilon$,	due to Eq. (14),
$\eta_2 = 2\|A\|_F^2\kappa(C)\epsilon$,	due to Eq. (15),
$\eta_3 = 1/\min_k|\lambda_k| = \kappa(C)$,	due to Eq. (13),
$\eta_4 \leqslant (1-\epsilon)^{-1} \approx 1$,	due to Eqs. (13) and (14),

the error for obtaining $|C^{-1}A\rangle$ is bounded by

$$3\epsilon^2 + 12\kappa^4(C)\epsilon^2\frac{\|A\|_F^4}{\|C^{-1}A\|_F^4}. \quad (16)$$

Denote $\|C^{-1}A\|_F = \beta\|A\|_F$, then Eq. (16) can be simplified into

$$3\epsilon^2 + 12\epsilon^2\kappa^4(C)/\beta^4. \quad (17)$$

Since $\frac{\|A\|_F}{\|C\|} \leqslant \|C^{-1}A\|_F \leqslant \|C^{-1}\|\|A\|_F$, we have

$$1 = \frac{1}{\|C\|} \leqslant \beta \leqslant \|C^{-1}\| = \kappa(C). \quad (18)$$

To keep the error (17) bounded by size ϵ_0^2, we should choose ϵ such that $\epsilon^2\kappa(C)^4 = \epsilon_0^2\beta^4$, i.e., $\epsilon = \epsilon_0\beta^2/\kappa(C)^2$. By Lemma 2 and Eq. (10), the complexity to get the quantum state of $C^{-1}A$ is

$$\widetilde{O}\left(\frac{\|A\|_F}{\|C\|_F}\frac{\kappa(C)^2\|C\|_F}{\epsilon}\right) = \widetilde{O}\left(\frac{\|A\|_F\kappa(C)^4}{\epsilon_0\beta^2}\right)$$
$$= \widetilde{O}\left(\frac{\kappa(C)^4\|A\|_F^3}{\epsilon_0\|C^{-1}A\|_F^2}\right). \quad (19)$$

This is the complexity to generate $U_\mathcal{N}$ for the SVE of $C^{-1}A$. In Eq. (11), if we change ϵ into ϵ_0, then it becomes $\widetilde{O}(\|A\|_F\kappa(C)^4/\epsilon_0\beta^2)$, which is the same as Eq. (19).

Finally, by Lemma 2, the complexity of the quantum algorithm to solve $Ax = b$ based on the circulant preconditioner and SVE is summarized as follows:

Theorem 1. The quantum state of the solution of $Ax = b$ by using the preconditioner C to accuracy ϵ can be obtained in time

$$\widetilde{O}\big(\kappa(C)^4\kappa(C^{-1}A)^2\|A\|_F^2/\epsilon^2\big). \quad (20)$$

Proof. By formulas (11) and (19), it costs $\widetilde{O}(\kappa(C)^4\|A\|_F^3/\epsilon\|C^{-1}A\|_F^2)$ to generate $U_\mathcal{M}$ and $U_\mathcal{N}$ for the SVE of $C^{-1}A$. By Lemma 2, the linear solver costs $\widetilde{O}(\|C^{-1}A\|_F\kappa(C^{-1}A)^2/\epsilon)$. Therefore, the complexity to solve the linear system $C^{-1}Ax = C^{-1}b$ is

$$\widetilde{O}\left(\frac{\kappa(C)^4\|A\|_F^3}{\epsilon\|C^{-1}A\|_F^2}\frac{\|C^{-1}A\|_F\kappa(C^{-1}A)^2}{\epsilon}\right)$$
$$= \widetilde{O}\big(\kappa(C)^4\kappa(C^{-1}A)^2\|A\|_F^2/\epsilon^2\big). \quad ■$$

Generally, it is not easy to compare the complexity given in Theorem 1 with the HHL algorithm and its variants, as well as the quantum algorithm given in Ref. [9]. Table I gives a list of already known quantum algorithms to solve linear systems.

Remark 2. For a good preconditioner, it is reasonable to assume that $\kappa(C) \ll \kappa(A)$ and $\kappa(C^{-1}A) \ll \kappa(A)$. That is, $\kappa(C)$ and $\kappa(C^{-1}A)$ can be assumed to be of $O(1)$. Under these conditions, the complexity can be further simplified to $\widetilde{O}(\|A\|_F^2/\epsilon^2)$.

