

## Article

---

# An Inexact Feasible Quantum Interior Point Method for Linearly Constrained Quadratic Optimization

---

Zeguan Wu, Mohammadhossein Mohammadisiahroudi, Brandon Augustino, Xiu Yang and Tamás Terlaky

## Special Issue

Quantum Machine Learning 2022

Edited by

Prof. Dr. Andreas Wichert

Article

# An Inexact Feasible Quantum Interior Point Method for Linearly Constrained Quadratic Optimization

Zeguan Wu , Mohammadhossein Mohammadisahroudi , Brandon Augustino , Xiu Yang  and Tamás Terlaky  \*

Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, USA

\* Correspondence: terlaky@lehigh.edu

**Abstract:** Quantum linear system algorithms (QLSAs) have the potential to speed up algorithms that rely on solving linear systems. Interior point methods (IPMs) yield a fundamental family of polynomial-time algorithms for solving optimization problems. IPMs solve a Newton linear system at each iteration to compute the search direction; thus, QLSAs can potentially speed up IPMs. Due to the noise in contemporary quantum computers, quantum-assisted IPMs (QIPMs) only admit an inexact solution to the Newton linear system. Typically, an inexact search direction leads to an infeasible solution, so, to overcome this, we propose an inexact-feasible QIPM (IF-QIPM) for solving linearly constrained quadratic optimization problems. We also apply the algorithm to  $\ell_1$ -norm soft margin support vector machine (SVM) problems, and demonstrate that our algorithm enjoys a speedup in the dimension over existing approaches. This complexity bound is better than any existing classical or quantum algorithm that produces a classical solution.

**Keywords:** quantum computing; interior point method; quadratic optimization

**MSC:** 90C20; 90C51; 81P68



**Citation:** Wu, Z.; Mohammadisahroudi, M.; Augustino, B.; Yang, X.; Terlaky, T. An Inexact Feasible Quantum Interior Point Method for Linearly Constrained Quadratic Optimization. *Entropy* **2023**, *25*, 330. <https://doi.org/10.3390/e25020330>

Academic Editor: Andreas Wichert

Received: 31 December 2022

Revised: 7 February 2023

Accepted: 8 February 2023

Published: 10 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Linearly constrained quadratic optimization (LCQO) is defined as optimizing a convex quadratic objective function over a set of linear constraints. Linear optimization is a special case of LCQO that corresponds to the case where the objective function is linear. LCQO has rich theory, algorithms, and applications. Many problems in machine learning can be formulated as LCQO problems, including variants of least square problems and variants of support vector machine training [1,2]. Some important optimization algorithms also have LCQO subproblems, e.g., sequential quadratic programming [1].

The modern age of IPMs was launched by Karmarkar's projective method for linear optimization (LO). Since then, many variants of IPMs have also been applied to nonlinear optimization problems, including LCQO problems [3,4]. Contemporary IPMs progress towards the set of optimal solutions by moving within a neighbourhood of an analytic curve known as the central path. IPMs can be categorized according to whether or not the sequence of iterates produced by the algorithm satisfies feasibility. Feasible IPMs are initialized with a strictly feasible solution and maintain feasibility in each iteration, whereas infeasible IPMs start from an infeasible interior solution and do not require feasibility to be exactly satisfied at any point of the algorithm. For LCQO problems with  $n$  variables, feasible IPMs can produce an  $\epsilon$ -approximate solution using  $\mathcal{O}(\sqrt{n} \log(1/\epsilon))$  iterations, whereas infeasible IPMs require  $\mathcal{O}(n^2 \log(1/\epsilon))$  IPM iterations to converge to an  $\epsilon$ -approximate solution [5,6].

