COMPUTING EIGENVALUES OF MATRICES IN A QUANTUM COMPUTER

CHANGPENG SHAO

Abstract. Eigenproblem arises in a large number of disciplines of sciences and engineering. Combining quantum algorithms to solve linear differential equations, quantum singular value estimation and quantum phase estimation, we propose quantum algorithms to compute the eigenvalues of diagonalizable matrices of two types: matrices that only have real eigenvalues and normal matrices. The quantum algorithms return a quantum state such that the first register stores eigenvalues and the second register stores the corresponding eigenvectors. For normal matrices, the complexity to obtain this quantum state is dominated by the complexity to perform quantum singular value estimation. For the other case, the complexity is determined by the solving of a linear system of differential equations. Under certain conditions (e.g. sparse), the complexities are polylog on the size $n$ of the given matrix. If we perform measurements, then we can obtain all the eigenvalues classically. If $M$ is $s$ sparse, then the complexity is $\tilde{O}(sn\|M\|_{\infty}\kappa^2/\epsilon^2)$, where $\kappa$ is the condition number of the matrix of eigenvectors.

Key words. quantum algorithm, quantum phase estimation, eigenvalues, normal matrices.

1. Introduction. One important problem in numerical linear algebra is to find efficient and stable algorithms to compute the eigenvalues of matrices. The numerical computations of eigenvalues is a problem of major importance in many scientific and engineering applications [39]. Many classical algorithms (e.g. QR algorithm, Lanczos algorithm and Arnoldi iteration etc.) were discovered [23,39]. Except some special cases, the costs of these algorithms to estimate the eigenvalues of $n \times n$ matrices range from $O(n^2)$ to $O(n^3)$. On the other hand, recent developments on quantum algorithms [1,6,7,11,14,15,27,32] prove the fact that quantum computer is good at solving many linear algebraic problems. For instance, under certain assumptions we can efficiently estimate the eigenvalues of unitary matrices [33] and Hermitian matrices [1], the singular values of general matrices [11,32]. For the general eigenvalue problem, it is shown in [19,45] that if the eigenvector is given, then we can estimate the corresponding eigenvalue efficiently in a quantum computer. To the best of our knowledge these are the only two works in computing eigenvalues of nonunitary (and non-Hermitian) matrices. One difficulty behind this is the results in complexity theory [29,46], which suggesting that many eigenvalue problems are QMA-complete, i.e., they are hard to solve even for quantum computers.

The classical eigenvalue algorithms are mainly iterative. Based on quantum linear algebraic techniques, we can generalize them into quantum algorithms. However, in this paper, we choose to follow the idea of quantum phase estimation (QPE), a method that is totally different from any classical algorithm to estimate eigenvalues. We propose quantum algorithms to estimate the eigenvalues of diagonalizable matrices of two types: normal matrices and matrices whose eigenvalues are all real. The goal of the assumption of diagonalizability is to estimate the eigenvalues without the need to know the eigenvectors in advance. This is similar to the idea of QPE, which is the main technique to estimate the eigenvalues of unitary and Hermitian matrices. We shall prove that in these two types of matrices, we can obtain similar results to QPE. More precisely, if $M$ is an $n \times n$ diagonalizable matrix that only has real eigenvalues $\{\lambda_1,\ldots,\lambda_n\}$. Assume that the corresponding unit eigenvectors are $\{|E_1\rangle,\ldots,|E_n\rangle\}$, then we can implement the following transformation

$$\sum_{j=1}^{n} \beta_j |E_j\rangle \mapsto \sum_{j=1}^{n} \beta_j |\tilde{\lambda}_j\rangle |E_j\rangle$$

---

*School of Mathematics, University of Bristol, Bristol, UK (changpeng.shao@bristol.ac.uk).
up to a normalization, where $\tilde{\lambda}_j$ is an approximation of $\lambda_j$. If $M$ is normal, we write $\lambda_j = \sigma_j e^{i\theta_j}$, then we can implement
\begin{equation}
\sum_{j=1}^{n} \beta_j |E_j\rangle \mapsto \sum_{j=1}^{n} \beta_j |\tilde{\sigma}_j\rangle |\tilde{\theta}_j\rangle
\end{equation}
in a quantum computer, where $\tilde{\sigma}_j$, $\tilde{\theta}_j$ are respectively the approximations of $\sigma_j$ and $\theta_j$. Diagonalizability implies that the left hand side of (1.1), (1.2) can be any quantum state.

The main technique we use are QPE, quantum singular value estimation and quantum algorithms to solve linear differential equations. These are important techniques in designing quantum machine learning algorithms [8, 12, 13, 17, 30, 34, 38, 40–42, 47]. Under certain conditions (e.g. sparse, existence of block-encoding or QRAM), these algorithms are implemented in polylog time in a quantum computer, which help us solving many machine learning problems “exponentially” faster than the classical algorithms. As a result, the quantum algorithms proposed in this paper may efficiently help us estimate the eigenvalues of diagonalizable matrices.

If we perform measurements on the right hand side of equations (1.1) and (1.2), we can obtain all the eigenvalues of $M$ classically. For the matrices considered in this paper, generally classical algorithms [9, 22, 24, 28] still cost $O(n^w)$ to compute their eigenvalues, where $2 \leq w \leq 3$. However, in the quantum case, let $s$ be the sparsity of the given matrix $M$, then it costs $O(sn\|M\|_{\max}\kappa^2/\epsilon^2)$ to build the sparse access oracles. Consequently, we can obtain all the eigenvalues in cost $\tilde{O}(sn\|M\|_{\max}\kappa^2/\epsilon^2)$ if $M$ is $s$ sparse, see Theorems 2.10 and 3.3 for more details. Here $\|M\|_{\max}$ is the max norm and $\kappa$ is the condition number of the matrix of the eigenvectors. For normal matrices $\kappa = 1$.

One potential application of quantum eigenvalue estimation that quantum singular value estimation may not perform well is to implement functions of matrices in a quantum computer. For Hermitian matrices, singular value decomposition is enough [44]. Generally, we can only use eigenvalue decomposition. If $M = EDE^{-1}$ is the eigenvalue decomposition of $M$, and $f$ is a function, then usually $f(M) = Ef(D)E^{-1}$. This is correct if $f$ is a polynomial and many other functions, like exponential functions and trigonometric functions. When we have the transformation (1.1) or (1.2), then we can implement $f(M)$ efficiently by using control rotations generated by the eigenvalues. Note that if $f$ is a polynomial, then based on the block-encoding [11] and linear combinations of unitaries [16], we can also implement $f(M)$ by the quantum singular value estimation. However, if $f$ is not a polynomial, such as $f(x) = e^x, \sqrt{x}$ or $\log x$, then quantum eigenvalue estimation technique is more preferable.

The paper is outlined as follows: In Section 2, we apply quantum linear differential equation solver to propose a quantum algorithm to estimate the eigenvalues of matrices that only have real eigenvalues. In Section 3, we propose the quantum algorithm to estimate the eigenvalues of normal matrices based on quantum SVD and QPE. For the readers that are nor familiar with QPE, we refer to Appendix A for a brief review.

2. Diagonalizable matrices only have real eigenvalues. In this section, we consider the problem of computing eigenvalues of diagonalizable matrices that only have real eigenvalues. Assume that $M$ is an $n \times n$ diagonalizable matrix with real eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ and eigenvectors $\{|E_1\rangle, \ldots, |E_n\rangle\}$ of unit norm, then the goal is to implement the following transformation
\begin{equation}
\sum_{j=1}^{n} \beta_j |E_j\rangle \rightarrow \sum_{j=1}^{n} \beta_j |\tilde{\lambda}_j\rangle |E_j\rangle
\end{equation}
COMPUTING EIGENVALUES OF MATRICES IN A QUANTUM COMPUTER

up to a normalization, where \( \tilde{\lambda}_j \) is an approximation of \( \lambda_j \). The main techniques are quantum algorithms to solve linear differential equations and quantum phase estimation.

The assumption of diagonalizability implies that the eigenvectors are linearly independent so that any vector is a linear combinations of them. Thus the left hand side of (2.1) can be any quantum state. The class of diagonalizable matrices only has real eigenvalues may not general, but it includes Hermitian matrices. It is shown in [20] that these kind of matrices are similar to Hermitian matrices. Computing the eigenvalues of these kind of matrices may not easy classically. A parallelizable algorithm was proposed in [28], which heavily depends on matrix multiplication. The required operations is \( O(n^3) \). In the quantum case, if we can efficiently implement the transformation (2.1), then we can obtain all the eigenvalues of \( M \) efficiently by measurements.

2.1. The quantum algorithm to implement (2.1). In the subsection, we present a quantum algorithm to implement the transformation (2.1). The main idea of the algorithm is to change the eigenvalue estimating problem into a differential equations solving problem. Then use quantum algorithms to solve the differential equations to obtain the quantum state of the solution. It turns out the quantum state of the solution is close to the state \( |\psi_1\rangle \) defined in the QPE (see Algorithm 3 in Appendix A).

First we consider the special case that the left hand side of equation (2.1) is \( |E_j\rangle \) for some \( j \). For simplicity, we assume that \( |E_j\rangle \) is given. Later we will extend the idea to the general case in which there is no need to know the eigenvectors in advance. Now let us start from the following linear system of differential equations

\[
\begin{align*}
\frac{dx(t)}{dt} &= 2\pi i M x(t), \\
x(0) &= |E_j\rangle.
\end{align*}
\]

It is easy to see that the solution of this differential equation is

\[
 x(t) = e^{2\pi i M t} |E_j\rangle = e^{2\pi i \lambda_j t} |E_j\rangle.
\]

Since \( M \) is not Hermitian, it may not easy to apply \( e^{2\pi i M t} \) directly to \( |E_j\rangle \) to obtain the solution. Fortunately, there are some quantum algorithms [6,7,15] to solve differential equations in the quantum computer. It is interesting to see that the differential equation (2.2) satisfies the assumptions of the quantum algorithms proposed in [6,7,15]. Thus we can apply their results directly. In the following, we use the idea of [7] to solve the differential equation (2.2). Due to the speciality of our case, the analysis of the error and the complexity is much simplified.

For any integer \( k \) and any \( z \in \mathbb{C} \), denote the \((k+1)\)-terms truncation of exponential function \( e^z \) as

\[
 T_k(z) = \sum_{j=0}^{k} \frac{z^j}{j!}.
\]

If \( r \in \mathbb{R} \) and \( |r| < 1 \), then it is easy to show that \( |T_k(ir) - e^{ir}| \leq e/(k+1)! \).

