INDUSTRIAL AND SYSTEMS ENGINEERING

ISE

A Preconditioned Inexact Infeasible Quantum Interior Point Method for Linear Optimization

Zeguan Wu, Xiu Yang, and Tamás Terlaky

Department of Industrial and Systems Engineering, Lehigh University, USA

COR@L Technical Report 24T-017

LEHIGH UNIVERSITY

COR@L
COMPUTATIONAL OPTIMIZATION
RESEARCH AT LEHIGH

A Preconditioned Inexact Infeasible Quantum Interior Point Method for Linear Optimization

Zeguan Wu*[1], Xiu Yang†[1], and Tamás Terlaky‡[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, USA

Dec 11, 2024

Abstract

Quantum Interior Point Methods (QIPMs) have been attracting significant interests recently due to their potential of solving optimization problems substantially faster than state-of-the-art conventional algorithms. In general, QIPMs use Quantum Linear System Algorithms (QLSAs) to substitute classical linear system solvers. However, the performance of QLSAs depends on the condition numbers of the linear systems, which are typically proportional to the square of the reciprocal of the duality gap in QIPMs. To improve conditioning, a preconditioned inexact infeasible QIPM (II-QIPM) based on optimal partition estimation is developed in this work. We improve the condition number of the linear systems in II-QIPMs from quadratic dependence on the reciprocal of the duality gap to linear, and obtain better dependence with respect to the accuracy when compared to other II-QIPMs. Our method also attains better dependence with respect to the dimension when compared to other inexact infeasible Interior Point Methods.

1 Introduction

Linear Optimization (LO) is defined as optimizing a linear objective function under a set of linear constraints. The two most popular families of algorithms for solving LO problems are simplex algorithms and Interior Point Methods (IPMs) [3, 25]. Despite its efficiency for many practical problems, simplex methods may take exponentially many iterations to find an optimal solution, whereas IPMs guarantee an optimal solution in polynomial number of iterations.

The modern age of IPMs were launched by Karmarkar's projective method for LO problems [15]. Since then, many IPM variants have been proposed and studied for not only LO problems but also nonlinear optimization problems [25, 21, 22]. Contemporary IPMs start from an interior point and progress to the optimal set by moving within a neighbourhood of an analytic curve, known as the central path. Depending on whether feasibility is satisfied by the iterates, IPMs can be categorized into feasible IPMs and infeasible IPMs. Feasible IPMs start with a strictly feasible solution and maintain feasibility at each iterations, whereas infeasible IPMs does not require feasibility to be

*E-mail: `zew220@lehigh.edu`

†E-mail: `xiy518@lehigh.edu`

‡E-mail: `tat208@lehigh.edu`

exactly satisfied by any iterates. To find an ϵ-approximate solution for an LO problem with n variables and m constraints ($m \leq n$), feasible IPMs require $\mathcal{O}\left(\sqrt{n}\log(1/\epsilon)\right)$ IPM iterations while infeasible IPMs require $\mathcal{O}\left(n\log(1/\epsilon)\right)$ IPM iterations [23, 24].

In each IPM iteration, a Newton linear system needs to be solved for the search direction, which is called the Newton direction. The Newton linear system can take different forms, including full Newton system, augmented system, and normal equation system. Specifically, the full Newton system is asymmetric and can be solved exactly using Gaussian elimination with $\mathcal{O}\left(n^3\right)$ arithmetic operations. The augmented system is symmetric but indefinite, and it can be solved exactly using Bunch–Parlett factorization with $\mathcal{O}\left(n^3\right)$ arithmetic operations [4]. The normal equation system is symmetric positive definite and can be solved exactly using Cholesky decomposition with $\mathcal{O}\left(n^3\right)$ arithmetic operations. These systems can also be solved inexactly using methods including Krylov subspace methods, which may require fewer iterations if the target accuracy is moderate. However, the Newton directions obtained in this way are inexact, which may make the IPM iterates infeasible. Thus, the resulting IPMs are infeasible IPMs and have worse iteration complexity. On the other hand, it is possible to maintain feasibility when using inexact Newton system solutions. For example, in [18, 28, 2], the authors proposed an orthogonal subspace system whose inexact solutions give feasible Newton directions.

The recent development of quantum computing has triggered the study of novel quantum algorithms, including hybrid quantum-classic algorithms [9]. Many hybrid algorithms use Quantum Linear System Algorithms (QLSAs) as they can solve some linear systems faster than classical linear system algorithms [13, 7, 6]. For example, based on the seminal work of the QLSA [13], various approaches that improve the performance of IPMs by solving the Newton systems with QLSAs are proposed [17, 19, 2]. Although QLSAs are more efficient w.r.t. dimension, there are several issues when applying them to IPMs. One issue is that QLSAs only produce quantum states, which cannot be used by classical IPMs directly. So QLSAs are always coupled with quantum state tomography algorithms (QTAs) [11], which extract classical results from the quantum results. At each iteration of these quantum IPMs (QIPMs), QLSAs are used to solve quantized Newton systems and QTAs are used to obtain classical solutions. Here, QTAs have polynomial complexity with respect to the reciprocal of their precisions. Also, such precisions should be of the same magnitude as the precision of QIPMs. As a result, although many QIPMs attain better complexities with respect to problem size than their classical counterparts, they could have worse complexity with respect to precision. Another issue with QLSAs is their linear (or worse) complexity dependence on the condition number of Newton systems. The condition number of the Newton systems arising from IPMs can increase to infinity when IPMs approach optimality. Despite the improved dependence on the problem size, the dependence on condition number indicates the necessity to precondition the Newton systems arising from QIPMs.

In this work, we propose a preconditioned II-QIPM using the preconditioning method proposed by Chai and Toh in [5] and the II-QIPM framework proposed by Mohammadisiahroudi et al in [19]. We prove the condition number of the preconditioned linear system is improved from $\mathcal{O}\left(1/\mu^2\right)$ to $\mathcal{O}\left(1/\mu\right)$, where μ is the corresponding central path parameter. Our algorithm demonstrates $\mathcal{O}(1/\epsilon^3)$ speed-up compared with the II-QIPM proposed in [19] when the iterative refinement method is not used.

2 Preliminaries

In this section, we introduce the notation used in this paper. Then we give a brief introduction to classical inexact infeasible IPMs (II-IOMs) for LO problems. Finally, we introduce the II-QIPM for LO proposed in [19].

2.1 Notation

In this work, we take the following conventions. Vectors are denoted by lowercase letters, matrices by capital letters. For a vector $v \in \mathbb{R}^n$, its ith element is denoted by $v_{[i]}$, i.e., $v = \left(v_{[1]}, \ldots, v_{[n]}\right)^T$. For a vector with subscript, e.g., $v_1 \in \mathbb{R}^n$, its ith element is denoted by $v_{1,[i]}$. For a diagonal matrix $M \in \mathbb{R}^{n\times n}$, its ith diagonal element is denoted by $M_{[i]}$. For a diagonal matrix with subscript, e.g., $M_1 \in \mathbb{R}^{n\times n}$, its ith diagonal element is denoted by $M_{1,[i]}$.

For a matrix $M \in \mathbb{R}^{m\times n}$ with $m \leq n$, it has m singular values. We assume that the singular values are sorted in non-increasing order starting from index 1. The singular values of M are denoted by

$$\sigma_1(M) \geq \sigma_2(M) \geq \cdots \geq \sigma_m(M),$$

except for the special notation $\sigma_0(M)$ denoting the smallest nonzero singular value of M. For a matrix $M \in \mathbb{R}^{m\times n}$ $(m \leq n)$ with k $(k < m)$ nonzero singular values, we have

$$\sigma_1(M) \geq \cdots \geq \sigma_k(M) = \sigma_0(M) > \sigma_{k+1}(M) = \cdots = \sigma_m(M) = 0.$$

The condition number of matrix M is denoted by $\kappa(M)$ and is defined as

$$\kappa(M) = \frac{\sigma_1(M)}{\sigma_0(M)},$$

which implies that, in this work, the condition number for any matrix, except zero matrices, is always finite. For a nonsingular matrix $M \in \mathbb{R}^{(n+m)\times(n+m)}$ with the following 2-2 block structure

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

where $M_{11} \in \mathbb{R}^{n\times n}$ and $M_{22} \in \mathbb{R}^{m\times m}$ and both nonsingular, we denote the Schur complements [8] of M by

$$(M/M_{11}) = M_{22} - M_{21}M_{11}^{-1}M_{12} \text{ and } (M/M_{22}) = M_{11} - M_{12}M_{22}^{-1}M_{21}.$$

We use e_n and 0_n to denote the n-dimensional all one vector and all zero vector, respectively. When the dimension is obvious from the context, we may use e and 0 for simplicity. We use $I_{n\times m}$ and $0_{n\times m}$ to denote the $n \times m$ dimensional identity matrix and zero matrix, respectively. When the dimension is obvious from the context, we may write I or 0. We take the convention of big-O, big-Omega, and big-Theta notations for complexity. We also use ˜ above these notations when ignoring the polylogarithmic overhead. For example, $\tilde{\mathcal{O}}_n(n^2) = \mathcal{O}(n^2 \text{poly}\log(n))$.

2.2 Inexact Infeasible IPMs for LO Problems

In this section, we briefly review II-IPMs for LO problems. In this work, an LO problem is defined as follows.

Definition 1 (LO Problem). *For vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and matrix $A \in \mathbb{R}^{m \times n}$ with* $\operatorname{rank}(A) = m \leq n$, *we define the primal LO problem as:*

$$\min_{x \in \mathbb{R}^n} \; c^T x, \text{ s.t. } Ax = b, \; x \geq 0 \tag{P}$$

and the dual LO problem as

$$\max_{y \in \mathbb{R}^m, s \in \mathbb{R}^n} \; b^T y, \text{ s.t. } A^T y + s = c, \; s \geq 0. \tag{D}$$

We make the following assumptions on problems (P) and (D).