TABLE I. Comparison of quantum algorithms to solve linear system $Ax = b$, where $s(A)$ is the sparsity of A and $\kappa(A)$ is the condition number of A. Matrices M and C are the SPAI and circulant preconditioner of A, respectively. The first four quantum algorithms are suitable to deal with linear systems with small condition number. The last two can be used to solve linear systems with large condition number.

Quantum algorithm	Complexity	Requirement
HHL [4]	$\widetilde{O}(s(A)\kappa(A)^2/\epsilon)$	Sparse
Ambainis [5]	$\widetilde{O}(s(A)\kappa(A)/\epsilon^3)$	Sparse
CKS [6]	$\widetilde{O}(s(A)\kappa(A)(\log 1/\epsilon)^{4.5})$	Sparse
WZP [9]	$\widetilde{O}(\kappa(A)^2\|A\|_F/\epsilon)$	None
CJS [21]	$\widetilde{O}(s(A)^7\kappa(MA)/\epsilon)$	SPAI
Theorem 1	$\widetilde{O}(\kappa(C)^4\kappa(C^{-1}A)^2\|A\|_F^2/\epsilon^2)$	None

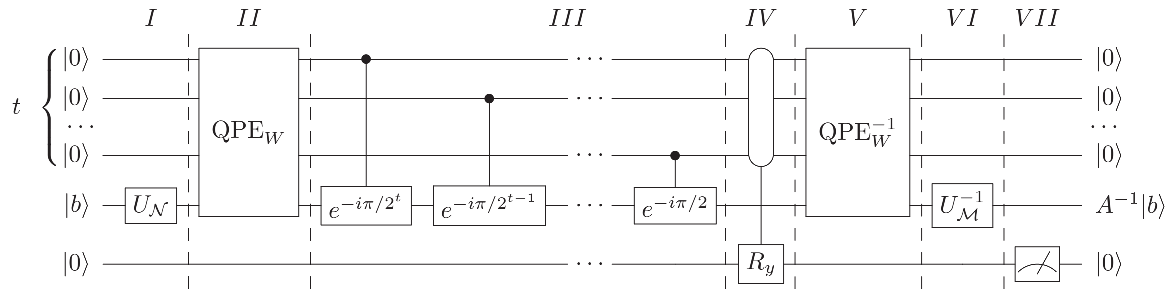

FIG. 1. The circuit implementation of applying the matrix inverse by SVE.

The above method for the circulant preconditioner C can actually be extended to general cases. We consider a general preconditioner M. The preconditioned linear system reads $M^{-1}Ax = M^{-1}b$. Assume that the matrices A and M are stored in a quantum state, for example, via qRAM. Since we do not need to construct the preconditioner M in a quantum state, as we do for the preconditioner C, the complexity term $\|A\|_F/\|M\|_F$ associated preconditioner construction similar to that in Eq. (10) disappears. With the same analysis as Theorem 1, we have the following theorem:

Theorem 2. Given the matrices A and M are stored in a quantum state, then the total time complexity for solving $M^{-1}Ax = M^{-1}b$ to accuracy ϵ in quantum computer is $\widetilde{O}(\kappa(M)^4\kappa(M^{-1}A)^2\|A\|_F\|M\|_F/\epsilon^2)$.

C. Circuit implementations

Since W has the following unitary decomposition:

$$W = U_{\mathcal{N}}\left[2\sum_{i=0}^{n-1}|i\rangle|0\rangle\langle i|\langle 0| - I_{n^2}\right]U_{\mathcal{N}}^{\dagger}$$
$$\times U_{\mathcal{M}}\left[2\sum_{j=0}^{n-1}|0\rangle|j\rangle\langle 0|\langle j| - I_{n^2}\right]U_{\mathcal{M}}^{\dagger},$$

the circuit of implementing W is totally determined by the circuits of implementing $U_{\mathcal{M}}$ and $U_{\mathcal{N}}$. In the following, we assume that the circuits to implement $U_{\mathcal{M}}$ and $U_{\mathcal{N}}$ are known [7].

Denote $t = \lceil\log(\mathrm{poly}\log n)/\epsilon\rceil$, where $\mathrm{poly}\log n$ is determined in advance. If we are not concerned too much about the success probability in Lemma 1, then we can set it as a constant and so $t = \lceil\log 1/\epsilon\rceil$. Denote the SVD by $A = \sum\sigma_i|u_i\rangle\langle v_i|$, and let $|b\rangle = \sum\beta_i|u_i\rangle$ be an initial state. Then $A^{-1}|b\rangle = \sum\sigma_i^{-1}\beta_i|v_i\rangle$. The SVE procedure is used to achieve $\sum\beta_i|u_i\rangle \mapsto \sum\beta_i|v_i\rangle|\tilde{\sigma}_i\rangle$. The circuit implementation of applying SVE to solve linear equations is described as in Fig. 1.