To solve the differential equation, we discrete the time interval \([0,t]\) into \( m \) short time intervals by setting \( t_0 = 0, t_1 = \Delta t, t_2 = 2\Delta t, \ldots, t_m = m\Delta t = t \). The final time \( t = 1/2|\lambda_{\max}| \), where \( \epsilon \) is the precision to approximate the eigenvalue \( \lambda_j \) and \( |\lambda_{\max}| = \max_j |\lambda_j| \). The integer \( m = \Theta(1/\epsilon) \) and \( \Delta t = \Theta(1/|\lambda_{\max}|) \).
Set $d = m(k + 1)$, define the $(d + 1)n \times (d + 1)n$ matrix $C_{m,k}$ by

\begin{align}
C_{m,k}(2\pi iM\Delta t) &= \sum_{p=0}^{d} |p\rangle\langle p| \otimes I - \sum_{p=0}^{m-1} \sum_{q=1}^{k} |p(k + 1) + q\rangle\langle p(k + 1) + q| \otimes \frac{2\pi iM\Delta t}{q} \\
&\quad - \sum_{p=0}^{m-1} \sum_{q=0}^{k} |(p + 1)(k + 1)\rangle\langle p(k + 1) + q| \otimes I.
\end{align}

The goal of this matrix is to implement $T_k(2\pi iM\Delta t)$ without computing matrix powers. The definition of this matrix is a special case of the one defined in [7]. We delete their fourth summation term here. This is caused by the different goals. In [7], the authors aim to compute the quantum state of the solution at time $t_m$. The fourth term is introduced to improve the probability of this state. However, in this paper we need the superposition of the quantum state of the solution at time $t_0, \ldots, t_m$. Thus, the solution at time $t_m$ is not special for us.

Before doing detailed analysis, we first state the main idea briefly. By equation (2.3), the state of the superposition of the solution at time $t_0, \ldots, t_m$ is

\begin{equation}
\frac{1}{\sqrt{m+1}} \sum_{l=0}^{m} |l\rangle|x(l)\rangle = \frac{1}{\sqrt{m+1}} \sum_{l=0}^{m} e^{2\pi il\lambda_j\Delta t}|l\rangle|E_j\rangle.
\end{equation}

We can extract $\lambda_j$ easily by applying quantum Fourier inverse transform to the first register. Quantum linear differential equations solver can return an approximation of the state (2.7). Below, we consider the error caused by the discretization method.

Consider the following linear system

\begin{equation}
C_{m,k}(2\pi iM\Delta t)\tilde{x} = |0..0\rangle|E_j\rangle.
\end{equation}

In the right hand side, $|0..0\rangle$ is added to make sure both sides have the same dimension. Here $\tilde{x}$ refers to the approximated solution of the differential equation (2.2). We can formally write the solution as

\begin{equation}
\tilde{x} = \sum_{p=0}^{m-1} \sum_{q=0}^{k} |p(k + 1) + q\rangle|x_{p,q}\rangle + |m(k + 1)\rangle|x_{m,0}\rangle.
\end{equation}

In the above expression, we use the ket notation to simplify the expression of $\tilde{x}$, but $|x_{p,q}\rangle$ may not be a unit vector.

It is easy to check that (see Appendix B)

\begin{align}
|x_{0,0}\rangle &= |E_j\rangle, \\
|x_{0,q}\rangle &= \frac{(2\pi iM\Delta t)^q}{q!}|E_j\rangle = \frac{(2\pi i\lambda_j\Delta t)^q}{q!}|E_j\rangle, \quad (q = 1, \ldots, k).
\end{align}

If $p > 0$, then

\begin{align}
|x_{p,0}\rangle &= T_k(2\pi iM\Delta t)|x_{p-1,0}\rangle = T_k(2\pi i\lambda_j\Delta t)|x_{p-1,0}\rangle, \\
|x_{p,q}\rangle &= \frac{(2\pi iM\Delta t)^q}{q!}|x_{p,0}\rangle = \frac{(2\pi i\lambda_j\Delta t)^q}{q!}|x_{p,0}\rangle, \quad (q = 1, \ldots, k).
\end{align}
**Lemma 2.1.** Assume that $2\pi \Delta t |\lambda_j| < 1$ and $e \approx 2.718$ is the Euler’s number, then for any $p \in \{1, 2, \ldots, m\}$, we have

\[
\|x(t_p)) - |x_{p,0}\|_2 \leq \left(1 + \frac{e}{(k + 1)!}\right)^p - 1.
\]

Moreover,

\[
\left|\|x_{p,0}\|_2 - 1\right| \leq \left(1 + \frac{e}{(k + 1)!}\right)^p - 1.
\]

**Proof.** By equations (2.3), (2.12),

\[
\|x(t_p)) - |x_{p,0}\|_2 = \|e^{2\pi i \lambda_j \Delta t} x(t_{p-1}) - T_k(2\pi i \lambda_j \Delta t) |x_{p-1,0}\|_2 \\
\leq |e^{2\pi i \lambda_j \Delta t} - T_k(2\pi i \lambda_j \Delta t)| + |T_k(2\pi i \lambda_j \Delta t)| \|x(t_{p-1})\|_2 - |x_{p-1,0}\|_2 \\
\leq \frac{e}{(k + 1)!} + \left(1 + \frac{e}{(k + 1)!}\right) \|x(t_{p-1})\|_2 - |x_{p-1,0}\|_2 \\
\leq \frac{e}{(k + 1)!} \sum_{r=0}^{p-1} \left(1 + \frac{e}{(k + 1)!}\right)^r \\
= \frac{e}{(k + 1)!} \left(1 + \frac{e}{(k + 1)!}\right)^p - 1 \\
= \left(1 + \frac{e}{(k + 1)!}\right)^p - 1.
\]

This proves the first claim. Since $|x(t_p))$ is unit, the second result comes naturally. $\square$

**Proposition 2.2.** Let $|x) = \frac{1}{\sqrt{m+1}} \sum_{p=0}^m |p) |x(t_p))$ be the superposition of the exact solution of the differential equation (2.2). Denote the quantum state of $\sum_{p=0}^m |p) |x_{p,0}\$ as $|\hat{x}\rangle$, then

\[
\|x) - |\hat{x}\rangle\|_2 \leq 2(m + 1) \left(1 + \frac{e}{(k + 1)!}\right)^{m/2} \left(2 - \left(1 + \frac{e}{(k + 1)!}\right)^m\right)^{-1/2}.
\]

**Proof.** Denote the $l_2$-norm of $\sum_{p=0}^m |p) |x_{p,0}\$ as $\sqrt{L}$. Thus

\[
\|x) - |\hat{x}\rangle\|_2 = \sum_{p=0}^m \frac{|x(t_p)) - |x_{p,0}\|_2}{\sqrt{m+1}} \\
\leq \sum_{p=0}^m \frac{1}{\sqrt{m+1}} - \frac{1}{\sqrt{L}} + \frac{1}{\sqrt{L}} \sum_{p=0}^m \|x(t_p)) - |x_{p,0}\|_2.
\]

By Lemma 2.1,

\[
|L - m - 1| \leq (m + 1) \left(1 + \frac{e}{(k + 1)!}\right)^{m/2} =: (m + 1)A,
\]

so

\[
\left|\frac{1}{\sqrt{m+1}} - \frac{1}{\sqrt{L}}\right| = \frac{|L - m - 1|}{\sqrt{L}(m + 1)(\sqrt{m+1} + \sqrt{L})} \leq \frac{A}{\sqrt{1 - A}}.
\]
Thus the first term of (2.18) is bounded by \((m + 1)A/\sqrt{1 - A}\). Again by Lemma 2.1, the second term of (2.18) is bounded by \((m + 1)^{1/2}A/\sqrt{1 - A}\). This completes the proof.

Proposition 2.2 states that if we choose \(k\) such that \((k + 1)! \gg m\), then the quantum state of the solution of the differential equation (2.2) is approximated by the part of the quantum state of the solution of the linear system (2.8) with \(q = 0\). The following lemma further shows that this part occupies an constant amplitude. If we perform amplitude amplification, then we can enlarge this amplitude close to 1 with \(O(1)\) extra operations. As a result, the quantum state of the solution of the linear system (2.8) provides a good approximation of the the quantum states of the solution of the differential equation (2.2).

**Lemma 2.3.** Assume that \(2\pi\lambda_j \Delta t < 1\), then the \(l_2\)-norm of the solution \(\hat{x}\) of the linear system (2.8) satisfies

\[
(2.19) \quad \left\| \hat{x} \right\|_2^2 - e^{2\pi\lambda_j \Delta t} \sum_{p=0}^{m-1} \left\| x_{p,0} \right\|_2^2 - \left\| x_{m,0} \right\|_2^2 \leq \frac{e}{(k + 1)!}.
\]

**Proof.** By equation (2.9)

\[
\left\| \hat{x} \right\|_2^2 = \sum_{p=0}^{m-1} \sum_{q=0}^{k} \left\| x_{p,q} \right\|_2^2 + \left\| x_{m,0} \right\|_2^2
\]

\[
= \sum_{p=0}^{m-1} \sum_{q=0}^{k} \frac{(2\pi\lambda_j \Delta t)^q}{q!} \left\| x_{p,0} \right\|_2^2 + \left\| x_{m,0} \right\|_2^2
\]

\[
= T_k(2\pi\lambda_j \Delta t) \sum_{p=0}^{m-1} \left\| x_{p,0} \right\|_2^2 + \left\| x_{m,0} \right\|_2^2.
\]

It is easy to prove that

\[
|T_k(2\pi\lambda_j \Delta t) - e^{2\pi\lambda_j \Delta t}| \leq \frac{e}{(k + 1)!}.
\]

This completes the proof.

**Proposition 2.4.** Let \(M\) be an \(n \times n\) diagonalizable matrix which only has real eigenvalues. Assume that one of its eigenvector is known. Suppose that the linear system (2.5) can be solved in time \(T(\kappa(C_m,k), \epsilon, m(k + 1)n)\) to precision \(\epsilon\). Then we can estimate the corresponding eigenvalue in a quantum computer to precision \(|\lambda_{\text{max}}|\epsilon\) in cost \(O(1)\).

\[
(2.20) \quad T(\kappa(C_m,k), \epsilon, nc^{-1}\log(1/\epsilon)).
\]

**Proof.** By Proposition 2.2, to make sure the error between \(|x\) and \(|\hat{x}\) is smaller than \(\epsilon\), we can choose \(k\) such that \((k + 1)! \geq m^2/\epsilon\). That is \(k \geq \log(m/\epsilon)\). Thus, we can set \(k = O(\log(m/\epsilon)) = O(\log(1/\epsilon))\) as \(m = O(1/\epsilon)\).