Assumption 1. *The input data, i.e., entries of b, c, and A, are integers.*

Assumption 2. *There exists a solution (x, y, s) such that*

$$Ax = b, \; A^T y + s = c, \; \text{and } (x, s) > 0.$$

Assumption 1 implicitly exists in most IPMs work. In this paper, it is explicitly used to prove Theorem 1. Assumption 2 is not restrictive because we can apply the self-dual embedding model [25] to problem (P) and (D) to obtain an equivalent LO problem which has a strictly feasible solution. The set of primal-dual feasible solutions can be defined as

$$\mathcal{PD} = \left\{ (x, y, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n : \; Ax = b, \; A^T y + s = c, \; (x, s) \geq 0 \right\}$$

and the set of interior primal-dual feasible solutions can be defined as

$$\mathcal{PD}^0 = \left\{ (x, y, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n : \; Ax = b, \; A^T y + s = c, \; (x, s) > 0 \right\}.$$

According to the Strong Duality Theorem [25], the set of optimal solutions can be defined as

$$\mathcal{PD}^* = \left\{ (x, y, s) \in \mathcal{PD} : XSe = 0 \right\},$$

where $X = \operatorname{diag}(x)$ and $S = \operatorname{diag}(s)$. Next, let $\epsilon > 0$, and the set of ϵ-approximate solutions to problem (P) and (D) can be defined as

$$\mathcal{PD}_\epsilon = \left\{ (x, y, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n : x^T s \leq n\epsilon, \; \left\| (r_p, r_d) \right\|_2 \leq \epsilon \right\},$$

where $r_p = b - Ax$, $r_d = c - A^T y - s$. Under Assumption 2, for any $\mu > 0$, the perturbed optimality conditions

$$\begin{aligned} Ax = b, \; x \geq 0 \\ A^T y + s = c, \; s \geq 0 \\ XSe = \mu e, \end{aligned} \tag{1}$$

have a unique solution $(x(\mu), y(\mu), s(\mu))$ that defines the primal and dual central path

$$\mathcal{CP} = \left\{ (x, y, s) \in \mathcal{PD}^0 \mid XSe = \mu e, \; \forall \mu > 0 \right\}.$$

IPMs apply Newton's method to solve system (1) approximately. At each iteration of inexact infeasible IPMs (II-IPMs), a candidate solution to problem (P) and (D) is updated by moving along the solution to the full Newton system (NS), i.e., by solving

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}, \tag{NS}$$

where $r_c = \beta_1 \mu e - XSe$ and $\beta_1 \in (0,1)$ is the central path parameter reduction factor. Here, $(r_p,\ r_d,\ r_c)$ are defined at each iteration but for simplicity here we omit the superscript for iteration index. (NS) can be reduced into the so-called augmented system (AS)

$$\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{dc} \\ r_p \end{bmatrix}, \tag{AS}$$

where $D = X^{-1}S$ and $r_{dc} = r_d - X^{-1}r_c$. (AS) can be further reduced into the normal equation system (NES)

$$AD^{-1}A^T \Delta y = AD^{-1} r_{dc} + r_p. \tag{NES}$$

In each II-IPM iterations, a linear solver is used to inexactly solve for the Newton direction $(\Delta x, \Delta y, \Delta s)$. Depending on which linear system is solved and which linear solver is used, the requirements and properties of the inexact Newton directions might be different [10, 16, 1]. If one uses any inexact method to solve (NS), then all the three residual terms will be nonzero, i.e., $A\Delta x - r_p \neq 0, A^T\Delta y + \Delta s - r_d \neq 0, S\Delta x + X\Delta s - r_c \neq 0$. These nonzero residual terms will bring more challenges in the analysis of II-IPMs. In [1, 20], the authors used similar techniques to eliminate the residuals for the first two equations in (NS) and to move all the residuals into the third equation. This is helpful for convergence analysis and we discuss this in more details in the next section. Once an inexact Newton direction is obtained, the candidate solution is updated by moving along the Newton direction while staying in a neighbourhood of the central path. In this work, for any $0 < \gamma_1 < 1$ and $1 \leq \gamma_2$, we define a central path neighborhood as

$$\mathcal{N}(\gamma_1, \gamma_2) = \left\{ (x, y, s) \in \mathbb{R}^{n+m+n} :\ (x, s) \geq 0,\ xs \geq \gamma_1 \mu e,\ \|(r_p, r_d)\|_2 \leq \gamma_2 \mu \right\}, \tag{2}$$

where $\mu = x^T s / n$.

2.3 Inexact Infeasible QIPMs for LO Problems

QIPMs use quantum algorithms to solve Newton linear systems. Due to the non-negligible inaccuracy in current quantum computers, the Newton directions obtained from quantum algorithms are inexact and thus might be infeasible. In [19], the authors designed and analyzed the first II-QIPM for LO problems. They followed the general framework of the II-IPM proposed in [1] and used quantum algorithms to obtain the Newton directions. Instead, if one solves (NES) using quantum algorithms to get Δy and sets

$$\begin{aligned} \Delta s &= r_d - A^T \Delta y \\ \Delta x &= S^{-1}\left(r_c - X\Delta s\right), \end{aligned}$$

then the second and third equation in (NS) are satisfied exactly while the residual of the first equation in (NS) equals to the residual of (NES). It was recognized in [1, 10] that one can move the residual from the first equation to the third, which makes the first two equations in (NS) be satisfied exactly and makes the analysis easier. To do so, one first needs a basis for A. In [19], the authors claimed that the cost to find such a basis is dominated by the cost of solving (NS) and thus ignored. In this work, for simplicity and w.l.o.g., we assume such a basis is known.

Assumption 3. *Matrix A is of the form $A = \begin{bmatrix} A_B & A_N \end{bmatrix}$ with $A_B \in \mathbb{R}^{m\times m}$ being nonsingular and the condition number of A_B being no larger than that of A.*

In fact, Assumption 3 is not restrictive because we can rewrite problem (P) into the canonical form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} c^T x \\ &\text{s.t.} \begin{bmatrix} A \\ -A \end{bmatrix} \begin{bmatrix} x \\ x \end{bmatrix} \leq \begin{bmatrix} b \\ -b \end{bmatrix} \\ &\qquad x \geq 0. \end{aligned}$$

Then, we can apply the self-dual embedding model to the canonical form and obtain another standard form LO problem with both Assumption 2 and 3 satisfied. See [18] for details.

With Assumption 3, we partition D correspondingly as

$$D = \begin{bmatrix} D_B & 0 \\ 0 & D_N \end{bmatrix},$$

where D_B is an $m \times m$ matrix and D_N is an $(n-m) \times (n-m)$ diagonal matrix. We also define $\hat{A} = A_B^{-1} A$ and $\hat{b} = A_B^{-1} b$. In the general case where A_B is not trivial, this one-time computation contributes $\mathcal{O}(mn^2)$ arithmetic operations to total complexity, which can also be avoided if one uses the aforementioned canonical form reformulation and applies the self-dual embedding model as described in [18]. Following [19], we can rewrite (NES) as the following modified normal equation system (MNES)

$$\left(D_B^{\frac{1}{2}}\hat{A}\right) D^{-1} \left(D_B^{\frac{1}{2}}\hat{A}\right)^T z = \left(D_B^{\frac{1}{2}} A_B^{-1}\right)\left(AD^{-1} r_{dc} + r_p\right). \tag{MNES}$$

We introduce the following notations:

$$\begin{aligned} M_{\text{NES}} &= AD^{-1}A^T \\ M_B &= \left(D_B^{\frac{1}{2}} A_B^{-1}\right)^T \\ M_{\text{MNES}} &= M_B^T M_{\text{NES}} M_B \\ v_{\text{MNES}} &= M_B^T \left(AD^{-1} r_{dc} + r_p\right). \end{aligned}$$

Δy and z satisfies the bijective relationship: $\Delta y = M_B z$. In [19], the authors introduced the following procedure to obtain inexact Newton directions.

Step 1. Find z such that $\|r\|_2 \leq \eta\sqrt{\mu/n}$, where $\eta \in (0,1)$ and

$$r = M_{\text{MNES}} z - v_{\text{MNES}}. \tag{3}$$

Step 2. Calculate

$$\Delta y = M_B z. \tag{4}$$

Step 3. Calculate $\nu = \begin{bmatrix} D_B^{-\frac{1}{2}} r \\ 0 \end{bmatrix} \in \mathbb{R}^n$.

Step 4. Calculate $\Delta s = r_d - A^T \Delta y$ and $\Delta x = S^{-1}\left(r_c - X\Delta s\right) - \nu$.

We are not using this procedure to calculate Newton steps, but introducing this procedure helps us determine the convergence condition of our proposed algorithm. For the Newton directions obtained in this way, the residual of (NS) only shows up in its third equation as $X\Delta s + S\Delta x - r_c = -S\nu$. To get the II-QIPM converge, the authors in [19] proposed the stopping criteria

$$\|S\nu\|_\infty \le \eta\mu.$$

They proved the following lemma.

Lemma 2.1 (Lemma 4.2 in [19]). *If $\|r\|_2 \le \eta\sqrt{\mu/n}$, then $\|S\nu\|_\infty \le \eta\mu$.*

To solve (MNES) to the desired accuracy above, i.e., $\|r\|_2 \le \eta\sqrt{\mu/n}$, the authors in [19] analyzed the accuracy required from the quantum algorithm. Specifically, they used the QLSA by [6] and the QTA by [27]. Before introducing their conclusion on the required accuracy for quantum algorithm, we first compare the different meanings of accuracy between classical linear system solves and quantum linear system solves. Different from classical solvers, QLSAs compute a quantum state representing the normalized vector of the classical solution and store the quantum state in the quantum machine. We use QTAs to get a classical normalized vector close to the quantum state and then rescale it properly. Therefore, we take into account the precision of QTAs in the analysis.

Now we discuss the accuracy needed for the quantum algorithms. Let us consider (NES). When we use classical algorithms to compute an ϵ_{c}-approximate solution Δy, we get $\Delta\hat{y}$ such that

$$\epsilon_c = \left\|M_{\mathrm{NES}}\Delta\hat{y} - AD^{-1}r_{dc} - r_p\right\|_2 = \|M_{\mathrm{NES}}(\Delta\hat{y} - \Delta y)\|_2 .$$

According to the definition of (MNES) and the definition of r, see equation (3), it is obvious that

$$\epsilon_{\mathrm{c}} = \|M_{\mathrm{NES}}(\Delta\hat{y} - \Delta y)\|_2 = \left\|\left(M_B^T\right)^{-1} r\right\|_2 . \tag{5}$$

In contrast, when we use QLSAs and QTAs to solve (NES) with accuracy ϵ_{q}, we get a classical unit vector $\Delta\tilde{y}$ with accuracy ϵ_{q} such that

$$\mathcal{E}_q = \frac{\Delta\tilde{y}}{\|\Delta\tilde{y}\|_2} - \frac{\Delta y}{\|\Delta y\|_2} = \Delta\tilde{y} - \frac{\Delta y}{\|\Delta y\|_2} \quad \text{with } \|\mathcal{E}_q\|_2 = \epsilon_{\mathrm{q}}. \tag{6}$$

The norm of the actual solution Δy can be retrieved inexactly using quantum algorithms [6]. But this approach introduces extra inexactness to the algorithm. Instead, we consider rescaling $\Delta\tilde{y}$ to minimize the ℓ_2 norm of the residual of (MNES). Denote the rescaling factor by $l_{\Delta\tilde{y}}$. The minimization problem is a univariate convex quadratic optimization problem with unique optimal solution

$$l^*_{\Delta\tilde{y}} = \arg\min_{l_{\Delta\tilde{y}}\in\mathbb{R}} \left\|M_{\mathrm{MNES}}M_B^{-1}l_{\Delta\tilde{y}}\Delta\tilde{y} - v_{\mathrm{MNES}}\right\|_2 .$$