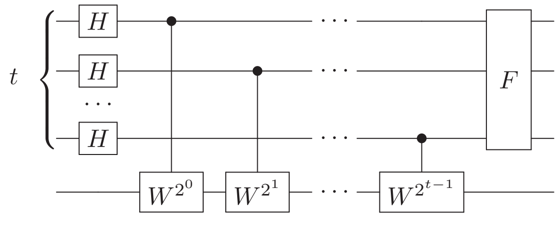

FIG. 2. Circuit implementation of QPE_W, the operator of quantum phase estimation for W.

Stage I is the first step of SVE; it obtains the initial state $U_{\mathcal{N}}|b\rangle$ by applying $U_{\mathcal{N}}$ to $|b\rangle$.

Stage II is the standard quantum phase estimation algorithm for W, denoted by QPE_W; see Fig. 2.

Stage III inserts a relative phase $e^{\mp i\theta_i/2}$ into the eigenvector $|x_{\pm}^{(i)}\rangle$. For each i, the output of quantum phase estimation is $y_i = y_{i0} + y_{i1}2 + \cdots + y_{i,t-1}2^{t-1}$, such that $|2\pi y_i/2^t - \theta_i| \leqslant \epsilon$, where $y_{i0},\ldots,y_{i,t-1} \in \{0,1\}$. Hence, $e^{i\theta_i/2} \approx e^{i\pi y_i/2^t} = e^{i\pi y_{i0}/2^t}e^{i\pi y_{i1}/2^{t-1}}\cdots e^{i\pi y_{i,t-1}/2}$, where the item $e^{i\pi y_{ij}/2^{t-j}}$ gives a nontrivial phase if $y_{ij} = 1$.

Stage IV is the controlled rotation that maps $|0\rangle$ to $Z\sigma_i^{-1}|0\rangle + (1 - Z^2\sigma_i^{-2})^{1/2}|1\rangle$, where $\sigma_i = \|A\|_F\cos(\theta_i/2)$ and $Z = \min\sigma_i$. If $Z = \|A\|_F\cos(\theta_{i_0}/2)$, then the rotation angle ϕ_i satisfies $\cos\phi_i = \cos(\theta_{i_0}/2)/\cos(\theta_i/2)$. That is $\phi_i = \arccos[\cos(\theta_{i_0}/2)/\cos(\theta_i/2)] \approx 2\pi z_i/2^t$ for some integer z_i. Denote $z_i = z_{i0} + z_{i1}2 + \cdots + z_{i,t-1}2^{t-1}$, where $z_{i0},\ldots,z_{i,t-1} \in \{0,1\}$, then the controlled rotation can be achieved in a similar way to stage III as shown in Fig. 3, where

$$R_y(a) := \begin{pmatrix}\cos a & -\sin a\\ \sin a & \cos a\end{pmatrix}$$

for any a.

Stage V implements QPE_W^{-1}, undoing the quantum phase estimation.

Stage VI applies $U_{\mathcal{M}^{-1}}$ to the achieved state to get $|0\rangle^{\otimes t}\sum\beta_i|v_i\rangle[Z\sigma_i^{-1}|0\rangle + (1 - Z^2\sigma_i^{-2})^{1/2}|1\rangle]$.

State VII performs a measurement on the last qubit. If we get $|0\rangle$, then the remaining state is $A^{-1}|b\rangle$.

For solving the linear system $C^{-1}Ax = Cd^{-1}b$, we need the SVE of $C^{-1}A$. Furthermore, during the procedure for $C^{-1}A$, the $U_{\mathcal{M}}$ to achieve $|C^{-1}A\rangle$ and the $U_{\mathcal{N}}$ to achieve $|C^{-1}A_j\rangle$ are based on the SVE of C. When we have the SVE of C, then we can apply the circuit in Fig. 1 to realize the

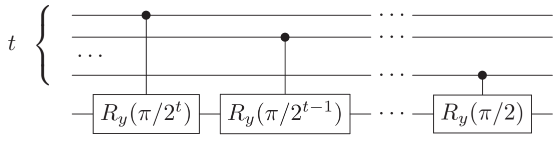

FIG. 3. Circuit implementation of the controlled rotation R_y.