Let \(|\hat{x}\) be the quantum state obtained by the quantum linear solver to solve the linear system (2.5). Since \(2\pi\Delta t|\lambda_j| < 1\), Lemma 2.3 shows that in \(|\hat{x}\) the amplitude of \(|\hat{x}\) is close to \(e^{-2\pi\Delta t|\lambda_j|} > e^{-1} > 0.36\). Apply amplitude amplification with \(O(1)\) operations, we can increase the amplitude of \(|\hat{x}\) close to 1 in \(|\hat{x}\). As a result, we have \(|\|x\) - \(|\hat{x}\)||_2 \leq \epsilon\). Since

\[
|x\) = \frac{1}{\sqrt{m + 1}} \sum_{l=0}^{m} |l\rangle x(t_l)\rangle = \frac{1}{\sqrt{m + 1}} \sum_{l=0}^{m} e^{2\pi i\lambda_j \Delta t |l\rangle E_j\rangle}.
\]
If we apply quantum Fourier inverse transform to the first register of $|\hat{x}\rangle$, then we can obtain an $\epsilon$ approximation of

$$\frac{1}{m+1} \sum_{k=0}^{m} \left( \sum_{l=0}^{m} e^{2\pi i l (\lambda_j \Delta t - \frac{k}{m+1})} \right) |k\rangle |E_j\rangle.$$  

Based on the analysis of QPE, we can obtain $k$ such that $k/m$ is an $\epsilon$ approximation of $\lambda_j \Delta t$ with probability close to 1.

The complexity is dominated by the solving the linear system (2.5). Based on the choices of $k$ and $m$, we obtain the complexity of the above procedure easily. \hfill \square

In a quantum computer, under certain conditions, the complexity to solve a linear system is linear on the condition number, logarithm on the precision and the dimension. For the linear system (2.5), the dimension is $m(k+1)n$. As proved in Theorem 5 of [7], the condition number of $C_{m,k}$ is bounded by $O(\kappa(E)\epsilon^{-1}(\log 1/\epsilon))$, where $\kappa(E)$ is the condition number of the matrix generated by the eigenvectors. Thus, we can simplify the result of (2.4) into $O(\kappa(E)\epsilon^{-1}(\log 1/\epsilon)(\log n)).$

The general case can be done similarly. We summarize the idea into the following algorithm.

**Algorithm 1** Quantum algorithm for computing the real eigenvalues

**Input:** (1). An $n \times n$ diagonalizable matrix $M$ only with real eigenvalues. Suppose the eigenvalues are $\{\lambda_1, \ldots, \lambda_n\}$ and the unit eigenvectors are $\{|E_1\rangle, \ldots, |E_n\rangle\}$.

(2). A upper bound $|\lambda_{\text{max}}|$ of the eigenvalues, i.e., $|\lambda_j| \leq |\lambda_{\text{max}}|$ for all $j$.

(3). A quantum state $|\phi\rangle$ which formally equals $\sum_{j=1}^{n} \beta_j |E_j\rangle$.

(4). The precision $\epsilon \in (0, 1)$, $\Delta t = 1/2|\lambda_{\text{max}}|$, $m = \lceil 1/\epsilon \rceil$, $k = \log(1/\epsilon)$.

**Output:** The quantum state

$$(2.23) \sum_{j=1}^{n} \beta_j |\tilde{\lambda}_j\rangle |E_j\rangle$$

up to a normalisation, where $|\tilde{\lambda}_j - \lambda_j| \leq |\lambda_{\text{max}}|\epsilon$ for all $j$.

1: Construct the matrix $C_{m,k}$ based on equation (2.5).
2: Use quantum linear algebraic technique to construct $C_{m,k}^{-1}|0..0\rangle|\phi\rangle$.
3: Apply quantum Fourier inverse transform to the first register of $C_{m,k}^{-1}|0..0\rangle|\phi\rangle$.

**Theorem 2.5.** Let $M$ be an $n \times n$ diagonalizable matrix which only has real eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$. Assume that the corresponding unit eigenvectors are $\{|E_1\rangle, \ldots, |E_n\rangle\}$. Let $|\lambda_{\text{max}}|$ be a upper bound of the eigenvalues. Suppose that $C_{m,k}^{-1}$ can be implemented in a quantum computer in time $T(\kappa(C_{m,k}), \epsilon, m(k+1)n)$ to precision $\epsilon$. Then we can perform the following transformation

$$(2.24) \sum_{j=1}^{n} \beta_j |E_j\rangle \mapsto \frac{1}{Z} \sum_{j=1}^{n} \beta_j |\tilde{\lambda}_j\rangle |E_j\rangle$$

in time

$$(2.25) O\left(T\left(\kappa(E)\epsilon^{-1}\log(1/\epsilon), \epsilon, n\epsilon^{-1}\log(1/\epsilon)\right) \times \frac{\kappa(E)}{\epsilon} \sqrt{\log \frac{1}{\epsilon}}\right),$$
where $Z$ is the normalization factor, $\kappa(E)$ is the condition number of the matrix generated by the eigenvectors and $|\lambda_j - \hat{\lambda}_j| \leq |\lambda_{\text{max}}|\epsilon$ for all $j$.

Proof. By equation (2.9), up to a normalization

$$
C_{m,k}^{-1}|0..0\rangle|\phi\rangle = \sum_{j=1}^{n} \beta_j C_{m,k}^{-1} |0..0\rangle|E_j\rangle
$$

$$
= \sum_{j=1}^{n} \beta_j \left( \sum_{p=0}^{m} |p(k+1)\rangle|x_{p,0}\rangle + \sum_{q=1}^{m-1} \sum_{p=0}^{k} |p(k+1) + q\rangle|x_{p,q}\rangle \right).
$$

As $k+1$ is invertible modulo $m(k+1)+1$, we can find $(k+1)^{-1}$ and multiple it on the first register. As a result, we obtain

$$
\sum_{j=1}^{n} \beta_j \left( \sum_{p=0}^{m} |p\rangle|x_{p,0}\rangle|0\rangle + |0\rangle^\perp \right).
$$

In the above state, we add a new ancilla qubit $|0\rangle$ to separate the two summation terms. This is possible as the first registers are orthogonal between them.

By equation (2.3) and Lemma 2.1, equation (2.26) is $\epsilon$-close to the state

$$
\sum_{j=1}^{n} \sum_{p=0}^{m} \beta_j e^{2\pi i \lambda_j p \Delta t} |E_j\rangle|0\rangle + |0\rangle^\perp.
$$

If we apply quantum Fourier inverse transform to it, then we obtain

$$
\frac{1}{\sqrt{m+1}} \sum_{j=1}^{n} \sum_{q=0}^{m} \sum_{p=0}^{m} \beta_j e^{2\pi i (\lambda_j p \Delta t - \frac{n}{m+1})} |q\rangle|E_j\rangle|0\rangle + |0\rangle^\perp \approx \sqrt{m+1} \sum_{j=1}^{n} |\lambda_j\rangle|E_j\rangle|0\rangle + |0\rangle^\perp.
$$

By [7, Lemma 3], $\|C_{m,k}^{-1}\| \leq 3\kappa(E)\sqrt{km}$. As a result, $\|C_{m,k}^{-1} |0..0\rangle|\phi\rangle\| \leq \|C_{m,k}^{-1}\| \leq 3\kappa(E)\sqrt{km}$. The amplitude of $|0\rangle$ in the state (2.27) equals

$$
\left\| \sum_{p=0}^{m} \sum_{j=1}^{n} \beta_j e^{2\pi i \lambda_j p \Delta t} |E_j\rangle \right\|_2^2.
$$

Note that if $p = 0$, then $\left\| \sum_{j=1}^{n} \beta_j |E_j\rangle \right\|_2^2 = \|\phi\|_2^2 = 1$. Therefore, the amplitude of $|0\rangle$ is larger than $1/3\kappa(E)\sqrt{km}$. By amplitude amplification and the choices of the parameters in the algorithm, the complexity to obtain the state (2.24) is (2.25) as claimed.

Remark 2.6. For eigenvalue problems of diagonalizable matrices, Bauer and Fike in [4] showed the condition number $\kappa(E)$ describes the stability and conditioning of calculating the eigenvalues. If $\kappa(E)$ is large, then small permutations on the matrix $M$ will give rise to large permutations on the eigenvalues. This makes the calculation of the eigenvalues inaccurate. Thus the complexity of Algorithm 1 depends on $\kappa(E)$ seems acceptable. But the quadratic form can be improved to linear by using the variable time amplitude amplification technique [2].
In the following, we consider sparse matrices in which the complexity of inverting $C_{m,k}$ is known. The following lemma about the sparsity of $C_{m,k}$ is easy to prove.