With this rescaling choice, one can show the resulting residual r_{MNES} satisfies

$$\begin{aligned}
\|r_{\mathrm{MNES}}\|_2 &\le \left\|M_{\mathrm{MNES}}M_B^{-1}\|\Delta y\|_2\Delta\tilde{y} - v_{\mathrm{MNES}}\right\|_2 \\
&= \left\|M_{\mathrm{MNES}}M_B^{-1}(\Delta y + \|\Delta y\|_2\mathcal{E}_q) - v_{\mathrm{MNES}}\right\|_2 \\
&= \left\|M_{\mathrm{MNES}}M_B^{-1}\|\Delta y\|_2\mathcal{E}_q\right\|_2 \\
&= \left\|M_{\mathrm{MNES}}M_B^{-1}\mathcal{E}_q\right\|_2\|\Delta y\|_2 \\
&= \left\|M_{\mathrm{MNES}}M_B^{-1}\mathcal{E}_q\right\|_2\left\|\left(M_{\mathrm{MNES}}M_B^{-1}\right)^{-1}v_{\mathrm{MNES}}\right\|_2 \\
&\le \kappa_{M_{\mathrm{MNES}}M_B^{-1}}\|v_{\mathrm{MNES}}\|_2\epsilon_q.
\end{aligned}$$

Combining the inequality and Lemma 2.1, we get the following choice of accuracy

$$\epsilon_{\text{QLSA}} = \epsilon_{\text{QTA}} = \epsilon_q = \frac{\eta}{\kappa_{M_{\text{MNES}} M_B^{-1}} \|v_{\text{MNES}}\|_2} \sqrt{\frac{\mu}{n}}. \tag{7}$$

After obtaining the solution from QLSA and QTA, one updates the solution by taking a step along the Newton direction with the step length introduced in [19]. Before we provide the pseudocode for the II-QIPM proposed in [19], we define the following quantities:

$$\begin{aligned} \omega^* &= \max_{\mathcal{PD}^*} \|(x,\ 0 \times y,\ s)\|_\infty, \\ \omega^k &= \|(x^k,\ 0 \times y^k,\ s^k)\|_\infty, \\ \bar{\omega} &= \max_{k=1,\dots,\bar{k}} \omega^k, \end{aligned} \tag{8}$$

where $(x^k,\ y^k,\ s^k)$ is the iterate in the kth iteration and $\bar{k}$ is the total number of iterations. In some sections later, the superscript for iteration index might be ignored. In those context, we simply use ω for ω^k, i.e., $\omega = \omega^k$. The pseudocode for the II-QIPM proposed in [19] is provided in Algorithm 1.

Algorithm 1 II-QIPM [19]

```
Choose $\epsilon > 0$, $\gamma_1 \in (0,1)$, $\gamma_2 > 0$, $0 < \eta < \beta_1 < \beta_2 < 1$,
$k \leftarrow 0$, $(x^0, y^0, s^0) \leftarrow (\omega^* e, 0e, \omega^* e)$, and $\gamma_2 \leftarrow \max\left\{1, \frac{\|(r_p^0, r_d^0)\|_2}{\mu^0}\right\}$,
**while** $(x^k, y^k, s^k) \notin \mathcal{PD}_\epsilon$ **do**
    $\mu^k \leftarrow \frac{(x^k)^T s^k}{n}$
    set $\epsilon^k_{\text{QLSA}}$ and $\epsilon^k_{\text{QTA}}$ using (7),
    $(\Delta x^k, \Delta y^k, \Delta s^k) \leftarrow$ **solve** (MNES)($\beta_1$) by QLSA+QTA
                                    with precisions $\epsilon^k_{\text{QLSA}}$ and $\epsilon^k_{\text{QTA}}$, respectively,
    $\hat{\alpha}^k \leftarrow \max\Big\{\bar{\alpha} \in [0,1] \mid$ for all $\alpha \in [0, \bar{\alpha}]$ we have
                    $\big((x^k, y^k, s^k) + \alpha(\Delta x^k, \Delta y^k, \Delta s^k)\big) \in \mathcal{N}(\gamma_1, \gamma_2)$ and
                    $(x^k + \alpha \Delta x^k)^T (s^k + \alpha \Delta s^k) \leq \big(1 - \alpha(1 - \beta_2)\big)(x^k)^T s^k\Big\}$,
    $(x^{k+1}, y^{k+1}, s^{k+1}) \leftarrow (x^k, y^k, s^k) + \hat{\alpha}^k (\Delta x^k, \Delta y^k, \Delta s^k)$,
    **if** $\|x^{k+1}, s^{k+1}\|_\infty > \omega^*$ **then**
        **return** Primal or dual is infeasible.
    **end if**
    $k \leftarrow k + 1$
**end while**
**return** $(x^k, y^k, s^k)$
```

While II-QIPMs have better dependence on the dimension of the problem, they have linear dependence on the condition number of the coefficient matrix of (MNES), which in the worst case can be larger than the condition number of the coefficient matrix of (NES). It is known that the condition number of normal equation systems can go to infinity for LO problems [12]. In this work, we are interested in investigating how to precondition the linear systems arising from II-QIPMs and analyzing how preconditioning can improve the complexity of II-QIPMs.

3 Preconditioned Quantum Interior Point Method

In this section, we propose a new QIPM using the preconditioning method proposed by Chai and Toh in [5], analyze the condition number of the preconditioned linear system, and prove the complexity of our preconditioned II-QIPM. The preconditioning method we present here is a special adaption of the method proposed in [5]. Then, we discuss how to use this method in the QIPM setting and analyze the effect of the proposed preconditioning. Finally, we discuss the complexity of the proposed preconditioned QIPM.

3.1 A Special Adaptation & Analysis of Chai & Toh's Method

As explained in [5], when IPMs approach optimality with duality gap μ, according to the optimal partition of the problem, the diagonal entries in D will go into two clusters, one is of magnitude $\mathcal{O}(\mu)$ and the other $\mathcal{O}(1/\mu)$. Before the optimal partition is revealed, one can predict the optimal partition according to some partition rules and permute the matrix D into the following form:

$$D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix},$$

where D_1 and D_2 are both diagonal matrices and the diagonal elements of D_1 and D_2 are $\mathcal{O}(\mu)$ and $\mathcal{O}(1/\mu)$ respectively. Subsequently, one can partition all the related objects in the same manner, i.e.,

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$$

and

$$\Delta x = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}, \quad r_{dc} = \begin{bmatrix} (r_{dc})_1 \\ (r_{dc})_2 \end{bmatrix},$$

etc. These matrices and vectors are redefined[1] at each iteration, but for simplicity we have dropped the iteration index superscripts here and in the remaining of this section.

With the aforementioned partition, it is shown in [5] that (AS) can be transformed into another linear system as specified in the following lemma.

Lemma 3.1 (Lemma 4.1 in [5][2]). *The solution of* (AS) *can be computed from the following reduced augmented system (RAS):*

$$\begin{bmatrix} H & B \\ B^T & -D_1 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta \tilde{x}_1 \end{bmatrix} = \begin{bmatrix} h \\ F_1^{-1/2}(r_{dc})_1 \end{bmatrix}, \tag{RAS}$$

where $F_1 = I + D_1$, and

$$\begin{aligned} \Delta \tilde{x}_1 &= F_1^{-1/2} \Delta x_1, \\ B &= A_1 F_1^{-1/2}, \\ H &= A \operatorname{diag}(F_1^{-1}, D_2^{-1}) A^T = BB^T + A_2 D_2^{-1} A_2^T, \\ h &= r_p + A \operatorname{diag}(F_1^{-1}, D_2^{-1}) r_{dc}. \end{aligned}$$

[1] Ultimately, when the optimal partition is correctly identified, Δx_1 and Δx_2 correspond to the index partition in the optimal partition.

[2] Lemma 3.1 is a special case of Lemma 4.1 in [5] when their matrix E_1 is the identity matrix.

In this work, we take the following partition rule:

Definition 2. *If $x_{[i]} \geq \sqrt{[(1-\gamma_1)n+\gamma_1]\mu}$, then i is included in the index set defining partition indicated by subscript "1"; else, i is included in the partition indicated by "2".*

Lemma 3.2. *Under the partition rule in Definition 2, the value of $D_{1,[i]}$ lies in the interval $[\frac{\gamma_1\mu}{\omega^2}, 1]$; the value of $F_{1,[i]}$ lies in the interval $[\frac{\gamma_1\mu}{\omega^2}+1, 2]$; the value of $D_{2,[j]}$ lies in the interval $[\frac{\gamma_1}{(1-\gamma_1)n+\gamma_1}, \frac{\omega^2}{\gamma_1\mu}]$.*

Proof. From the definition of the neighbourhood (2), we have

$$x_{[i]}s_{[i]} \geq \gamma_1\mu, \text{ for all } i \in \{1,\dots,n\},$$

so

$$x_{[i]}s_{[i]} = x^Ts - \sum_{j\neq i} x_{[j]}s_{[j]} \leq n\mu - \gamma_1(n-1)\mu.$$

From the first inequality we have

$$D_{1,[i]} = x_{1,[i]}^{-1}s_{1,[i]} = x_{1,[i]}s_{1,[i]}/x_{1,[i]}^2 \geq \frac{\gamma_1\mu}{x_{1,[i]}^2} \geq \frac{\gamma_1\mu}{\omega^2},$$

and from the second inequality and Definition 2 we derive

$$D_{1,[i]} = x_{1,[i]}^{-1}s_{1,[i]} = x_{1,[i]}s_{1,[i]}/x_{1,[i]}^2 \leq \frac{n\mu-\gamma_1(n-1)\mu}{x_{1,[i]}^2} \leq 1.$$