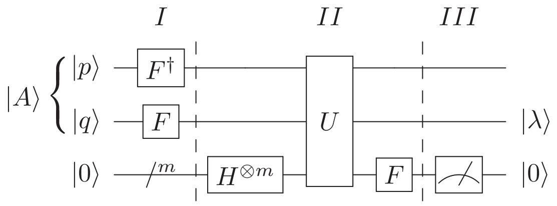

FIG. 4. The circuit implementation of $U_\mathcal{N}$ for Λ.

circuit of $U_\mathcal{M}$ and $U_\mathcal{N}$ for $C^{-1}A$. Hence we need to construct the circuit to implement $U_\mathcal{M}, U_\mathcal{N}$ of C. Since $U_\mathcal{M}$ for Λ is trivial, it suffices to build the circuit of $U_\mathcal{N}$ for Λ.

Define $U|u,v,r\rangle = e^{2\pi i(u-v)r/2^m}|u,v,r\rangle$, where $m = \lceil \log n \rceil$. Then the circuit of $U_\mathcal{N}$ for Λ is as follows:

We can see that stage II aims at preparing $|u,v\rangle|u-v\rangle$ from $|u,v\rangle|0\rangle$.

$$\begin{aligned}
|u,v\rangle|0\rangle \xrightarrow{I\otimes I\otimes H^{\otimes m}} & \frac{1}{\sqrt{2^m}}\sum_{r=0}^{2^m-1}|u,v\rangle|r\rangle,\\
\overset{U}{\to} \quad & \frac{1}{\sqrt{2^m}}\sum_{s=0}^{2^m-1} e^{2\pi i(u-v)r/2^m}|u,v\rangle|r\rangle,\\
\xrightarrow{I\otimes I\otimes F} \quad & \frac{1}{2^m}\sum_{s=0}^{2^m-1}\sum_{r=0}^{2^m-1} e^{2\pi i[(u-v)r-rs]/2^m}|u,v\rangle|s\rangle\\
= \quad & |u,v\rangle|u-v\rangle.
\end{aligned}$$

We would like to say more about the implementation of U in Fig. 4. Denote $u = \sum_{j=0}^{m-1} u_j 2^j$, and $r = \sum_{k=0}^{m-1} r_k 2^k$, then

$$2\pi ur/2^m = \sum_{j,k=0}^{m-1} u_j r_k \pi/2^{m-j-k-1}.$$

Adding a new qubit $|0\rangle$ into $|u\rangle|r\rangle$, we have $|u\rangle|r\rangle|0\rangle = |u_0,\ldots,u_{m-1}\rangle|r_0,\ldots,r_{m-1}\rangle|0\rangle$. For each $|u_j\rangle|r_k\rangle$, insert a phase $e^{\pi i/2^{m-j-k-1}}$ into $|0\rangle$ if $u_j = r_k = 1$. The circuit of this transformation is given in Fig. 5.

The composition of all the above circuits for $j,k = 0,\ldots,m-1$ leads to the phase $e^{2\pi i ur/2^m}$ in $|0\rangle$. We can construct similar circuit to obtain the phase $e^{-2\pi i vr/2^m}$. Then the circuit of U is the composition of the above two circuits. It contains $2m^2 = 2(\log n)^2$ elementary gates.

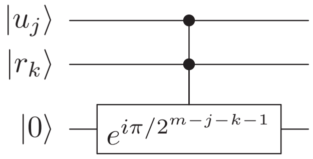

FIG. 5. Component of the circuit implementation of U.

The analysis above fulfills the circuit implementation of $U_\mathcal{N}$ for Λ. Since $U_\mathcal{M}$ for Λ is trivial, we can get $|C^{-1}A\rangle$ and $C^{-1}|A_j\rangle$ from the SVE of Λ, and the circuit is the same as in Fig. 1. Once we get $|C^{-1}A\rangle$ and $C^{-1}|A_j\rangle$, then we have $U_\mathcal{M}$ and $U_\mathcal{N}$ for $C^{-1}A$. And finally we can apply SVE (see Fig. 1) again to solve the linear system $C^{-1}Ax = C^{-1}b$.