**Lemma 2.7.** If $M$ is $s$-sparse, then the sparsity of $C_{m,k}$ is $\Theta(s + k)$.

**Proof.** By equation (2.5), for any $p \in \{0, \ldots, m\}$, $q \in \{0, \ldots, k\}$ and $r \in \{1, \ldots, n\}$

$$\langle p(k+1) + q | \otimes | r \rangle = \langle p(k+1) + q | \otimes | r \rangle - \langle p(k+1) + q - 1 | \otimes | r \rangle \frac{2\pi i M \Delta t}{q}$$

$$- \delta_q^p \sum_{q'=0}^{k} \langle p(k+1) + q' | \otimes | r \rangle.$$

The nonzero element of this row vector is bounded by $1 + s + k + 1 = \Theta(s + k)$. Similar analysis also holds for the columns of $C_{m,k}$. □

By the quantum linear solver [14, Theorem 5], if $M$ is $s$-sparse, then combining Lemma 2.7,

$$T(\kappa(C_{m,k}), \epsilon, m(k+1)n)$$

$$= O((s + \log(1/\epsilon))\kappa(C_{m,k})(\log n \epsilon^{-1} \log(1/\epsilon))polylog((s + \log(1/\epsilon))\kappa(C_{m,k})/\epsilon)).$$

By [7, Theorem 5], $\kappa(C_{m,k}) = O(\kappa(E)km) = O(\kappa(E)\epsilon^{-1}(\log 1/\epsilon))$, Thus, equation (2.25) can be simplified into $O(\kappa(E)^2/\epsilon^2)$. The quantum linear solver [14] depends an oracle $O_C$ to query $C_{m,k}$. Assume that we have the oracle $O_M$ to query $M$. It is defined as

$$O_M|i,j,z\rangle = |i,j,z + m_{ij}\rangle,$$

where $\nu(i,l)$ is the index of the $l$-th nonzero element in the $i$-th row/column. For any $0 \leq p_1, p_2 \leq m, 0 \leq q_1, q_2 \leq k$ and $1 \leq r_1, r_2 \leq n$, it is easy to check that

$$\langle p_1(k+1) + q_1, r_1 | C_{m,k} | p_2(k+1) + q_2, r_2 \rangle = \delta_{r_2}^{r_1} \delta_{p_2}^{p_1} \delta_{q_2}^{q_1} - [p_2 \leq m - 1][q_2 \leq k - 1] \delta_{p_2}^{p_1} \delta_{q_2+1}^{q_1} r_1 q_2 + 1 [r_2] \frac{2\pi i M \Delta t}{q_2 + 1} | r_2 \rangle$$

$$- [p_2 \leq m - 1] \delta_{r_2}^{r_1} \delta_{p_2+1}^{p_1} \delta_{q_2}^{q_1},$$

where the notation $[a \leq b]$ means that it equals $1$ if $a \leq b$ and $0$ otherwise. The three terms in equations (2.31), (2.32) cannot coexist, so it is easy to build the oracle $O_C$ to query $C_{m,k}$ when we have $O_M$. The cost to build this oracle $O_C$ is the same as that to build $O_M$.

**Corollary 2.8.** With the same assumptions as Theorem 2.5. If $M$ is $s$-sparse, then we can implement the transformation (2.24) in time $O(\kappa(E)^2/\epsilon^2)$.

To obtain all the eigenvalues classically, we can perform measurements on the right hand side of equation (2.24). The following lemma shows how many measurements we should perform.

**Lemma 2.9.** Let $\mathcal{P} = \{p_1, \ldots, p_n\}$ be a probability distribution. Set $p_{\text{max}} = \max\{p_1, \ldots, p_n\}$. To see every events under the distribution $\mathcal{P}$ it suffices to make $O(p_{\text{max}}^{-1} \log(n/\delta))$ measurements. The success probability is at least $1 - \delta$.

We defer the proof of this lemma to Appendix C. To apply Lemma 2.9, we need to analyze the complexity to obtain each eigenvalue. Assume that $M$ has $d$ distinct eigenvalues $\lambda_1, \ldots, \lambda_d$,
we can rewrite the initial state as \( \sum_{j=1}^{d} \gamma_j |V_j\rangle \), where \( |V_j\rangle \) is a normalized vector generated by the eigenvectors corresponding to \( \lambda_j \). Then the right hand side of equation (2.24) can be written as 
\[
\frac{1}{Z} \sum_{j=1}^{d} \gamma_j |\tilde{\lambda}_j\rangle |V_j\rangle.
\]
The probability to obtain \( \tilde{\lambda}_j \) equals \( p_j = |\gamma_j|^2/Z^2 \). Note that \( Z^2 = \sum_{j=1}^{d} |\gamma_j|^2, \) so \( p_{\text{max}} = \max_j |\gamma_j|^2/Z^2 \geq 1/d \). Thus, it suffices to make \( O(d \log d) = O(n \log n) \) measurements by Lemma 2.9. The sparse access oracle of \( M \) can be built in time \( O(sn) \). Thus we obtain the following result.

**Theorem 2.10.** Let \( M \) be an \( s \) sparse diagonalizable matrix that only has real eigenvalues and \( E \) the matrix generated by all the unit eigenvectors. Let \( |\lambda_{\text{max}}| \) be a upper bound of the eigenvalues. Then Algorithm 1 together with measurements can obtain all the eigenvalues of \( M \) to precision \( |\lambda_{\text{max}}| \epsilon \) in cost \( \tilde{O}(snk(E)^2/\epsilon^2) \).

### 2.2. What if the diagonalizable matrices have complex eigenvalues.

Algorithm 1 also works for diagonalizable matrices that only have pure imaginary eigenvalues. It suffices to change \( 2\pi iM \) into \( 2\pi M \) in equation (2.2). In the following, we consider the problem that what would happen if we apply Algorithm 1 directly to any diagonalizable matrix. We will show that Algorithm 1 has many difficulties to estimate the complex eigenvalues.

To do the analysis, we just focus on the special case that the \( j \)-th eigenvector \( |E_j\rangle \) is known. Denote the corresponding eigenvalue are \( \lambda_{j0} + i\lambda_{j1} \), where \( \lambda_{j0}, \lambda_{j1} \in \mathbb{R} \). Then similar to the analysis of obtaining (2.21), we will obtain an approximation of

\[
|x\rangle = \frac{1}{Z} \sum_{l=0}^{m} |l\rangle |x(t_l)\rangle = \frac{1}{Z} \sum_{l=0}^{m} e^{2\pi i \lambda_{j1} \Delta t} l |E_j\rangle,
\]

where \( Z \) is the normalization constant. Since \( \lambda_j \in \mathbb{C} \), usually \( Z \neq \sqrt{m+1} \). If we set \( \lambda_{j1} \Delta t = -b \) for simplicity, then

\[
Z^2 = \sum_{l=0}^{m} e^{4\pi bl} = \frac{1 - e^{4\pi b(m+1)}}{1 - e^{4\pi b}}.
\]

To make sure the quantum differential equation solver works, one assumption made in [7] is that the real part of the coefficient matrix is non-positive. This assumption relates to the stability of the differential equation. Here we should make the same assumption, that is \( \lambda_{j1} > 0 \) as we use \( 2\pi M \) in the differential equation (2.2). Thus \( b < 0 \). When concerning about computing eigenvalues, this assumption is easy to be satisfied. We just need to consider \( M - i|\lambda_{\text{max}}| I \) instead, where \( |\lambda_{\text{max}}| \) is some upper bound of the eigenvalues.

If we apply quantum Fourier inverse transform to the first register of \( |x\rangle \), then we obtain

\[
\frac{1}{Z \sqrt{m+1}} \sum_{k=0}^{m+1} \sum_{l=0}^{m+1} e^{2\pi i (\lambda_{j0} \Delta t - \frac{k}{m+1})} e^{2\pi lb} |k\rangle |E_j\rangle.
\]

For convenience, we set \( a = \lambda_{j0} \Delta t - \frac{k}{m+1} \). From equation (2.34), \( k \) only provides an estimation of \( \lambda_{j0} \). Now we assume that \( |a| \leq \epsilon = 1/(m+1) \), then the probability of this \( k \) is

\[
P = \frac{1}{Z^2(m+1)} \sum_{k=0}^{m+1} |e^{2\pi i (a-ib)}|^2 = \frac{1}{Z^2(m+1)} \left| e^{2\pi i (a-ib)} - 1 \right|^2.
\]

\[
= \frac{1}{Z^2(m+1)} \frac{(e^{2\pi b(m+1)} - 1)^2 + 4e^{2\pi b(m+1)} \sin^2(\pi(m+1)a)}{(e^{2\pi b} - 1)^2 + 4e^{2\pi b} \sin^2(\pi a)}.
\]
To make sure \( P \) large, a necessary condition is \( e^{2\pi b(m+1)} \approx 0 \), so we can assume that \( b = -C/(m+1) \) for some \( C < m + 1 \). Consequently,

\[
P = \frac{1}{m + 1} \frac{1 - e^{4\pi b}}{1 - e^{4\pi b(m+1)}} \frac{(e^{2\pi b(m+1)} - 1)^2 + 4e^{2\pi b(m+1)} \sin^2(\pi(m+1)a)}{(e^{2\pi b} - 1)^2 + 4e^{2\pi b} \sin^2(\pi a)}
\]

\[
\approx \frac{1}{m + 1} \frac{C/(m+1)}{C/m^2/(m+1)^2 + 4e^{2\pi b} \pi^2/(m+1)^2}
\]

\[
\approx \frac{1}{C}.
\]

This seems fine as we can choose \( C = O(1) \) or \( C = \text{poly} \log m \) for instance. However, \( b = -C/(m+1) \) implies that \( \Delta t = C/(m+1)\lambda_{j1} \). We only have

\[
|a| = \left| \lambda_{j0} \Delta t - \frac{k}{m+1} \right| = \left| \lambda_{j0} \frac{C}{(m+1)\lambda_{j1}} - \frac{k}{m+1} \right| \leq \frac{1}{m+1},
\]

thus we obtain

\[
\left| \lambda_{j0} - \frac{k\lambda_{j1}}{C} \right| \leq \frac{\lambda_{j1}}{C}.
\]

As an approximation, we hope the right hand side of equation (2.40) is small, that is \( C \) should be large. However, if \( C \) is large, then \( P \) is small. Perform measurements on the quantum state (2.34), if \( P \) is large (say close to 1), then we have more confidence that the measuring result gives a good approximation of the eigenvalue. If \( P \) is small (say \( 1/\log m \)), we need perform many measurements. But from the results obtained by the measurements, we still have difficulties to determine which one provides an good approximation of the eigenvalue. We cannot distinguish them from the probability distribution. What we can do is to check the results by the definition of eigenvalues. By doing so, we need to perform matrix-vector multiplication in a classical computer, which is not the original goal of the quantum eigenvalue algorithm.

The above analysis shows that when the eigenvalues are complex, Algorithm 1 may not find a good approximation of the eigenvalues, even the real parts.

Remark 2.11. To compute all the complex eigenvalues, it suffices to have a quantum algorithm to compute all the real parts. More precisely, let \( M \) be a diagonalizable matrix with complex eigenvalues \( \lambda_j + i\mu_j \), where \( j = 1, \ldots, n \). The corresponding unit eigenvalues are \( |E_1\rangle, \ldots, |E_n\rangle \). Suppose that we have a quantum algorithm that can output \( \sum_{j=1}^n \beta_j |\lambda_j\rangle |E_j\rangle \) (up to a normalization) when the input is \( \sum_{j=1}^n \beta_j |\lambda_j\rangle |E_j\rangle \). Then we can apply this algorithm further to obtain \( \sum_{j=1}^n \beta_j |\lambda_j\rangle |\mu_j\rangle |E_j\rangle \). This is obtained by considering \( iM \) and viewing \( \sum_{j=1}^n \beta_j |\lambda_j\rangle |E_j\rangle \) as the new initial state.