Similarly, we can obtain the conclusion for $F_{1,[i]}$ since $F_{1,[i]} = D_{1,[i]}+1$. As for $D_{2,[j]}$, from the definition of the neighbourhood $\mathcal{N}(\gamma_1,\gamma_2)$, see (2), we have

$$D_{2,[j]} = x_{2,[j]}^{-1}s_{2,[j]} = (x_{2,[j]}s_{2,[j]})^{-1}s_{2,[j]}^2 \leq \frac{s_{2,[j]}^2}{\gamma_1\mu} \leq \frac{\omega^2}{\gamma_1\mu}$$

and

$$D_{2,[j]} = x_{2,[j]}^{-1}s_{2,[j]} = \frac{x_{2,[j]}s_{2,[j]}}{x_{2,[j]}^2} \geq \frac{\gamma_1\mu}{x_{2,[j]}^2} \geq \frac{\gamma_1}{(1-\gamma_1)n+\gamma_1},$$

that completes the proof. □

Denote the coefficient matrix of (RAS) by K, i.e.,

$$K = \begin{bmatrix} H & B \\ B^T & -D_1 \end{bmatrix}.$$

It is proven in [5] that K, H and D_1 are all nonsingular. So one can use the Schur complement to write out the inverse matrix of K as

$$K^{-1} = \begin{bmatrix} H^{-1} - H^{-1}BZ^{-1}B^TH^{-1} & H^{-1}BZ^{-1} \\ Z^{-1}B^TH^{-1} & -Z^{-1} \end{bmatrix},$$

where $Z = -(K/H) = B^T H^{-1} B + D_1$. This naturally leads to the preconditioner with the following block structure

$$P_c^{-1} = \begin{bmatrix} \hat{H}^{-1} - \hat{H}^{-1} B \hat{Z}^{-1} B^T \hat{H}^{-1} & \hat{H}^{-1} B \hat{Z}^{-1} \\ \hat{Z}^{-1} B^T \hat{H}^{-1} & -\hat{Z}^{-1} \end{bmatrix},$$

where $\hat{H}$ is a selected positive definite matrix and $\hat{Z}$ is an approximation of Z by using $\hat{H}$ as the approximation of H. The preconditioned (RAS) system, that we refer to as P-RAS, is

$$P_c^{-1} K = P_c^{-1} \begin{bmatrix} h \\ F_1^{-1/2} (r_{dc})_1 \end{bmatrix}. \tag{P-RAS}$$

The coefficient matrix of (P-RAS) is

$$P_c^{-1} K = \begin{bmatrix} \hat{H}^{-1} H - \hat{H}^{-1} B \hat{Z}^{-1} B^T (\hat{H}^{-1} H - I) & \hat{H}^{-1} B \hat{Z}^{-1} (\hat{Z} - B^T \hat{H}^{-1} B - D_1) \\ \hat{Z}^{-1} B^T (\hat{H}^{-1} H - I) & \hat{Z}^{-1} (B^T \hat{H}^{-1} B + D_1) \end{bmatrix},$$

and the right-hand-side vector is

$$\begin{bmatrix} \xi' \\ \xi'' \end{bmatrix} = P_c^{-1} \begin{bmatrix} h \\ F_1^{-\frac{1}{2}} (r_{dc})_1 \end{bmatrix} = \begin{bmatrix} \hat{H}^{-1} h - \hat{H}^{-1} B \hat{Z}^{-1} B^T \hat{H}^{-1} h + \hat{H}^{-1} B \hat{Z}^{-1} F_1^{-\frac{1}{2}} (r_{dc})_1 \\ \hat{Z}^{-1} B^T \hat{H}^{-1} h - \hat{Z}^{-1} F_1^{-\frac{1}{2}} (r_{dc})_1 \end{bmatrix}.$$

In this work, we choose

$$\begin{aligned} \hat{H} &= \hat{h} I \\ \hat{h} &= \gamma_1 \sigma_0^2(A) / \omega^2 \\ \hat{Z} &= B^T \hat{H}^{-1} B + D_1. \end{aligned}$$

Then the coefficient matrix of (P-RAS) can be simplified to

$$P_c^{-1} K = \begin{bmatrix} \hat{H}^{-1} H + \hat{H}^{-1} B \hat{Z}^{-1} B^T - \hat{H}^{-1} B \hat{Z}^{-1} B^T \hat{H}^{-1} H & 0 \\ \hat{Z}^{-1} B^T (\hat{H}^{-1} H - I) & I \end{bmatrix}.$$

Take the following notations:

$$\begin{aligned} (P_c^{-1} K)_{11} &= \hat{H}^{-1} H + \hat{H}^{-1} B \hat{Z}^{-1} B^T - \hat{H}^{-1} B \hat{Z}^{-1} B^T \hat{H}^{-1} H \\ (P_c^{-1} K)_{21} &= \hat{Z}^{-1} B^T (\hat{H}^{-1} H - I). \end{aligned}$$

Then (P-RAS) can be simplified as

$$\begin{aligned} (P_c^{-1} K)_{11} \Delta y &= \xi' \\ (P_c^{-1} K)_{21} \Delta y + \Delta \tilde{x}_1 &= \xi''. \end{aligned}$$

Notice that we only need the first equation to compute Δy. So we call the normalized first equation of (P-RAS) as reduced (P-RAS), that we refer to as (RP-RAS), i.e.,

$$\Xi \Delta y = \xi, \tag{RP-RAS}$$

where

$$\Xi = \frac{(P_c^{-1} K)_{11}}{\|(P_c^{-1} K)_{11}\|_2}, \ \xi = \frac{\xi'}{\|(P_c^{-1} K)_{11}\|_2}.$$

We build and solve (RP-RAS) instead of (MNES) using the QLSA in [6] and the QTA in [27]. We discuss how to build and solve (RP-RAS) in Section 3.3. Before that, we analyze (RP-RAS) and determine the target accuracy for the quantum subroutine in the next section. The pseudocode for preconditioned II-QIPM is provided here.

Algorithm 2 Preconditioned II-QIPM

```
Choose $\epsilon > 0$, $\gamma_1 \in (0,1)$, $\gamma_2 > 0$, $0 < \eta < \beta_1 < \beta_2 < 1$,
$k \leftarrow 0$, $(x^0, y^0, s^0) \leftarrow (\omega^* e, 0e, \omega^* e)$, and $\gamma_2 \leftarrow \max\left\{1, \frac{\|(r_p^0, r_d^0)\|_2}{\mu^0}\right\}$,
while $(x^k, y^k, s^k) \notin \mathcal{PD}_\epsilon$ do
    $\mu^k \leftarrow \frac{(x^k)^T s^k}{n}$
    Partition according to Definition 2
    set $\epsilon^k_{\text{QLSA}}$ and $\epsilon^k_{\text{QTA}}$ using (7)
    $(\Delta x^k, \Delta y^k, \Delta s^k) \leftarrow$ build and solve (RP-RAS) by QLSA+QTA
                              with precision $\epsilon^k_{\text{QLSA}}$ and $\epsilon^k_{\text{QTA}}$
    $\hat{\alpha}^k \leftarrow \max\Big\{\bar{\alpha} \in [0,1] \mid$ for all $\alpha \in [0, \bar{\alpha}]$ we have
                                $\big((x^k, y^k, s^k) + \alpha(\Delta x^k, \Delta y^k, \Delta s^k)\big) \in \mathcal{N}(\gamma_1, \gamma_2)$ and
                                $(x^k + \alpha\Delta x^k)^T (s^k + \alpha \Delta s^k) \le \big(1 - \alpha(1-\beta_2)\big)(x^k)^T s^k\Big\}$
    $(x^{k+1}, y^{k+1}, s^{k+1}) \leftarrow (x^k, y^k, s^k) + \hat{\alpha}^k(\Delta x^k, \Delta y^k, \Delta s^k)$
    if $\|x_{k+1}, s_{k+1}\|_\infty > \omega^*$ then
        return Primal or dual is infeasible.
    end if
    $k \leftarrow k + 1$
end while
return $(x^k, y^k, s^k)$
```

3.2 Condition Number Analysis

In this section, we analyze the condition number of the coefficient matrix of (RP-RAS), or more precisely, the condition numbers of matrix $\hat{Z}$ and Ξ. To do so, we define

$$G = \hat{H}^{-1/2} B \hat{Z}^{-1} B^T \hat{H}^{-1/2}$$

and

$$Y = G + (I - G)\hat{H}^{-1/2} H \hat{H}^{-1/2}.$$

By construction, $Y = (P_c^{-1} K)_{11}$. We first show that the smallest nonzero singular value of matrix G is bounded from below by a constant.

Lemma 3.3. *The smallest nonzero singular value of G satisfies $\sigma_0(G) \ge \frac{1}{1 + 2\hat{h}\sigma_0^{-2}(A_1)}$.*

Proof. Let $J = B^T \hat{H}^{-\frac{1}{2}}$ and denote the singular value decomposition (SVD) of J by $J = Q\Sigma P^T = Q_0 \Sigma_0 P_0^T$, where the first SVD is the full SVD and the second one is the reduced SVD. Then matrix G can be expressed using J and D_1,

$$G = J^T (J J^T + D_1)^{-1} J = P \Sigma^T (\Sigma\Sigma^T + Q^T D_1 Q)^{-1} \Sigma P^T.$$

Let $G' = \Sigma\Sigma^T + Q^T D_1 Q$. Then G' and G'^{-1} can be expressed as

$$G' = \begin{bmatrix} \Sigma_0^2 + \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \text{ and } G'^{-1} = \begin{bmatrix} (G'^{-1})_{11} & (G'^{-1})_{12} \\ (G'^{-1})_{21} & (G'^{-1})_{22} \end{bmatrix}, \tag{9}$$

where Φ_{ij} are submatrices of the matrix $\Phi = Q^T D_1 Q$ and $(G'^{-1})_{11} = (\Sigma_0^2 + (\Phi/\Phi_{22}))^{-1}$. So matrix G can be expressed as

$$G = P \begin{bmatrix} (I + \Sigma_0^{-1}(\Phi/\Phi_{22})\Sigma_0^{-1})^{-1} & 0 \\ 0 & 0 \end{bmatrix} P^T \tag{10}$$

and the singular values of matrix G are the same as the singular values of the matrix in the middle of the right-hand-side expression. By Lemma A.5, the nonzero singular values of the matrix in the middle equals to the singular values of $\left(I + \Sigma_0^{-1}\left(\Phi/\Phi_{22}\right)\Sigma_0^{-1}\right)^{-1}$, i.e., the reciprocal of the result of the singular values of $\Sigma_0^{-1}\left(\Phi/\Phi_{22}\right)\Sigma_0^{-1}$ plus 1. By Lemma A.2, the singular values of $\Sigma_0^{-1}(\Phi/\Phi_{22})\Sigma_0^{-1}$ are bounded from above by the largest singular value of $\Sigma_0^{-1}\Phi\Sigma_0^{-1}$. Then, by using Lemma A.3, we have

$$\begin{aligned} \sigma_1(\Sigma_0^{-1}\Phi\Sigma_0^{-1}) &\leq \sigma_1(\Sigma_0^{-1})\sigma_1(\Phi)\sigma_1(\Sigma_0^{-1}) \\ &= [\sigma_0(\Sigma_0)]^{-2}\sigma_1(D_1) \\ &= [\sigma_0(F_1^{-\frac{1}{2}}A_1^T)]^{-2}\hat{h}\sigma_1(D_1) \\ &\leq [\sigma_0(F_1^{-\frac{1}{2}})\sigma_0(A_1^T)]^{-2}\hat{h}\sigma_1(D_1) \\ &= \hat{h}\sigma_0^{-2}(A_1)\sigma_1(F_1)\sigma_1(D_1) \\ &\leq 2\hat{h}\sigma_0^{-2}(A_1), \end{aligned}$$

where the second inequality follows from Lemma A.4 and the last inequality follows from Lemma 3.2. Thus

$$\sigma_0(G) \geq \frac{1}{1 + 2\hat{h}\sigma_0^{-2}(A_1)}. \tag{11}$$

The proof is complete. □

Lemma 3.4. *The largest singular value of G satisfies $\sigma_1(G) \leq \left(1 + \frac{\hat{h}\gamma_1\mu}{\omega^2}\sigma_1^{-2}(A_1)\right)^{-1}$.*