IV. CONCLUSION

In this paper, we present a new quantum algorithm based on circulant preconditioning technique to solve general linear systems, especially the dense cases with large condition numbers. The main technique we applied here is the modified version of SVE (Lemma 1). This modified SVE will be more suitable to deal with the cases where we are given quantum inputs, and will have many other applications. However, the new quantum algorithm to solve linear system (Theorem 1) depends on the Frobenius norm of the input matrix. As proved in Ref. [4], unless BQP = PSPACE, the condition number in the time complexity of solving linear systems cannot be removed, so for general case, we cannot expect that $O(\kappa(C)^4\kappa(C^{-1}A)^2\|A\|_F^2)$ is small of size $O(\text{poly}\log n)$ all the time. But a problem in how to improve the dependence of the complexity on $\|A\|_F$, since the result of Ref. [9] is linear in $\|A\|_F$. Also, as suggested by the work of Childs *et al.* [6], it may possible to improve the dependence on precision ϵ to polynomial of $\log 1/\epsilon$.

ACKNOWLEDGMENTS

H.X. is supported by the Natural Science Foundation of China under Grants No. 11571265 and No 11471253 and NSFC-RGC No. 11661161017. C.S. is supported by NSFC Project 11671388 and CAS Project QYZDJ-SSW-SYS022.

[1] J. W. Demmel, *Applied Numerical Linear Algebra* (SIAM, Philadelphia, 1997).

[2] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 4th ed. (The John Hopkins University Press, Baltimore, 2013).

[3] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (SIAM, Philadelphia, 2003).

[4] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[5] A. Ambainis, in *Proc. 29th International Symposium on Theoretical Aspects of Computer Science, Leibniz International Proceedings in Informatics*, edited by C. Dürr and T. Wilke (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, 2012), Vol 14, pp. 636–647.

[6] A. M. Childs, R. Kothari, and R. D. Somma, SIAM J. Comput. **46**, 1920 (2017).

[7] I. Kerenidis and A. Prakash, in *8th Innovations in Theoretical Computer Science Conference*, *Leibniz International Proceedings in Informatics*, edited by C. H. Papadimitriou (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, 2017), Vol. 67, pp. 49:1–49:21.

[8] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).

[9] L. Wossnig, Z. K. Zhao, and A. Prakash, Phys. Rev. Lett. **120**, 050502 (2018).

[10] C. H. Wang and L. Wossnig, arXiv:1803.08273v1.

[11] I. Kerenidis and A. Prakash, arXiv:1704.04992v3.

[12] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. **109**, 050505 (2012).

[13] A. Montanaro and S. Pallister, Phys. Rev. A **93**, 032324 (2016).

[14] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv:1307.0411.

[15] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).

[16] P. Rebentrost, T. R. Bromley, C. Weedbrook, and S. Lloyd, Phys. Rev. A **98**, 042308 (2018).

[17] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakic, A. Aspuru-Guzik, and P. Walther, Sci. Rep. **4**, 6115 (2014).

[18] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, M. Gu, M.-J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Phys. Rev. Lett. **110**, 230501 (2013).

[19] J. Pan, Y. D. Cao, X. Yao, Z. K. Li, C. Y. Ju, H. Chen, X. H. Peng, S. Kais, and J. F. Du, Phys. Rev. A **89**, 022313 (2014).

[20] Y. R. Zheng, C. Song, M. C. Chen, B. X. Xia, W. X. Liu, Q. J. Guo, L. B. Zhang, D. Xu, H. Deng, K. Q. Huang, Y. Wu, Z. G. Yan, D. N. Zheng, L. Lu, J. W. Pan, H. Wang, C. Y. Lu, and X. B. Zhu, Phys. Rev. Lett. **118**, 210504 (2017).

[21] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, Phys. Rev. Lett. **110**, 250504 (2013).

[22] T. F. Chan, SIAM J. Sci. Stat. Comput. **9**, 766 (1988).

[23] G. Strang, Stud. Appl. Math. **74**, 171 (1986).

[24] E. E. Tyrtyshnikov, SIAM J. Matrix Anal. Appl. **13**, 459 (1992).

[25] S. S. Zhou and J. B. Wang, R. Soc. Open Sci. **4**, 160906 (2017).

[26] L. Wan, C. Yu, S. Pan, F. Gao, Q. Wen, and S. Qin, Phys. Rev. A **97**, 062322 (2018).

[27] R. H. Chan, SIAM J. Matrix Anal. Appl. **10**, 542 (1989).

[28] R. H. Chan, X. Q. Jin, and M. C. Yeung, SIAM J. Numer. Anal. **28**, 871 (1991).

[29] R. H. Chan and X. Q. Jin, *An Introduction to Iterative Toeplitz Solvers* (SIAM, Philadelphia, 2007).