3. Normal matrices. A square matrix is called normal if it commutes with its conjugate transpose. The condition of normality may be strong, but it includes the unitary, Hermitian, skew-Hermitian matrices and their real counterparts as special cases. These matrices are of great interests to physicists [36]. In [26]. Grone et al. listed 70 different equivalent conditions of normal matrices. 19 more were added later in [21]. One interesting result we will use in this paper is the fact that normal matrices can be diagonalized by unitary matrices. The list [21, 26] reflects the fact that normality arises in many ways.

Computing eigenvalues of (normal) matrices is not easy classically. A special kind of normal matrices whose eigenvalues are easy to compute are circulant matrices [25]. They can be diagonalized by Fourier transform. Based on the fast Fourier transform, the eigenvalues can be obtained in
costs $O(n \log n)$. Generally we should use iterative algorithms to find the eigenvalues. These kind of algorithms typically proceed in two phases [23,39]: First, transform the matrix to a suitable condensed matrix format (e.g. Hessenberg or tridiagonal), sharing the eigenvalues; Second, compute the eigenvalues of the condensed matrix. The intermediate matrix can save a lot of the computing time. For instance, for an $n \times n$ Hermitian matrix, first we use Lanczos algorithm to transform it into a real tridiagonal symmetric matrix. Then use the divide-and-conquer algorithm [18] to compute all the eigenvalues of the tridiagonal matrices. The complexity of the the two steps are $O(sn^2)$ and $O(n \log n)$ respectively, where $s$ is the sparsity of the Hermitian matrix. Consequently, the total complexity is $O(sn^2)$. Usually, the complexity is dominated by the first step. For unitary matrices, the complexity to compute the eigenvalues is $O(n^3)$ in the Hessenberg case [24], and $O(n^3)$ in the general case [9].

For generic normal matrices, condensed matrix formats with $O(n)$ parameters does not exist. The intermediate structures are often of Hessenberg form, which requires $O(n^2)$ storage. If we use QR algorithm to compute all the eigenvalues of the Hessenberg matrices, then each step of iteration costs $O(n^2)$. If Arnoldi method is used in the first step, then the complexity is bounded by $O(n^3)$. In [22], Ferranti and Vandebril proposed an algorithm to find a memory cheap intermediate matrix of tridiagonal complex symmetric form for normal matrices. However, their algorithm to compute all the eigenvalues of generic normal matrices costs $O(n^3)$. When the spectral decomposition is known of the $(n-1) \times (n-1)$ principal minors, Cecchi et al. [10] showed that the eigenvalues of $n \times n$ normal matrices can be computed in costs $O(n^2)$.

### 3.1. Main result.

Assume that $M$ is an $n \times n$ normal matrix, then there exists a diagonal matrix $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ and a unitary matrix $U$ such that $M = U\Lambda U^\dagger$. Thus $\{\sigma_j := |\lambda_j| : j = 1, \ldots, n\}$ are the singular values of $M$. Denote $\lambda_j = \sigma_j e^{2\pi i \theta_j}$, the problem we want to solve in this section is to estimate $\sigma_j, \theta_j$ for all $j$ to certain precision in the quantum computer. To be more precise, assume that the $j$-th column of $U$ is $|u_j\rangle$. Let $|b\rangle$ be any given state. Since $U$ in unitary, there exist $\beta_1, \ldots, \beta_n$ such that $|b\rangle = \sum_{j=1}^n \beta_j |u_j\rangle$. The main objective we want to obtain is to implement the following transformation in the quantum computer:

$$\sum_{j=1}^n \beta_j |u_j\rangle |0\rangle \rightarrow \sum_{j=1}^n \beta_j |u_j\rangle |\tilde{\sigma}_j\rangle |\tilde{\theta}_j\rangle,$$

where $\tilde{\sigma}_j$ and $\tilde{\theta}_j$ are respectively the approximations of $\sigma_j$ and $\theta_j$.

To state our main result, we need the technique of quantum singular value estimation. Let $M$ be a $n$-by-$n$ matrix with singular value decomposition (SVD) $M = \sum_{j=1}^n \sigma_j |u_j\rangle \langle v_j|$. Denote

$$\tilde{M} = \begin{pmatrix} 0 & M \\ M^\dagger & 0 \end{pmatrix},$$

which is Hermitian. The eigenvalues of $\tilde{M}$ are $\{\pm \sigma_j : j = 1, \ldots, n\}$. The corresponding eigenvectors are

$$|w_j^\pm\rangle := \frac{1}{\sqrt{2}}(|0\rangle |u_j\rangle \pm |1\rangle |v_j\rangle), \quad j = 1, \ldots, n.$$

Based on quantum phase estimation, we can implement the following transformation

$$\sum_{j=1}^n \beta_j^+ |w_j^+\rangle + \beta_j^- |w_j^-\rangle \rightarrow \sum_{j=1}^n \beta_j^+ |w_j^+\rangle |\tilde{\sigma}_j\rangle + \beta_j^- |w_j^-\rangle |\tilde{\sigma}_j\rangle,$$
in a quantum computer, where $|\sigma_j - \hat{\sigma}_j| \leq \epsilon$. More details can be found in Appendix A. With this preliminary, our main result is stated as follows.

**Theorem 3.1.** Let $M$ be an $n$-by-$n$ normal matrix. Assume that its eigenvalue decomposition is $\sum_{j=1}^n \sigma_j e^{2\pi i \theta_j} |u_j\rangle\langle u_j|$. If the complexity to implement $(3.4)$ for $M$ is $O(T/\epsilon_1)$, then the transformation $(3.1)$ can be implemented in time $O(T/\epsilon_1 \epsilon_2)$ in a quantum computer, where $|\sigma_j - \hat{\sigma}_j| \leq \epsilon_1$, $|\theta_j - \hat{\theta}_j| \leq \epsilon_2$ for all $j$.

As a direct application of Theorem 3.1 and Proposition A.1, we have the following result.

**Corollary 3.2.** Let $M$ be an $s$ sparse $n \times n$ normal matrix, let $C$ be a upper bound of its eigenvalues. Then the complexity to implement $(3.1)$ is $O(sC\|M\|_{\text{max}}(\log n)/\epsilon_1 \epsilon_2)$.

There are several methods to find the upper bound of the eigenvalues. A simple method is based on the Gershgorin circle theorem [23], which states that every eigenvalue of $M$ lies within at least one of the Gershgorin discs $\{z \in \mathbb{C} : |z - m_{ii}| \leq \sum_{j \neq i} |m_{ij}|\}$, where $i = 1, \ldots, n$. As a result, $\|M\|_1, \|M\|_\infty$ are upper bound of the eigenvalues. The spectral norm $\|M\|$ also provides a upper bound of the eigenvalues. Since we can estimate the singular values in a quantum computer, we can estimate the spectral norm as well. For instance, see [31].

If we perform measurements on the right hand side of $(3.1)$, then we can obtain all the eigenvalues of $M$. The complexity is affected by the unknown coefficients $\beta_j$. Also if $\beta_j = 0$, then we cannot obtain the $j$-th eigenvalue by measuring the right hand side of $(3.1)$. Assume that $\beta_j \neq 0$ for $j \in S$. By Lemma 2.9, to obtain all the eigenvalues $\lambda_j$ with $j \in S$, it suffices to make $O((\max_j |\beta_j|^2)^{-1} \log n) = O(n \log n)$ measurements, where $\delta$ is ignored. Thus the complexity to obtain these eigenvalues is $O(n(\log n) T/\epsilon_1 \epsilon_2)$. If we randomly generate the initial states, then we can obtain all the eigenvalues.

For instance, if we choose the initial state as

$$|M\rangle = \frac{1}{\|M\|_F} \sum_{i,j=1}^n m_{ij} |i,j\rangle = \frac{1}{\|M\|_F} \sum_{i,j=1}^n \sigma_j e^{2\pi i \theta_j} |u_j\rangle |u_j\rangle,$$

where $m_{ij}$ is the $(i,j)$-th entry of $M$. The second equality comes from the definition of eigenvalue decomposition. Then the complexity to compute all nonzero eigenvalues of $M$ is

$$O \left( \frac{\|M\|_F^2}{\max_j \sigma_j} T/\epsilon_1 \epsilon_2 (\log n) \right) = O(n(\log n) T/\epsilon_1 \epsilon_2).$$

One underlying problem here is that $|M\rangle$ may not easy to prepare.

We can also use the language of density operator to describe the transformation $(3.1)$. Concerning about the influence of the unknown coefficient $\beta_j$, we choose the initial density operator as the maximally mixed state $I = \frac{1}{n} \sum_j |j\rangle \langle j|$ which also equals $\frac{1}{n} \sum_j |u_j\rangle \langle u_j|$ due to the unitary of $U$. Then $(3.1)$ becomes

$$\frac{1}{n} \sum_{j=1}^n |u_j\rangle \langle u_j| \to \frac{1}{n} \sum_{j=1}^n |u_j\rangle \langle u_j| \otimes |\tilde{\sigma}_j\rangle \langle \tilde{\sigma}_j| \otimes |\tilde{\theta}_j\rangle \langle \tilde{\theta}_j|.$$

To obtain all the eigenvalues, by Lemma 2.9 it suffices to make $O(n \log n)$ measurements on the right hand side of $(3.7)$. In this case, the complexity is $O(n(\log n) T/\epsilon_1 \epsilon_2)$. Thus by Corollary 3.2, we have
Theorem 3.3. Let \( M \) be an \( s \) sparse \( n \times n \) normal matrix with eigenvalues \( \sigma_j e^{i\theta_j}, j = 1, \ldots, n \). Let \( \epsilon_1, \epsilon_2 \) be the precisions to approximate all \( \sigma_j, \theta_j \) respectively. Assume that \( \sigma_j \leq C \) for all \( j \). Then there is a quantum algorithm to estimate all the eigenvalues in time \( O(snC\|M\|_{\text{max}}(\log n)^2/\epsilon_1\epsilon_2) \).