Proof. Similar to the proof for Lemma 3.3, we have $\sigma_1(G) \leq \left(1 + \sigma_0(\Sigma_0^{-1}\Phi\Sigma_0^{-1})\right)^{-1}$. Apply Lemma A.4 twice, then it follows that

$$\begin{aligned} \sigma_0(\Sigma_0^{-1}\Phi\Sigma_0^{-1}) &\geq \sigma_0(\Sigma_0^{-1})\sigma_0(\Phi)\sigma_0(\Sigma_0^{-1}) \\ &= [\sigma_1(\Sigma_0)]^{-2}\sigma_0(D_1) \\ &= [\sigma_1(F_1^{-\frac{1}{2}}A_1^T)]^{-2}\hat{h}\sigma_0(D_1) \\ &\geq [\sigma_1(F_1^{-\frac{1}{2}})\sigma_1(A_1^T)]^{-2}\hat{h}\sigma_0(D_1) \\ &= \hat{h}\sigma_1^{-2}(A_1)\sigma_0(F_1)\sigma_0(D_1) \\ &\geq \frac{\hat{h}\gamma_1\mu}{\omega^2}\sigma_1^{-2}(A_1), \end{aligned}$$

that completes the proof. □

With the spectral properties of matrix G, we then analyze matrix Y. For simplicity of the analysis we denote the SVD of matrix G as $G = R\Psi R^T = R_0 \Psi_0 R_0^T$, where the former SVD is the full SVD and the latter one is the reduced SVD. To study the properties of matrix Y, it is worth mentioning that Y can be represented in the following way

$$\begin{aligned} R^T Y R &= R^T G R + R^T (I - G)\frac{1}{\hat{h}} H R \\ &= R^T G R + R^T (I - G) R R^T \frac{1}{\hat{h}} H R \\ &= \Psi + (I - \Psi)\,\frac{1}{\hat{h}} R^T H R \\ &= \begin{bmatrix} \Psi_0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix} \mathcal{A} \\ &= \begin{bmatrix} \Psi_0 + \mu(I - \Psi_0) & \\ & \mu I \end{bmatrix} + \begin{bmatrix} I - \Psi_0 & \\ & I \end{bmatrix} (\mathcal{A} - \mu I), \end{aligned} \tag{12}$$

where

$$\mathcal{A} = \frac{1}{\hat{h}} R^T H R.$$

We present the spectral properties of matrix $\mathcal{A}$ in the following lemma and then we study the spectral properties of matrix Y.

Lemma 3.5. *The smallest nonzero singular value of $\mathcal{A}$ satisfies $\sigma_0(\mathcal{A}) \geq \mu$ and the largest singular value of $\mathcal{A}$ satisfies $\sigma_1(\mathcal{A}) \leq \frac{(1-\gamma_1)n+\gamma_1}{\gamma_1 \hat{h}} \sigma_1^2(A)$.*

Proof. Denote $\begin{bmatrix} F_1^{-\frac{1}{2}} & 0 \\ 0 & D_2^{-\frac{1}{2}} \end{bmatrix}$ by M_{FD}. By Lemma A.4, it follows that

$$\sigma_0(\mathcal{A}) = \frac{1}{\hat{h}} \sigma_0(A M_{FD})^2 \geq \frac{1}{\hat{h}} \sigma_0(A)^2 \sigma_0(M_{FD})^2 \geq \frac{\gamma_1 \mu}{\hat{h} \omega^2} \sigma_0(A)^2 \geq \mu,$$

where the second inequality follows from Lemma 3.2. Similarly, by Lemma A.3 and Lemma 3.2,

$$\sigma_1(\mathcal{A}) \leq \frac{(1-\gamma_1)n + \gamma_1}{\gamma_1 \hat{h}} \sigma_1^2(A),$$

that completes the proof. □

Now we are ready to study the condition number of Y.

Lemma 3.6. *Using the partition rule in Definition 2, the singular values of Y satisfy $\sigma_0(Y) = \Omega\left(\mu\right)$ and*

$$\sigma_1(Y) \leq 1 + \frac{(1-\gamma_1)n + \gamma_1}{\gamma_1 \hat{h}} \sigma_1^2(A),$$

thus the condition number of Y satisfies

$$\kappa(Y) = \mathcal{O}\left(\frac{1}{\mu}\left(1 + \frac{(1-\gamma_1)n + \gamma_1}{\gamma_1 \hat{h}} \sigma_1^2(A)\right)\right).$$

Proof. Notice that

$$
\begin{aligned}
&\begin{bmatrix} \Psi_0 + \mu(I - \Psi_0) & 0 \\ 0 & \mu I \end{bmatrix} + \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix} (\mathcal{A} - \mu I) \\
= &\left(\left(\begin{bmatrix} \Psi_0 + \mu(I - \Psi_0) & 0 \\ 0 & \mu I \end{bmatrix} + \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix} (\mathcal{A} - \mu I) \right) \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix} \right) \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix}^{-1}.
\end{aligned}
$$

The matrix in the outer parentheses is nonsingular because it is the summation of a positive diagonal matrix and a positive semidefinite matrix. The matrix outside the parentheses is positive diagonal. Applying Lemma A.4, it follows that

$$
\begin{aligned}
\sigma_0(Y) \geq \sigma_0 &\left(\left(\begin{bmatrix} \Psi_0 + \mu(I - \Psi_0) & 0 \\ 0 & \mu I \end{bmatrix} + \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix} (\mathcal{A} - \mu I) \right) \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix} \right) \\
&\times \sigma_0 \left(\begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix}^{-1} \right) \\
\geq \sigma_0 &\left(\begin{bmatrix} \Psi_0 + \mu(I - \Psi_0) & 0 \\ 0 & \mu I \end{bmatrix} \begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix} \right) \sigma_0 \left(\begin{bmatrix} I - \Psi_0 & 0 \\ 0 & I \end{bmatrix}^{-1} \right),
\end{aligned}
$$

where the second inequality follows from the fact that, the first matrix in the first line is the summation of a positive diagonal matrix and a positive semidefinite matrix, and thus we only keep the positive diagonal one. Then, using Lemma 3.4 and Lemma 3.3, we can get

$$
\sigma_0(Y) = \min \left\{ \sigma_0(G)(1 - \sigma_1(G)) + \mu(1 - \sigma_1(G))^2, \ \mu \right\} \times 1 = \Omega\left(\mu\right).
$$

Recall that (MNES) is equivalent to (NES), thus Y has no zero singular value, so $\sigma_m(Y) = \sigma_0(Y)$. As for the largest singular value, we have the following bound:

$$
\sigma_1(Y) \leq 1 + 1 \times (\sigma_1(\mathcal{A}) - \mu) \leq 1 + \frac{(1 - \gamma_1)n + \gamma_1}{\gamma_1 \hat{h}} \sigma_1^2(A).
$$

The proof is complete. □

We have proved that the condition number of the coefficient matrix of (MNES) is $\mathcal{O}(1/\mu)$. However, when constructing the coefficient matrix, the inverse matrix of $\hat{Z}$ is also needed. Thus the condition number of $\hat{Z}$ also matters.

Lemma 3.7. *The singular values of $\hat{Z}$ satisfy $\sigma_0(\hat{Z}) \geq \gamma_1 \mu / \omega^2$ and $\sigma_1(\hat{Z}) \leq 1 + \frac{1}{\hat{h}} \sigma_1^2(A_1)$; the condition number of $\hat{Z}$ satisfies $\kappa(\hat{Z}) \leq \frac{1}{\mu} \frac{\omega^2}{\gamma_1} (1 + \frac{1}{\hat{h}} \sigma_1(A)^2)$.*

Proof. By the definition of $\hat{Z}$ that $\hat{Z} = B \hat{H}^{-1} B^T + D_1$ and Lemma 3.2, the smallest nonzero singular value of $\hat{Z}$ satisfies $\sigma_0(\hat{Z}) \geq 0 + \sigma_0(D_1) \geq \gamma_1 \mu / \omega^2$. The largest singular value of $\hat{Z}$ satisfies

$$
\begin{aligned}
\sigma_1(\hat{Z}) &\leq \frac{1}{\hat{h}} \sigma_1(BB^T) + \sigma_1(D_1) \\
&\leq \frac{1}{\hat{h}} \sigma_1(A_1)^2 \sigma_0(F_1)^{-1} + \sigma_1(D_1) \\
&\leq \frac{1}{\hat{h}} \sigma_1(A_1)^2 + 1,
\end{aligned}
$$

where the first inequality holds because both terms in $\hat{Z}$ are positive definite, the second inequality follows from Lemma A.3, and the third follows from Lemma 3.2. Combining the above with $\sigma_1(A_1) \leq \sigma_1(A)$, we have the claimed result. □

In the sequel we use κ_Y and $\kappa(Y)$ (resp. $\kappa_{\hat{Z}}$ and $\kappa(\hat{Z})$) interchangeably. In the next section, we describe how to use QLSAs and QTAs to solve (RP-RAS).

3.3 Solve (RP-RAS) w. QLSAs & QTAs

In this work, we follow the II-QIPM introduced in [19] and use the QLSA proposed in [11] as well as the QTA proposed in [27] to solve (RP-RAS). In this section, we discuss the block-encoding of (RP-RAS). We assume that we have access to quantum RAM (QRAM) and have the initial data stored in QRAM. For those operations that contribute polylogarithmic overhead, we say those operations can be implemented *efficiently* without spelling out their complexity. We also assume we can efficiently construct diagonal matrices from vectors with the diagonal entries being powers of the entries of vectors. For example, if we have a vector x in QRAM, then we can efficiently construct X and $X^{-1/2}$ in QRAM. Finally, we assume that we can efficiently construct submatrices of any matrix stored in QRAM.

Provided access to QRAM, the complexity associated with block-encoding the involved matrices and preparing a quantum state encoding of the right-hand-side amounts to polylogarithmic overhead, which is dominated by the cost of the block-encoding of negative powers of matrices and that of QLSAs and QTAs. Thus, we ignore the cost of the block-encoding of matrices here and, in turn, we can ignore the accuracy parameter of block-encoding in our analysis – we keep the accuracy parameters of block-encoding in our analysis but we do not analyze their value.