3.2. Proof of Theorem 3.1. For any \( j \), denote \( |v_j\rangle = e^{-2\pi i\theta_j}|u_j\rangle \), then the SVD of \( M \) is \( \sum_{j=1}^{n} \sigma_j |u_j\rangle \langle v_j| \). We can rewrite the initial state \(|0\rangle|b\rangle\) as \(|0\rangle|b\rangle = \sum_{j=1}^{n} \beta_j |u_j\rangle |v_j\rangle \). The main technique is described in the following lemma.

Lemma 3.4. With the same notation as Theorem 3.1, then there is a unitary operator \( W \) that performs

\[
W: \sum_{j=1}^{n} \beta_j |0\rangle |u_j\rangle \mapsto \sum_{j=1}^{n} \beta_j e^{-2\pi i\theta_j} |0\rangle |u_j\rangle.
\]

The operator \( W \) is implemented in time \( O(T/\epsilon_1) \).

Proof. By viewing \(|0\rangle|b\rangle\) as the initial state, if we perform the transformation (3.4) to it, then we obtain \( \sum_{j=1}^{n} \beta_j |0\rangle |u_j\rangle |\bar{\sigma}_j\rangle |\bar{\bar{\theta}}_j\rangle \). Since \( -\bar{\sigma}_j \leq 0 \), we can add a negative sign to the second term to change it into \( \sum_{j=1}^{n} \beta_j |0\rangle |u_j\rangle |\bar{\bar{\sigma}}_j\rangle |\bar{\bar{\bar{\theta}}}_j\rangle \). Based on analysis of QPE in Appendix A about the signs of eigenvalues, the operation here is feasible. Now we perform the inverse procedure of the transformation (3.4), then we obtain \( \sum_{j=1}^{n} \beta_j |0\rangle |u_j\rangle = \sum_{j=1}^{n} \beta_j |0\rangle |v_j\rangle \). Finally, apply Pauli-X to the first register to generate \( \sum_{j=1}^{n} \beta_j |0\rangle |v_j\rangle = \sum_{j=1}^{n} \beta_j e^{-2\pi i\theta_j} |0\rangle |u_j\rangle \). The complexity is determined by the implementation of the transformation (3.4), which equals \( O(T/\epsilon_1) \).

With the lemma, we state the quantum algorithm to implement (3.1) as follows:

Algorithm 2 Quantum eigenvalue estimation of normal matrices

Input: (1). An \( n \times n \) normal matrix \( M \) with (unknown) eigenvalues \( \{\sigma_1 e^{2\pi i\theta_1}, \ldots, \sigma_n e^{2\pi i\theta_n}\} \) and (unknown) eigenvectors \( \{|u_1\rangle, \ldots, |u_n\rangle\} \).

(2). A quantum state \(|b\rangle\) which formally equals \( \sum_{j=1}^{n} \beta_j |u_j\rangle \).

(3). The precisions \( \epsilon_1, \epsilon_2 \).

Output: The quantum state

\[
\sum_{j=1}^{n} \beta_j |0\rangle |u_j\rangle |\bar{\bar{\sigma}}_j\rangle |\bar{\bar{\bar{\theta}}}_j\rangle,
\]

where \( |\bar{\bar{\sigma}}_j - \sigma_j| \leq \epsilon_1 \) and \( |\bar{\bar{\bar{\theta}}}_j - \theta_j| \leq \epsilon_2 \) for all \( j \).

1: Apply Lemma 3.4 to construct \( W \) defined in equation (3.8).
2: Apply QPE (see Algorithm 3 in Appendix A) to \( W \) with initial state \(|0\rangle|b\rangle\).
3: Apply quantum SVD (see equation (3.4)) to estimate the singular values of \( M \).

In the following, we do more analysis about the step 2 and 3 of Algorithm 2.

Lemma 3.5. The state obtained in step 2 of Algorithm 2 is an approximation of

\[
\sum_{j=1}^{n} \beta_j |0\rangle |u_j\rangle |\bar{\bar{\sigma}}_j\rangle |\bar{\bar{\bar{\theta}}}_j\rangle.
\]

The complexity to obtain this state is \( O(T/\epsilon_1\epsilon_2) \).
Proof. This is a direct application of QPE. More precisely, to apply QPE, we first generate the following state by Hadamard transformation $|0\rangle|b\rangle \otimes \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |k\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{n} \sum_{k=0}^{m-1} \beta_j |0\rangle|u_j\rangle |k\rangle$, where $m$ is determined by the precision $\varepsilon_2$. Usually, it equals $O(1/\varepsilon_2)$. Then view $|k\rangle$ as a control qubit to apply $W^k$ to $|0\rangle|b\rangle$, we obtain $\frac{1}{\sqrt{m}} \sum_{j=1}^{n} \sum_{k=0}^{m-1} \beta_j e^{-2\pi i k j} |0\rangle|u_j\rangle |k\rangle$. Finally, apply quantum Fourier transform to $|k\rangle$ to generate $\frac{1}{m} \sum_{j=1}^{n} \sum_{l=0}^{m-1} \beta_j e^{2\pi i k j (\frac{l}{m} - \theta_j)} |0\rangle|u_j\rangle |l\rangle$. Similar to the analysis of QPE, the final state is an approximation of (3.10). The complexity analysis comes from QPE, which equals $O(T/\varepsilon_1\varepsilon_2)$.

In the state (3.10), we already obtain approximations of the phases. As for the singular values, we can apply the quantum singular value estimation technique, which is a simple application of the transformation (3.4). More precisely, apply (3.4) to the state (3.10), then we have $\sum_{j=1}^{n} \frac{\beta_j}{\sqrt{2}} (|w_j^+\rangle|\tilde{\sigma}_j| + |w_j^-\rangle|\tilde{\sigma}_j| - \tilde{\sigma}_j |\tilde{\sigma}_j|)$. Apply the oracle $|x\rangle|0\rangle \mapsto |x\rangle||x\rangle$ to the second register to obtain $\sum_{j=1}^{n} \frac{\beta_j}{\sqrt{2}} (|w_j^+\rangle|\tilde{\sigma}_j| + |w_j^-\rangle|\tilde{\sigma}_j| - \tilde{\sigma}_j |\tilde{\sigma}_j|)$. The implementation of this oracle is discussed in detail in Appendix A (see equation (A.10)). Finally, apply the inverse of (3.4) to yield the claimed state (3.9). The complexity is $O(T/\varepsilon_1)$. Therefore, the total cost of Algorithm 2 is $O(T/\varepsilon_1\varepsilon_2)$ as claimed.

4. Conclusions. In this paper, we proposed a quantum algorithm to compute the eigenvalue of normal matrices based on QPE and quantum SVD. We also proposed a quantum algorithm based on QPE and quantum linear solver to estimate the eigenvalues of diagonalizable matrices whose eigenvalues are real. The complexity is dominated by the complexity of the quantum linear algebraic techniques. Under certain conditions, we can obtain the quantum state that stores the eigenpairs in registers in polylog time. By measurements, the costs to obtain all the eigenvalues can be faster than any classical algorithm with a polynomial speedup.

Note that classically one reason leads to the importance of computing eigenvalues is that it can help solving differential equations. However, our results showed that the roles are reversed in the quantum case. Currently, based on quantum linear solvers, we can solve linear differential equations efficiently in a quantum computer [5,6,15]. A closely related problem is solving the integral equation (e.g. Volterra integral equation and Fredholm integral equation [3]). By discretization method, this kind of problem reduces to either a linear system or an eigenvalue problem. In the first case, we can apply quantum linear solvers. The algorithms proposed in this paper maybe helpful in solving the second type of problems.

Diagonalizability seems a necessary assumption to estimate the eigenvalues in the quantum computers. We hope the idea proposed in this paper can be generalized to estimate the eigenvalues of all diagonalizable matrices. As discussed in Subsection 2.2, Algorithm 1 works not so well in the general case. Thus new ideas are needed.

5. Acknowledgement. This work was supported by the QuantERA ERA-NET Cofund in Quantum Technologies implemented within the European Union’s Horizon 2020 Programme (QuantAlgo project), and EPSRC grants EP/L021005/1 and EP/R043957/1. No new data were created during this study.

Appendix A. Quantum phase estimation: brief overview. Quantum phase estimation (QPE) [33] is a useful technique to design quantum algorithms. Many quantum algorithms are related to it, such as Shor’s algorithm [43] and HHL algorithm to solve linear systems [27]. QPE is an algorithm to estimate the eigenvalues of unitary matrices. As a simple generalization, it can be used to estimate the eigenvalues of Hermitian matrices [1] or the singular values of general
matrices [27, 32]. In this paper, QPE is also an important technique we will apply to estimate the eigenvalues of more general matrices. Thus in the following, we briefly review the QPE to estimate the eigenvalues of Hermitian matrices, the error and complexity analyses can be found in [37, Section 5.2]. One aspect we want to emphasize is how to determine the signs of the eigenvalues.

A.1. Apply QPE to estimate the eigenvalues of Hermitian matrices. Let \( H \) be an \( n \times n \) Hermitian matrices with eigenvalues \( \lambda_1, \ldots, \lambda_n \) and eigenvectors \( |u_1, \ldots, u_n \rangle \). Let \( |b \rangle \) be any given quantum state, which can formally be written as a linear combinations of the eigenvectors \( |b \rangle = \sum_{k=1}^{n} \beta_k |u_k \rangle \). Suppose we know a upper bound of the eigenvalues, then choose a upper bound \( C > 0 \) such that \( |\lambda_j/C| < 1/2 \). Let \( \epsilon \) be the precision to approximate the eigenvalues and \( \delta \) be the failure probability of the quantum algorithm, denote \( q = \lceil \log 1/\epsilon \rceil + \lceil \log(2 + 1/2\delta) \rceil \) and \( Q = 2^q = O(1/\epsilon \delta) \), then QPE can be stated as follows:

\textbf{Algorithm 3} Quantum phase estimation (QPE)

\textbf{Input:} (1). An \( n \times n \) Hermitian matrix \( H \) with (unknown) eigenvalues \( \{\lambda_1, \ldots, \lambda_n\} \) and (unknown) eigenvectors \( \{|u_1, \ldots, u_n\rangle\} \).
(2). A quantum state \( |b \rangle \) which formally equals \( \sum_{k=1}^{n} \beta_k |u_k \rangle \).
(3). The precision \( \epsilon \) and failure probability \( \delta \).

\textbf{Output:} The quantum state \( \sum_{k=1}^{n} \beta_k |\tilde{\lambda}_k \rangle |u_k \rangle \), where \( |\tilde{\lambda}_k - \lambda_k| \leq \epsilon \) for all \( k \).

1: Set the initial state as

\( |\psi_0 \rangle = \frac{1}{\sqrt{Q}} \sum_{j=0}^{Q-1} |j \rangle |b \rangle \).

2: Apply control operator \( \sum_{j=0}^{Q-1} |j \rangle \langle j| \otimes e^{2\pi i j H/C} \) to \( |\psi_0 \rangle \) to prepare

\( |\psi_1 \rangle = \frac{1}{\sqrt{Q}} \sum_{j=0}^{Q-1} \sum_{k=1}^{n} \beta_k e^{2\pi i j \lambda_k / C} |j \rangle |u_k \rangle \).

3: Apply quantum inverse Fourier transform to the first register of \( |\psi_1 \rangle \)

\( |\psi_2 \rangle = \frac{1}{Q} \sum_{l=0}^{Q-1} \sum_{k=1}^{n} \beta_k Q^{-1} \sum_{j=0}^{Q-1} e^{2\pi i j (\frac{\lambda_k}{C} - \frac{l}{Q})} |l \rangle |u_k \rangle \).

Denote

\( \Lambda_k = \begin{cases} \{l \in \{0, 1, \ldots, Q-1\} : |\frac{\lambda_k}{C} - \frac{l}{Q}| \leq \epsilon\} & \text{if } \lambda_k \geq 0, \\
\{l \in \{0, 1, \ldots, Q-1\} : |1 + \frac{\lambda_k}{C} - \frac{l}{Q}| \leq \epsilon\} & \text{if } \lambda_k < 0. \end{cases} \)

\( |\Lambda_k \rangle = \frac{1}{Q} \sum_{l \in \Lambda_k} \sum_{j=0}^{Q-1} e^{2\pi i j (\frac{\lambda_k}{C} - \frac{l}{Q})} |l \rangle \).

In theory, we do not know \( \Lambda_k \). The notation introduced here is to simplify the expression of \(|\psi_2\rangle\). Then we can rewrite \(|\psi_2\rangle\) as

\[
|\psi_2\rangle = \sum_{k=1}^{n} \beta_k |\Lambda_k\rangle |u_k\rangle + \text{others.} \tag{A.7}
\]

It is shown in [37, Section 5.2] that the amplitude of the first term is larger than \( \sqrt{1-\delta} \). Thus, if we choose \( \delta \) small enough (e.g. 0.01 or \( 1/poly \log n \)), then we can approximately write \(|\psi_2\rangle\) as

\[
|\psi_2\rangle \approx \sum_{k=1}^{n} \beta_k |\Lambda_k\rangle |u_k\rangle. \tag{A.8}
\]

Any integer \( l \) in \( \Lambda_k \) provides an \( \epsilon \)-approximation \( l/Q \) of \( \lambda_k/C \) or \( 1 + \lambda_k/C \).

One problem we need pay attention in the above procedure is the signs of the eigenvalues. In\( \epsilon \)-unitary we determine the signs of the eigenvalues. In this case, the complexity of QPE is \( O(sC/\max \log \log 1/\epsilon) \log n \), which is optimal at the parameters \( t' \) and \( \epsilon' \). Here \( \max |H_{ij}| \). In this case, the complexity of QPE is \( O(sC/\max \log \log 1/\epsilon) \) if we set \( \epsilon' = \epsilon \).

In the following of this paper, we will simply write equation \( \text{(A.8)} \) as

\[
\sum_{k=1}^{n} \beta_k |\tilde{\Lambda}_k\rangle |u_k\rangle. \tag{A.9}
\]

Although this state is not rigorous especially when \( \lambda_k < 0 \), it clearly describes the result of QPE. It can be viewed as the quantum eigenvalue decomposition of Hermitian matrices. Moreover, for simplicity we will ignore \( \epsilon \) in the complexity analysis by setting it as a small constant.

In the end of this part, we give a method to implement the following transformation

\[
\sum_{k=1}^{n} \beta_k |\tilde{\Lambda}_k\rangle |u_k\rangle |0\rangle \mapsto \sum_{k=1}^{n} \beta_k |\tilde{\Lambda}_k\rangle |u_k\rangle |\tilde{\Lambda}_k\rangle, \tag{A.10}
\]

which will be useful in Section 3. Define the function \( f : \mathbb{Z}_Q \rightarrow \mathbb{Z}_Q \) by

\[
f(l) = \begin{cases} 
  l & \text{if } l \leq Q/2, \\
  Q - l & \text{if } l > Q/2.
\end{cases} \tag{A.11}
\]
It defines an oracle $U_f$ as follows: $U_f : |x, y⟩ → |x, y ⊕ f(x)⟩$. Based on this oracle, we can implement (A.10) via

$$
\sum_{k=1}^{n} \beta_k \sum_{l \in \Lambda_k} \Lambda_{kl} |l⟩|u_k⟩|0⟩ → \sum_{k=1}^{n} \beta_k \sum_{l \in \Lambda_k} \Lambda_{kl} |l⟩|u_k⟩|f(l)⟩.
$$

(A.12)

The equivalence comes naturally from the above analysis about the signs of eigenvalues. If $\lambda_k \geq 0$, then $|\lambda_k| = \lambda_k$ and $l \leq Q/2$ for all $l \in \Lambda_k$. Thus $f(l) = l$. If $\lambda_k < 0$, then $|\lambda_k| = -\lambda_k$ and $l > Q/2$ for all $l \in \Lambda_k$. Moreover, $(Q - l)/Q$ are approximations of $\lambda_k/C$ for all $l \in \Lambda_k$, thus $f(l) = Q - l$.

**A.2. Apply QPE to estimate the singular values of matrices.** For any matrix $M = (m_{ij})_{m \times n}$, in a quantum computer we can compute its singular value decomposition (SVD). More precisely, assume that its SVD is $M = \sum_{j=1}^{d} \sigma_j |u_j⟩⟨v_j|$, where $d = \min\{m, n\}$. Denote

$$
\tilde{M} = \begin{pmatrix}
0 & M \\
M^\dagger & 0
\end{pmatrix},
$$

(A.13)

which is Hermitian. The eigenvalues of $\tilde{M}$ are $\{±\sigma_j : j = 1, \ldots, d\}$. The corresponding eigenvectors are

$$
|w_j^±⟩ := \frac{1}{\sqrt{2}}(|0⟩|u_j⟩ ± |1⟩|v_j⟩), \quad j = 1, \ldots, d.
$$

(A.14)

Based on QPE, we can implement the following transformation (see equation (A.9))

$$
\sum_{j=1}^{d} \beta_j^+ |w_j^+⟩ + \beta_j^- |w_j^-⟩ \mapsto \sum_{j=1}^{d} \beta_j^+ |w_j^+⟩|\tilde{\sigma}_j⟩ + \beta_j^- |w_j^-⟩| - \tilde{\sigma}_j⟩
$$

(A.15)

in a quantum computer [11], where $|\sigma_j - \tilde{\sigma}_j| \leq \epsilon$. Based on the analysis about QPE, the minus sign in equation (A.15) is reasonable.

If $M$ is $s$ sparse, then we can implement (A.15) in cost $O(sC∥M∥_{\max} \log(m + n))/\epsilon)$. Finally, we conclude these two situations in the following proposition.

**Proposition A.1.** Let $M$ be an $m \times n$ matrix. Let $C$ be a upper bound of its singular values. If $M$ is $s$ sparse, then we can implement (A.15) in time $O(\epsilon^{-1} sC∥M∥_{\max} \log(m + n))$.

**Appendix B. Verification of equations (2.10)-(2.12).** Recall from (2.5), (2.9) that

$$
C_{m,k}(2\pi i M Δt) = \sum_{p=0}^{m(k+1)} |p⟩⟨p| \otimes I - \sum_{p=0}^{m-1} \sum_{k=0}^{k-1} |p(k+1) + q + 1⟩⟨p(k+1) + q| \otimes \frac{2\pi i M Δt}{q + 1}
$$

$$
- \sum_{p=0}^{m-1} \sum_{q=0}^{k} |p(k+1)(k+1)⟩⟨p(k+1) + q| \otimes I,
$$

$$
\hat{x} = \sum_{p=0}^{m-1} \sum_{q=0}^{k} |p(k+1) + q⟩|x_{p,q}⟩ + |m(k+1))|x_{m,0}⟩.
$$

Then

$$
C_{m,k}(2\pi i M Δt)|m(k+1))|x_{m,0}⟩ = |m(k+1))|x_{m,0}⟩.
$$
Moreover,

\[ C_{m,k}(2\pi i M \Delta t) \sum_{p=0}^{m-1} \sum_{q=0}^{k} |p(k+1)+q\rangle |x_{p,q}\rangle \]

\[ = \sum_{p=0}^{m-1} \sum_{q=0}^{k} |p(k+1)+q\rangle |x_{p,q}\rangle - \sum_{p=0}^{m-1} \sum_{q=1}^{k} |p(k+1)+q\rangle \otimes \frac{2\pi i M \Delta t}{q} |x_{p,q-1}\rangle \]

\[ - \sum_{p=0}^{m-1} |(p+1)(k+1)\rangle \sum_{q=0}^{k} |x_{p,q}\rangle \]

\[ = |0\rangle |x_{0,0}\rangle - m(k+1) \sum_{q=0}^{k} |x_{p,q}\rangle + \sum_{p=1}^{m-1} |p(k+1)\rangle \left( |x_{p,0}\rangle - \sum_{q=0}^{k} |x_{p-1,q}\rangle \right) \]

\[ + \sum_{p=0}^{m-1} \sum_{q=1}^{k} |p(k+1)+q\rangle \left( |x_{p,q}\rangle - \frac{2\pi i M \Delta t}{q} |x_{p,q-1}\rangle \right). \]

Consequently, \( |x_{0,0}\rangle = |E_j\rangle \). If \( 0 \leq p \leq m-1 \) and \( 1 \leq q \leq k \), then

\[ |x_{p,q}\rangle = \frac{2\pi i M \Delta t}{q} |x_{p,q-1}\rangle = \frac{(2\pi i M \Delta t)^q}{q!} |x_{p,0}\rangle. \]

If \( 0 \leq p \leq m \) and \( q = 0 \), then

\[ |x_{p,0}\rangle = \sum_{q=0}^{k} |x_{p-1,q}\rangle = \sum_{q=0}^{k} \frac{(2\pi i M \Delta t)^q}{q!} |x_{p-1,0}\rangle = T_k(2\pi i M \Delta t) |x_{p-1,0}\rangle. \]

**Appendix C. Proof of Lemma 2.9.** First we consider the special case of uniform distribution. Denote \( |\phi\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{n} |j\rangle \). This corresponds to a uniform distribution, and \( |j\rangle \) can be viewed as the \( j \)-th event. Consider \( |\phi\rangle \otimes m \). Perform measurements on the basis states, then the probability to obtain all the basis states such that all events happen is

\[ P := \frac{1}{n^m} \sum_{i_1+\cdots+i_n=m}^{i_1,\ldots,i_n \geq 1} \binom{m}{i_1,\ldots,i_n}, \]

where

\[ \binom{m}{i_1,\ldots,i_n} := \frac{m!}{i_1!\cdots i_n!} \]

is multinomial coefficient.