We start with the block-encoding of B. Recall that the definition of B is provided in Lemma 3.1 as $B = A_1 F_1^{-1/2}$. We have the following lemma.

Lemma 3.8 (Block-encodings of A and A_1). *An $(\|A\|_F, \log(n) + 2, \epsilon_A)$-block-encoding of A and an $(\|A_1\|_F, \log(n) + 2, \epsilon_{A_1})$-block-encoding of A_1 can be implemented efficiently.*

Proof. The results follow directly from the two assumptions we just made earlier in this section and Lemma 50 from [11]. □

The following lemma indicates that the block-encoding of a diagonal matrix with entries bounded between −1 and 1 can be implemented easily.

Lemma 3.9 (Block encode diagonal matrices). *Let $M \in \mathbb{R}^{n\times n}$ and $\tilde{M} \in \mathbb{R}^{n\times n}$ be two diagonal matrices and let us assume that for all $i \in \{1, \ldots, n\}$, the following conditions hold:*

$$M_{[i]} \in [-1, 1],\ \tilde{M}_{[i]} \in [-1, 1],\ M_{[i]}^2 + \tilde{M}_{[i]}^2 = 1.$$

Then, a $(1, \mathcal{O}(\log(n)), 0)$-block-encoding of M can be implemented efficiently[3].

Proof. By construction,

$$\begin{bmatrix} M & \tilde{M} \\ \tilde{M} & -M \end{bmatrix} \begin{bmatrix} M & \tilde{M} \\ \tilde{M} & -M \end{bmatrix}^T = \begin{bmatrix} M^2 + \tilde{M}^2 & 0 \\ 0 & M^2 + \tilde{M}^2 \end{bmatrix} = I.$$

Then according Lemma 50 from [11] the conclusion holds. □

[3] Note that a (p_1, p_2, p_3)-block-encoding can be implemented efficiently indicates that the cost of the block-encoding is polylogarithmic in terms of p_1, p_2, and p_3.

Based on this lemma, we can have the following corollary.

Corollary 3.1. *Let $M \in \mathbb{R}^{n\times n}$ be a diagonal matrix and $\|M\|_{\max}$ be the maximum of the absolute value of the entries of M. Then, an $(\alpha_M, \mathcal{O}(\log(n)), 0)$-block-encoding of M can be implemented efficiently, where $\alpha_M \geq \|M\|_{\max}$.*

Then we can efficiently implement the block-encoding of diagonal matrices with proper scaling. Now we discuss the block-encoding of matrix B.

Lemma 3.10. *An*

$$\left(\|A_1\|_F \|F_1^{-1/2}\|_{\max}, \mathcal{O}(\log(n)), \epsilon_B\right)$$

-block-encoding of matrix B can be implemented efficiently.

Proof. Using Lemma 3.8, we can implement an

$$(\|A_1\|_F, \log(n_1) + 2, \epsilon_{A_1})$$

-block-encoding of A_1 efficiently. From Corollary 3.1, we can implement an

$$\left(\|F_1^{-1/2}\|_{\max}, \mathcal{O}(\log(n)), 0\right)$$

-block-encoding of $F_1^{-1/2}$ efficiently. Then, by Lemma 50 of [11], we can implement an

$$\left(\|A_1\|_F \|F_1^{-1/2}\|_{\max}, \mathcal{O}(\log(n)), \epsilon_B\right)$$

-block-encoding of B efficiently. □

Now we are ready to discuss the block-encoding of matrix H that is defined in Lemma 3.1 as $H = A \operatorname{diag}\left(F_1^{-1}, D_2^{-1}\right) A^T$.

Lemma 3.11. *An*

$$\left(\|A\|_F^2 \max\{\|F_1^{-1}\|_{\max}, \|D_2^{-1}\|_{\max}\}, \mathcal{O}(\log(n)), \epsilon_H\right)$$

-block-encoding of matrix H can be implemented efficiently.

Proof. From Corollary 3.1, we can implement a

$$\left(\max\{\|F_1^{-1}\|_{\max}, \|D_2^{-1}\|_{\max}\}, \mathcal{O}(\log(n)), 0\right)$$

-block-encoding of $\operatorname{diag}\left(F_1^{-1}, D_2^{-1}\right)$ efficiently. Following Lemma 53 of [11] and Lemma 3.8, we can implement a

$$\left(\|A\|_F^2 \max\{\|F_1^{-1}\|_{\max}, \|D_2^{-1}\|_{\max}\}, \mathcal{O}(\log(n)), \epsilon_H\right)$$

-block-encoding of matrix H efficiently. □

Then we discuss the block-encoding of matrix $\hat{Z} = B^T \hat{H}^{-1} B + D_1 = \frac{1}{\hat{h}} B^T B + D_1$.

Lemma 3.12. *An*

$$\left(\alpha_H \|A_1\|_F^2 \|F_1^{-1/2}\|_{\max}^2, \mathcal{O}(\log(n)), \epsilon_{\hat{Z}}\right)$$

-block-encoding of $\hat{Z}$ can be implemented efficiently, where $\alpha_H = \mathcal{O}\left(1/\hat{h}\right)$.

Proof. From Lemma 53 of [11] and Lemma 3.10, we can implement an

$$\left(\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2, \mathcal{O}(\log(n)), \epsilon_{B^TB}\right)$$

-block-encoding of B^TB efficiently. From Lemma 3.2, we have $\|D_1\|_{\max} \leq 1$. Then from Corollary 3.1, we can implement an

$$\left(\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2, \mathcal{O}(\log(n)), 0\right)$$

-block-encoding of $\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2 D_1$ efficiently. From Lemma 52 of [11], we can implement an

$$\left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2, \mathcal{O}(\log(n)), \epsilon_{\hat{Z}}\right)$$

-block-encoding of $\hat{Z}$ efficiently, where

$$\alpha_H = 1/\hat{h} + 1/\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2 = \mathcal{O}\left(1/\hat{h}\right).$$

The last equation follows from Lemma 3.2. □

Then we discuss the block-encoding of $\hat{Z}^{-1}$.

Lemma 3.13. *A*

$$\left(2/\|\hat{Z}\|_2, \mathcal{O}\left(\operatorname{poly}\log(n) + \operatorname{poly}\log\kappa_{\hat{Z}}\right), \epsilon_{\hat{Z}^{-1}}\right)$$

-block-encoding of $\hat{Z}^{-1}$ can be implemented with cost

$$\tilde{\mathcal{O}}_{n,\epsilon}\left(\kappa_{\hat{Z}}\left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2\right).$$

Proof. By applying Lemma 52 of [11] to $\hat{Z}$, we can efficiently implement an

$$\left(\left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2, \mathcal{O}(\log(n)), \left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2\epsilon_{\hat{Z}}\right)$$

-block-encoding of $\hat{Z}/\|\hat{Z}\|_2$. By applying Lemma 10 of [6] to $\hat{Z}/\|\hat{Z}\|_2$, a

$$\left(2, \mathcal{O}\left(\operatorname{poly}\log(n) + \operatorname{poly}\log\kappa_{\hat{Z}}\right), \epsilon_{\|\hat{Z}\|_2\hat{Z}^{-1}}\right)$$

-block-encoding of $\|\hat{Z}\|_2\hat{Z}^{-1}$ can be implemented with cost

$$\tilde{O}_{n,\epsilon}\left(\kappa_{\hat{Z}}\left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2\right).$$

Then the result follows from Lemma 52 of [11]. □

Then we discuss the block-encoding of $B\left(\hat{Z}^{-1}\right)B^T$.

Lemma 3.14. *A*

$$\left(2\left(\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2, \mathcal{O}\left(\operatorname{poly}\log(n) + \operatorname{poly}\log\kappa_{\hat{Z}}\right), \epsilon_{BZB}\right)$$

-block-encoding of $B\left(\hat{Z}^{-1}\right)B^T$ can be implemented efficiently.

Proof. The result follows from Lemma 3.10 and Lemma 3.13 and Lemma 53 of [11]. □

Then we discuss the block-encoding $H + B\hat{Z}^{-1}B^T$.

Lemma 3.15. *A*

$$\begin{pmatrix} 2\left(\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2 + \|A\|_F^2 \max\{\|F_1^{-1}\|_{\max}, \|D_2^{-1}\|_{\max}\}, \\ \mathcal{O}\left(\operatorname{poly}\log(n) + \operatorname{poly}\log\kappa_{\hat{Z}}\right), \\ \epsilon_{HBZB} \end{pmatrix}$$

-block-encoding of $H + B\hat{Z}^{-1}B^T$ can be implemented efficiently.

Proof. The result follows from Lemma 3.11, Lemma 3.14, and Lemma 52 of [11]. □

Lemma 3.16. *A*

$$\begin{pmatrix} 2\left(\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)\|A\|_F^2 \max\{\|F_1^{-1}\|_{\max}, \|D_2^{-1}\|_{\max}\}/\|\hat{Z}\|_2, \\ \mathcal{O}\left(\{\operatorname{poly}\log(n) + \operatorname{poly}\log\kappa_{\hat{Z}}\right), \\ \epsilon_{BZBH} \end{pmatrix}$$

-block-encoding of $B\hat{Z}^{-1}B^T H$ can be implemented efficiently.

Proof. The result follows from Lemma 3.11, Lemma 3.14, and Lemma 53 of [11]. □

Finally, we have the block-encoding of (RP-RAS).

Lemma 3.17. *Let*

$$C_{\text{RPRAS}} = \frac{\left(2\hat{h}\frac{\left(\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)}{\|\hat{Z}\|_2} + 1\right)\left(\|A\|_F^2 \max\{\|F_1^{-1}\|_{\max}, \|D_2^{-1}\|_{\max}\} + 1\right) - 1}{\hat{h}^2\left\|\left(P_c^{-1}K\right)_{11}\right\|_2}.$$

Then a

$$\left(C_{\text{RPRAS}}, \mathcal{O}\left(\operatorname{poly}\log(n) + \operatorname{poly}\log\kappa_{\hat{Z}}\right), \epsilon_{\text{RPRAS}}\right)$$

-block-encoding of Ξ can be implemented with $\tilde{O}_{n,\epsilon}\left(\kappa_{\hat{Z}}\left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2\right)$ queries to QRAM.

Proof. The result follows from Lemma 3.15, Lemma 3.16 and Lemma 52 of [11]. □

We have the following lemma for the value of C_{RPRAS}.

Lemma 3.18. $C_{RPRAS} = \mathcal{O}\left(\frac{\|A\|_F^4}{\hat{h}\left\|\hat{Z}\right\|_2\|Y\|_2}\right)$.

Proof. From Lemma 3.2, we have

$$\left\|F_1^{-1/2}\right\|_{\max}^2 = \mathcal{O}(1)$$

and

$$\max\{\|F_1^{-1}\|_{\max}, \|D_2^{-1}\|_{\max}\} = \mathcal{O}(1).$$

Then

$$C_{RPRAS} = \mathcal{O}\left(\frac{\|A\|_F^4}{\hat{h}\left\|\hat{Z}\right\|_2\left\|\left(P_c^{-1}K\right)_{11}\right\|_2}\right) = \mathcal{O}\left(\frac{\|A\|_F^4}{\hat{h}\left\|\hat{Z}\right\|_2\|Y\|_2}\right).$$

The proof is complete. □

3.4 Worst-case Complexity

In this section, we analyze the complexity of Algorithm 2. In each iteration of Algorithm 2, we use Definition 2 to partition the variables. We use the QLSA of [6] to build and solve (RP-RAS), use the QTA from [27] to readout the solution, and then use line search to update the iterates. The complexity is dominated by the complexity of QLSA and QTA.

Theorem 1. *In the k^{th} iteration of Algorithm 2, the QLSA by [6] and the QTA by [27] can build and solve* (RP-RAS) *with a solution satisfying $\|\hat{r}^k\|_2 \leq \eta\sqrt{\mu^k/n}$ with*

$$\tilde{\mathcal{O}}_{n,\bar{\omega},\frac{1}{\epsilon}}\left(\frac{n^{1.5}\bar{\omega}^{13}}{\epsilon^6}\kappa_A^5\|A\|_F^6\left(\left\|\hat{A}\right\|_2+\left\|\hat{b}\right\|_2\right)\right)$$

queries to QRAM. The total complexity for Algorithm 2 to generate an ϵ-approximate solution is

$$\tilde{\mathcal{O}}_{n,\bar{\omega},\frac{1}{\epsilon}}\left(n^2\cdot\left(\frac{n^{1.5}\bar{\omega}^{13}}{\epsilon^6}\kappa_A^5\|A\|_F^6\left(\left\|\hat{A}\right\|_2+\left\|\hat{b}\right\|_2\right)\right)\right)$$

queries to QRAM and $\tilde{\mathcal{O}}_{n,\bar{\omega},\frac{1}{\epsilon}}(n^3m)$ classical arithmetic operations.

Proof. Let T_{RPRAS} be the cost of implementing the block-encoding of Ξ as in Lemma 3.17. Then, we have

$$T_{\text{RPRAS}} = \tilde{\mathcal{O}}_{n,\epsilon}\left(\kappa_{\hat{Z}}\left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2\right).$$

Let T_ξ be the cost of implementing state-preparation of ξ in equation (RP-RAS) as introduced in [6]. Then, the complexity for QLSA + QTA, that we denote by T_{iter}, is

$$T_{\text{iter}} = \tilde{\mathcal{O}}_{n,\kappa_{\hat{Z}},\kappa_Y}\left(\kappa_Y\left(C_{\text{RPRAS}}T_{\text{RPRAS}}\log^2\left(\frac{\kappa}{\epsilon_{QLSA}}\right)+T_\xi\right)\frac{n}{\epsilon_{QTA}}\right)$$

queries to QRAM. According to Corollary 2 of [6], T_ξ is $\mathcal{O}\left(\text{polylog}n/\epsilon\right)$, which is dominated by T_{RPRAS}. Thus we ignore T_ξ in the complexity of QLSA + QTA. Following from Lemma 3.18, and Lemma 3.13, we have

$$T_{\text{iter}} = \tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\kappa_Y\frac{\|A\|_F^4}{\hat{h}\left\|\hat{Z}\right\|_2\|Y\|_2}\kappa_{\hat{Z}}\left(\alpha_H\|A_1\|_F^2\|F_1^{-1/2}\|_{\max}^2\right)/\|\hat{Z}\|_2\frac{n}{\epsilon_{QTA}}\right).$$

Then, using Lemma 3.2, we have

$$T_{\text{iter}} = \tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\kappa_Y\frac{\|A\|_F^6}{\hat{h}\left\|\hat{Z}\right\|_2\|Y\|_2}\kappa_{\hat{Z}}\alpha_H/\|\hat{Z}\|_2\frac{n}{\epsilon_{QTA}}\right).$$

Now we can use Lemma 3.12 to derive

$$T_{\text{iter}} = \tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\kappa_Y\frac{\|A\|_F^6}{\hat{h}^2\left\|\hat{Z}\right\|_2\|Y\|_2}\kappa_{\hat{Z}}/\|\hat{Z}\|_2\frac{n}{\epsilon_{QTA}}\right).$$

Then, by the definition of the condition number, we have

$$T_{\text{iter}} = \tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\frac{\|A\|_F^6\kappa_{\hat{Z}}}{\hat{h}^2\sigma_1(\hat{Z})^2\sigma_0(Y)}\frac{n}{\epsilon_{QTA}}\right).$$

Following Lemma A.1, we have

$$\sigma_1(\hat{Z}) \geq \sigma_1(B^T\hat{H}^{-1}B).$$

By the definition of the largest singular value, we have

$$\sigma_1(B^T\hat{H}^{-1}B) = \max_{\|v\|_2=1} v^TB^T\hat{H}^{-1}Bv \geq \frac{1}{\hat{h}}\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}B^TB\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}^T.$$

Combining this with Assumption 1 and Lemma 3.2, we have

$$\sigma_1(\hat{Z}) \geq \frac{1}{2\hat{h}}.$$

Then, using the required precision set by equation (7), we have the total number of queries to QRAM in the k^{th} iteration of Algorithm 2 equals to

$$\begin{aligned}
T_{\text{iter}} =&\tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\|A\|_F^6\frac{\kappa_{\hat{Z}}}{\sigma_0(Y)}\frac{n}{\epsilon_{QTA}}\right)\\
=&\tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\|A\|_F^6\frac{n^{1.5}\kappa_{\hat{Z}}}{\sigma_0(Y)}\frac{\kappa_{M_{\text{MNES}}M_B^{-1}}\|v_{\text{MNES}}\|_2}{\sqrt{\mu}}\right)\\
=&\tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\|A\|_F^6\frac{n^{1.5}}{\mu}\frac{1}{\mu}\frac{\omega^2}{\gamma_1}(1+\frac{1}{\hat{h}}\sigma_1(A)^2)\frac{\kappa_{M_{\text{MNES}}M_B^{-1}}\|v_{\text{MNES}}\|_2}{\sqrt{\mu}}\right)\\
=&\tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\|A\|_F^6\frac{n^{1.5}\omega^2}{\mu^2}\frac{1}{\hat{h}}\sigma_1(A)^2\frac{\kappa_{M_{\text{MNES}}M_B^{-1}}\|v_{\text{MNES}}\|_2}{\sqrt{\mu}}\right),
\end{aligned}$$

where the second equality follows from Lemma 3.7. By the definition of $\hat{h}$, we have

$$\frac{1}{\hat{h}}\sigma_1(A)^2 = \frac{\kappa_A^2\omega^2}{\gamma_1}.$$

By the definition of the condition number and the sub-multiplicativity of spectral norm, we have

$$\begin{aligned}
\kappa_{M_{\text{MNES}}M_B^{-1}} &= \kappa_{M_B^TM_{\text{NES}}}\\
&\leq \kappa_{M_B}\kappa_{\text{NES}}\\
&\leq \kappa_A^3\kappa_D^{1.5},
\end{aligned}$$

where the last inequality holds due to Assumption 3 and the fact that D is a positive diagonal matrix. It follows from Lemma 3.2 that

$$\kappa_D \leq \frac{\omega^4}{\mu^2}.$$

From the proof of part (ii) of Theorem 4.2 in [19], we have

$$\frac{\|v_{\text{MNES}}\|_2}{\sqrt{\mu}} = \mathcal{O}\left(\frac{\omega^3\left(\left\|\hat{A}\right\|_2 + \left\|\hat{b}\right\|_2\right)}{\mu}\right).$$

Putting all this together, in each IPM iteration we need

$$\tilde{\mathcal{O}}_{n,\omega,\frac{1}{\epsilon}}\left(\frac{n^{1.5}\omega^{13}}{\mu^6}\kappa_A^5\|A\|_F^6\left(\left\|\hat{A}\right\|_2 + \left\|\hat{b}\right\|_2\right)\right)$$

queries to QRAM. Recall the relationship between ω, ω^k and $\bar{\omega}$ in equation (8), and the stopping criteria, $\mu \leq \epsilon$. The total number of queries to QRAM is

$$\tilde{\mathcal{O}}_{n,\bar{\omega},\frac{1}{\epsilon}}\left(n^2 \cdot \left(\frac{n^{1.5}\bar{\omega}^{13}}{\epsilon^6}\kappa_A^5\|A\|_F^6\left(\left\|\hat{A}\right\|_2 + \left\|\hat{b}\right\|_2\right)\right)\right).$$

And the total classical arithmetic operations for building (RP-RAS) and updating variables is

$$\tilde{\mathcal{O}}_{n,\bar{\omega},\frac{1}{\epsilon}}\left(n^3 m\right),$$

where $\mathcal{O}(nm)$ is the number of classical arithmetic operations needed for matrix-vector multiplication in the updating process. □

The total number of queries to QRAM of the preconditioned II-QIPM is better than the original II-QIPM proposed in [19] with a improvement factor $\mathcal{O}\left(\frac{\bar{\omega}^6}{\epsilon^3}\right)$ when $\|A\|_F = \mathcal{O}(\|A\|_2)$. The improvement comes from two sources: one is the preconditioning and the other is different rescaling.

Preconditioning mainly improves the complexity of the QLSA, which relies on the condition number of the linear system to be solved and the complexity to build the linear system using block-encoding. Due to preconditioning, both of these two terms are improved by $\mathcal{O}\left(\frac{\bar{\omega}^2}{\epsilon}\right)$.

Rescaling mainly affects the complexity of QTA, which is almost linear with respect to the reciprocal of the accuracy of QTA. The accuracy of QTA is determined by the convergence condition and the residual of (MNES), see Lemma 2.1. In [19], they solve (MNES) to get z directly. They then analyze the residual of (MNES) to determine the required accuracy of QTA, which is linear in the reciprocal of the condition number of (MNES).

In our algorithm, we solve (RP-RAS) to get Δy and then plug Δy into (MNES) using (4), which gives a linear system different from, but similar to (MNES). Similar analysis shows that the required accuracy of our QTA is linear in the reciprocal of the condition number of the different linear system, whose condition number is better than (MNES) by $\mathcal{O}\left(\frac{\bar{\omega}^2}{\epsilon}\right)$. The total improvement is the product of the improvements from these different sources.