Let \( x_1,\ldots,x_n \) be \( n \) variables, then

\[ (x_1 + \cdots + x_n)^m = \sum_{i_1+\cdots+i_n=m}^{i_1,\ldots,i_n} \binom{m}{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}. \]

Especially when \( x_1 = \cdots = x_n = 1 \), we have

\[ n^m = \sum_{i_1+\cdots+i_n=m}^{i_1,\ldots,i_n} \binom{m}{i_1,\ldots,i_n}. \]
Then similar analysis shows that

\[ \binom{m}{i_1, \ldots, i_n} \geq n^m - n(n-1)^m. \]

Thus it suffices to choose \( m \). Denote \( p = \frac{m}{n} \).

The right hand side of the above inequality means that we set \( x_i = 0 \) and the remaining variables as \( 1 \) in \( (x_1 + \cdots + x_n)^m \). There are \( n \) possibilities. However, it may happen that different cases have common terms. Thus

\[ P \geq \frac{n^m - n(n-1)^m}{n^m} = 1 - n(1 - \frac{1}{n})^m \approx 1 - ne^{-m/n}. \]

Choose \( m \) such that \( ne^{-m/n} = \delta \) is small (say 0.01), that is \( m = n \log(n/\delta) \), then \( P \geq 1 - \delta \).

The above analyses show that in a uniform distribution, to make sure every events happen with probability close to 1, it suffices to make \( m = O(n \log n) \) experiments.

In the general case, assume that \( \langle \phi \rangle = \sum_{j=1}^{n} \sqrt{p_j} |j \rangle \), where \( p_1 + \cdots + p_n = 1 \). Also consider \( \langle \phi \rangle^\otimes m \), then

\[ P = \sum_{i_1 + \cdots + i_n = m} \binom{m}{i_1, \ldots, i_n} p_1^{i_1} \cdots p_n^{i_n}. \]

Denote \( p_{\max} = \max\{p_1, \ldots, p_n\} \). We assume that \( p_{\max} \neq 1 \), otherwise the distribution is trivial. Then similar analysis shows that

\[ P \geq 1 - \sum_{j=1}^{n} (1 - q_j)^m \geq 1 - n(1 - p_{\max})^m \approx 1 - ne^{-mp_{\max}}. \]

Thus it suffices to choose \( m = p_{\max}^{-1} \log(n/\delta) \). Since \( \sum_j q_j = 1 \), we have \( p_{\max} \geq 1/n \). Consequently, \( m \leq n \log(n/\delta) \).

REFERENCES

[1] D. S. Abrams and S. Lloyd, Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors, Phys. Rev. Lett., 83 (1999), p. 5162.
[2] A. Ambainis, Variable time amplitude amplification and quantum algorithms for linear algebra problems, in STACS’12 (29th Symposium on Theoretical Aspects of Computer Science), T. W. Christoph Dür, ed., vol. 14, Paris, France, Feb. 2012, LIPIcs, pp. 636–647, https://hal.archives-ouvertes.fr/hal-00678197.
[3] C. T. H. Baker, The Numerical Treatment of Integral Equations, Oxford University Press, Oxford, 1977.
[4] F. L. Bauer and C. T. Fike, Norms and exclusion theorems, Numer. Math., 2 (1960), pp. 137–141.
[5] C. M. Bender, Making sense of non-hermitian hamiltonians, Rep. Prog. Phys., 70 (2007), p. 947.
[6] D. W. Berry, High-order quantum algorithm for solving linear differential equations, J. Phys. A: Math. Theor., 47 (2014), p. 105301.
[7] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Quantum algorithm for linear differential equations with exponentially improved dependence on precision, Comm. Math. Phys., 356 (2017), pp. 1057–1081.
[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature, 549 (2017), pp. 195–202.
[9] A. Bunse-Gerstner and L. Elsner, Schur parameter pencils for the solution of the unitary eigenproblem, Linear Algebra Appl., 154 (1991), pp. 741–778.
[10] M. M. Cecchi and E. Di Nardo, The modified bordering method to evaluate eigenvalues and eigenvectors of normal matrices, Numerical Algorithms, 11 (1996), pp. 285–309.
[11] S. Chakraborty, A. Gilyén, and S. Jeffery, The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation, in 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019), C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, eds., vol. 132 of Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2019, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, pp. 33:1–33:14, https://doi.org/10.4230/LIPIcs.ICALP.2019.33.

[12] Y.-A. Chen and X.-S. Gao, Quantum algorithms for boolean equation solving and quantum algebraic attack on cryptosystems, arXiv preprint arXiv:1712.06239, (2017).

[13] Y.-A. Chen, X.-S. Gao, and C.-M. Yuan, Quantum algorithm for optimization and polynomial system solving over finite field and application to cryptanalysis, arXiv preprint arXiv:1802.03856, (2018).

[14] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM J. Comput., 46 (2017), pp. 1920–1950.

[15] A. M. Childs and J.-P. Liu, Quantum spectral methods for differential equations, arXiv preprint arXiv:1901.00961, (2019).

[16] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Information and Computation, (2012), pp. 901–924.

[17] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossning, Quantum machine learning: a classical perspective, P. Roy. Soc. A, 474 (2018), p. 20170551.

[18] E. S. Coakley and V. Rokhlin, A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices, Appl. Comput. Harmon. A., 34 (2013), pp. 379–414.

[19] A. Daskin, A. Grama, and S. Kais, A universal quantum circuit scheme for finding complex eigenvalues, Quantum Inf. Process., 13 (2014), pp. 333–353.

[20] M. P. Drazen and E. V. Haversworth, Criteria for the reality of matrix eigenvalues, Math. Z., 78 (1962), pp. 449–452.

[21] L. Elsner and K. D. Ikramov, Normal matrices: an update, Linear Algebra Appl., 285 (1998), pp. 291–303.

[22] M. Ferranti and R. Vandebriel, Computing eigenvalues of normal matrices via complex symmetric matrices, J. Comput. Appl. Math., 259 (2014), pp. 281–293.

[23] G. H. Golub and C. F. Van Loan, Matrix computations, Johns Hopkins University Press, Baltimore, 4 ed., 2013.

[24] W. B. Gragg, The qr algorithm for unitary hessenberg matrices, J. Comput. Appl. Math., 16 (1986), pp. 1–8.

[25] R. M. Gray, Toeplitz and circulant matrices: A review, Foundations and Trends® in Communications and Information Theory, 2 (2006), pp. 155–239.

[26] R. Grone, C. R. Johnson, E. M. Sa, and H. Wolkowicz, Normal matrices, Linear Algebra Appl., 87 (1987), pp. 213–225.

[27] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett., 103 (2009), p. 150502.

[28] S. Huss-Lederman, A. Tsao, and T. Turnbull, A parallelizable eigensolver for real diagonalizable matrices with real eigenvalues, SIAM J. Sci. Comput., 18 (1997), pp. 869–885.

[29] J. Kempe, A. Kitaev, and O. riggy, The complexity of the local hamiltonian problem, SIAM J. Comput., 35 (2006), pp. 1070–1097.

[30] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, q-means: A quantum algorithm for unsupervised machine learning, in Advances in Neural Information Processing Systems, 2019, pp. 4136–4146.

[31] I. Kerenidis and A. Prakash, Quantum gradient descent for linear systems and least squares, arXiv preprint arXiv:1704.04992, (2017).

[32] I. Kerenidis and A. Prakash, Quantum Recommendation Systems, in 8th Innovations in Theoretical Computer Science Conference (ITCS 2017), C. H. Papadimitriou, ed., vol. 67 of Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2017, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 49:1–49:21, https://doi.org/10.4230/LIPIcs.ITCS.2017.49.

[33] A. Y. Kitaev, Quantum measurements and the abelian stabilizer problem, arXiv preprint quant-ph/9510019, (1995).

[34] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nature Phys., 10 (2014), p. 631.

[35] G. H. Low and I. L. Chuang, Optimal hamiltonian simulation by quantum signal processing, Phys. Rev. Lett., 118 (2017), p. 010501.

[36] P. A. Macklin, Normal matrices for physicists, American J. Phys., 52 (1984), pp. 513–515.

[37] M. A. Nielsen and I. Chuang, Quantum computation and quantum information, Cambridge University Press, Cambridge, 2000.

[38] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification,
Phys. Rev. Lett., 113 (2014), p. 130503.

[39] Y. Saad, Numerical methods for large eigenvalue problems: revised edition, vol. 66, SIAM, 2011.

[40] M. Schuld, I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, Contemp. Phys., 56 (2015), pp. 172–185.

[41] C. Shao, Quantum speedup of training radial basis function networks, Quantum Information and Computation, 19 (2019), pp. 0609–0625.

[42] C. Shao, Quantum speedup of bayes’ classifiers, J. Phys. A: Math. Theor., 53 (2020), p. 045301.

[43] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput., 26 (1999), pp. 1484–1509.

[44] S. Subramanian, S. Brierley, and R. Jozsa, Implementing smooth functions of a hermitian matrix on a quantum computer, J. Phys. Commun., 3 (2019), p. 065002.

[45] H. Wang, L.-A. Wu, Y.-X. Liu, and F. Nori, Measurement-based quantum phase estimation algorithm for finding eigenvalues of non-unitary matrices, Phy. Rev. A, 82 (2010), p. 062303.

[46] T.-C. Wei, M. Mosca, and A. Nayak, Interacting boson problems can be qma hard, Phys. Rev. Lett., 104 (2010), p. 040501.

[47] N. Wiebe, D. Braun, and S. Lloyd, Quantum algorithm for data fitting, Phys. Rev. Lett., 109 (2012), p. 050505.