4 Conclusion

To mitigate the negative implication that the complexity of QLSA has linear dependence on condition number, and that the condition number of the Newton systems in II-QIPMs tend to infinity as optimality is approached, a preconditioned II-QIPM is proposed in this work. With the preconditioning method introduced by Chai and Toh in [5], we prove that the condition numbers in

II-QIPMs are improved from $\mathcal{O}(1/\mu^2)$ to $\mathcal{O}(1/\mu)$. In each II-QIPM iteration, the computational complexity dependence on n comparable to the $\mathcal{O}(n^2)$ in [19] and comparable to the computational complexity of matrix-vector multiplication in conjugate gradient methods for solving linear systems when $\|A\|_F = \mathcal{O}(\|A\|_2)$. The complexity dependence on ϵ is $\mathcal{O}(1/\epsilon^6)$, which is worse than most classical II-IPMs but is quadratically better than the II-QIPM in [19]. It is possible to further improve the dependence on ϵ by using the iterative refinement methods as discussed in [19] but it is out of the scope of this paper.

Declarations

- Funding: This work was supported by Defense Advanced Research Projects Agency as part of the project W911NF2010022: *The Quantum Computing Revolution and Optimization: Challenges and Opportunities.* This work was also supported by National Science Foundation CAREER DMS-2143915. This research also used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

- Competing interests: The corresponding author is an editor for the journal Computation Optimization and Applications.
- Ethics approval: Not applicable.
- Consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Availability of data and materials: Not applicable.
- Code availability: Not applicable.
- Authors' contributions: Not applicable.

Appendix A

The following lemma is a special case of Theorem 4 from [26] by considering positive semi-definite matrices instead of Hermitian matrices.

Lemma A.1 (Special case of Theorem 4 in [26])**.** *Consider two symmetric positive semi-definite matrices* $(M_1, M_2) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$*, and let* $\operatorname{rank}(M_2) = k$*. Then their singular values satisfy*

1. $\sigma_i(M_1 + M_2) \geq \sigma_i(M_1)$*, for* $i = 1, 2, \ldots, n$*, and*
2. $\sigma_{i+k}(M_1 + M_2) \leq \sigma_i(M_1)$*, for* $i = 1, 2, \ldots, n - k$*.*

For block matrices, we have the following lemma as a special case of Theorem 5 in [26] by considering $i = 1$ and $i = n - r$.

Lemma A.2. *Let M be a symmetric positive definite matrix with 2×2 block structure,*

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

where M_{11} and M_{22} are nonsingular. Then, the singular values of M and the singular values of its Schur complement (M/M_{22}) satisfy

$$\sigma_0(M) \leq \sigma_0(M/M_{22}) \leq \sigma_1(M/M_{22}) \leq \sigma_1(M).$$

From Example 5.6.6 in [14], it is known that the largest singular value of any matrix coincides with its spectral norm. So the following lemma is equivalent to the submultiplicativity of spectral norm of matrices [14].

Lemma A.3. *Let $M_1 \in \mathbb{R}^{m \times n}$ and $M_2 \in \mathbb{R}^{n \times l}$, then the largest singular value of $M_1 M_2$ is bounded from above by the largest singular values of M_1 and M_2, i.e.,*

$$\sigma_1(M_1 M_2) \leq \sigma_1(M_1)\sigma_1(M_2).$$

Below we prove two lemmas involving the smallest nonzero singular value.

Lemma A.4. *Let $M_1 \in \mathbb{R}^{m \times n}$ be a nonzero matrix and $M_2 \in \mathbb{S}^{n \times n}_{++}$ be a diagonal matrix. The smallest nonzero singular value of $M_1 M_2$ is bounded by the smallest nonzero singular value of M_1 and the smallest singular value, i.e., diagonal element, of M_2:*

$$\sigma_0(M_1 M_2) \geq \sigma_0(M_1)\sigma_0(M_2).$$

Proof. Without loss of generality, we can assume $M_1 M_2$ has k nonzero singular values:

$$\sigma_0(M_1 M_2) = \sigma_k(M_1 M_2).$$

Let $\mathbb{S} \subseteq \mathbb{R}^n$ be a subspace. According to Courant-Fischer Min-Max theorem [14], it follows that

$$\sigma_k(M_1 M_2) = \sqrt{\min_{\mathbb{S}:\dim(\mathbb{S})=m-k+1} \max_{x \in \mathbb{S}\setminus\{0\}} \frac{x^\top M_1 M_2 (M_1 M_2)^\top x}{x^\top x}}.$$

It can be rewritten as

$$\sigma_k(M_1 M_2) = \sqrt{\min_{\mathbb{S}:\dim(\mathbb{S})=m-k+1} \max_{x \in \mathbb{S}\setminus\{0\}} \frac{(M_1^\top x)^\top M_2 M_2^\top M_1^\top x}{(M_1^\top x)^\top M_1^\top x} \frac{(M_1^\top x)^\top M_1^\top x}{x^\top x}}.$$

Notice that

$$\frac{(M_1^\top x)^\top M_2 M_2^\top M_1^\top x}{(M_1^\top x)^\top M_1^\top x} \geq \sigma_0(M_2 M_2^\top)$$

as long as $M_1^\top x \neq 0$, which is true for $\sigma_k(M_1 M_2)$ since $M_1 M_2$ has k nonzero singular values. So

$$\begin{aligned} \sigma_k(M_1 M_2) &\geq \sqrt{\min_{\mathbb{S}:\dim(\mathbb{S})=m-k+1} \max_{x \in \mathbb{S}\setminus\{0\}} \frac{(M_1^\top x)^\top M_1^\top x}{x^\top x} \sigma_0(M_2 M_2^\top)} \\ &= \sigma_k(M_1)\sigma_0(M_2). \end{aligned}$$

On one hand, if M_1 has more than k nonzero singular values, then we can prove that $M_1 M_2$ has more than k nonzero singular values, which violates the assumption. On the other hand, if $\sigma_k(M_1) = 0$, then we have found an $x \in \mathbb{S} \setminus \{0\}$ with $\dim(\mathbb{S}) = m - k$ such that $M_1^\top x = 0$, which means $\sigma_0(M_1 M_2) = \sigma_k(M_1 M_2) = 0$ and violates the definition of $\sigma_0(M_1 M_2)$. So $\sigma_0(M_1) = \sigma_k(M_1)$ and $\sigma_0(M_1 M_2) \geq \sigma_0(M_1)\sigma_0(M_2)$. □

Lemma A.5. *Let $M_1 \in \mathbb{S}^n_{++}$ and $M_2 \in \mathbb{S}_{(n+l)\times(n+l)}$ with $l \in \mathbb{Z}_+$ and*

$$M_2 = \begin{bmatrix} M_1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Then the smallest nonzero singular value of M_2 equals to the smallest nonzero singular value of M_1, i.e.,

$$\sigma_0(M_1) = \sigma_0(M_2),$$

and the largest singular value of M_2 equals to the largest singular value of M_1, i.e.,

$$\sigma_1(M_1) = \sigma_1(M_2).$$

Proof. The results are straightforward by checking the SVD of M_1 and M_2. □

References

[1] G. Al-Jeiroudi and J. Gondzio. Convergence analysis of the inexact infeasible interior-point method for linear optimization. *Journal of Optimization Theory and Applications*, 141(2):231–247, 2009.

[2] B. Augustino, G. Nannicini, T. Terlaky, and L. F. Zuluaga. Quantum interior point methods for semidefinite optimization. *Quantum*, 7:1110, 2023.

[3] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, MA, 1997.

[4] J. R. Bunch and B. N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 8(4):639–655, 1971.

[5] J.-S. Chai and K.-C. Toh. Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming. *Computational Optimization and Applications*, 36(2):221–247, 2007.

[6] S. Chakraborty, A. Gilyén, and S. Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. *arXiv preprint arXiv:1804.01973*, 2018.

[7] A. M. Childs, R. Kothari, and R. D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017.

[8] D. E. Crabtree and E. V. Haynsworth. An identity for the Schur complement of a matrix. *Proceedings of the American Mathematical Society*, 22(2):364–366, 1969.

[9] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, et al. Quantum algorithms: A survey of applications and end-to-end complexities. *arXiv preprint arXiv:2310.03011*, 2023.

[10] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[11] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics [full version]. *arXiv preprint arXiv:1806.01838*, 4, 2018.

[12] O. Güler, D. den Hertog, C. Roos, T. Terlaky, and T. Tsuchiya. Degeneracy in interior point methods for linear programming: a survey. *Annals of Operations Research*, 46:107–138, 1993.

[13] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.

[14] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, UK, 2012.

[15] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[16] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* SIAM, Philadelphia, PA, 1995.

[17] I. Kerenidis and A. Prakash. A quantum interior point method for LPs and SDPs. *ACM Transactions on Quantum Computing*, 1(1):1–32, 2020.

[18] M. Mohammadisiahroudi, F. Fakhimi, Z. Wu, and T. Terlaky. An inexact feasible interior point method for linear optimization with high adaptability to quantum computers. *Technical Report: 21T-006, Lehigh University*, 2021.

[19] M. Mohammadisiahroudi, R. Fakhimi, and T. Terlaky. Efficient use of quantum linear system algorithms in interior point methods for linear optimization. *arXiv preprint arXiv:2205.01220*, 2022.

[20] R. D. Monteiro and J. W. O'Neal. Convergence analysis of a long-step primal-dual infeasible interior-point LP algorithm based on iterative linear solvers. *Georgia Institute of Technology*, 2003.

[21] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM, Philadelphia, PA, 1994.

[22] I. Pólik and T. Terlaky. Interior point methods for nonlinear optimization. In G. Di Pillo and F. Schoen, editors, *Nonlinear Optimization*, volume 1989, pages 215–276, Berlin, Heidelberg, 2010. Springer.

[23] F. A. Potra. An infeasible-interior-point predictor-corrector algorithm for linear programming. *SIAM Journal on Optimization*, 6(1):19–32, 1996.

[24] C. Roos. A full-Newton step $O(n)$ infeasible interior-point algorithm for linear optimization. *SIAM Journal on Optimization*, 16(4):1110–1136, 2006.

[25] C. Roos, T. Terlaky, and J.-P. Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach.* John Wiley & Sons, Chichester, 1997.

[26] R. L. Smith. Some interlacing properties of the Schur complement of a Hermitian matrix. *Linear Algebra and Its Applications*, 177:137–144, 1992.

[27] J. van Apeldoorn, A. Cornelissen, A. Gilyén, and G. Nannicini. Quantum tomography using state-preparation unitaries. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1265–1318. SIAM, 2023.

[28] Z. Wu, M. Mohammadisiahroudi, B. Augustino, X. Yang, and T. Terlaky. An inexact feasible quantum interior point method for linearly constrained quadratic optimization. *Entropy*, 25(2), 2023.