At each IPM iteration, a linear system needs to be solved to obtain the search direction, called the Newton direction. This so-called Newton linear system is traditionally in the form of the augmented system or the normal equation system. Classically, these linear

systems can be solved exactly using Bunch–Parlett factorization if the matrices in the systems are symmetric indefinite [7], or Cholesky factorization if the matrices are symmetric positive definite. Solving the Newton linear systems using direct factorization approaches requires the use of  $\mathcal{O}(n^3)$  arithmetic operations, which suggests that feasible IPMs based on factoring methods cannot exhibit complexity better than  $\mathcal{O}(n^{3.5} \log(1/\epsilon))$ , whereas, with the partial update, they achieve  $\mathcal{O}(n^3 \log(1/\epsilon))$  arithmetic operation complexity. The linear systems can also be solved inexactly using some inexact methods, e.g., Krylov subspace methods, which may require fewer iterations if the desired accuracy of the solutions to the linear systems is not high. However, inaccurately solving the Newton linear systems (i.e., the inaccuracy of the search directions) may result in the infeasibility of the sequence of solutions generated by IPMs; therefore, they have only been used in infeasible IPMs.

The advent of quantum technology has led to the development of many quantum-assisted algorithms for optimization and machine learning applications, such as linear regression [8] and the support vector machine training problem [9]. Following the seminal work on quantum algorithms for solving linear systems of equations [10], researchers have been studying whether QLSAs could yield quantum speedups in classical optimization algorithms. In particular, quantum IPMs (QIPMs) that utilize QLSAs to solve the Newton linear system arising in each iteration have been proposed for LO problems [11,12] and semidefinite optimization problems [13]. To maintain the feasibility of the iterates using quantum subroutines, the authors of [13,14] introduce the so-called orthogonal subspace system (OSS) for SDO and LO problems, and, in particular, demonstrate that a feasible solution to the original Newton system can be recovered from an inexact solution to the OSS. However, linearly constrained quadratic optimization problems, which are fundamental to both optimization and machine learning, have yet to be formally studied in the quantum literature.

In this work, we generalize the OSS for LO problems in [14] to LCQO problems and provide an efficient method for constructing the OSS using a quantum computer. Using the OSS, we can obtain an inexact feasible IPM, solving for the search directions inexactly but maintaining the feasibility of the iterates throughout the process of our IPM. The feasibility of the iterates gives better IPM iteration complexity and the bottleneck becomes solving the linear system, OSS. In particular, we show that a quantum implementation of our algorithm with access to quantum RAM (QRAM) obtains an  $\epsilon$ -approximate solution to a given LCQO problem with worst-case complexity

$$\tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}} \left( \sqrt{n} \left( n \left( \frac{\bar{\omega}^2}{\epsilon} + \sigma_{\max}(Q) \right) \kappa_{VAQ} + n^2 \right) \right),$$

where  $\bar{\omega} = \max_k \omega^k$ ,  $\sigma_{\max}(Q)$  is the maximum singular value of the Hessian of the objective function and  $\kappa_{VAQ}$  is the condition number of a matrix determined by initial data; see Lemma 3. We also consider the application of  $\ell_1$ -norm soft margin SVM problems, in which case, an  $\epsilon$ -approximate solution is obtained with complexity

$$\tilde{\mathcal{O}}_{m, n, \bar{\omega}, \frac{1}{\epsilon}} \left( (m+n)^{1.5} \left( \frac{\bar{\omega}^2}{\epsilon} + \sigma_{\max}(Q) \right) \kappa_{VAQ} + (m+n)^{2.5} \right).$$

Here,  $m$  is the number of features and  $n$  is the number of data points.  $\bar{\omega}$ ,  $Q$ , and  $\kappa_{VAQ}$  are defined similarly from the LCQO formulation of the SVM problem; see Section 4. The dependence on dimension is better than any existing quantum or classical algorithm.

The rest of this paper is organized as follows: in Section 2, we introduce IPMs for LCQO and the OSS system; in Section 3, we discuss how to use quantum algorithms to find the Newton directions and analyze the complexity of our IF-QIPM; in Section 4, we apply our IF-QIPM to the support vector machine problem. Discussions are provided in Section 5, and some technical proofs are moved to the Appendixes A and B.

## 2. Preliminaries

In this section, we introduce notations before reviewing the theory of IPMs applied to LCQO, and derive the OSS system for the class of problems.

### 2.1. Notation

Vectors are typically represented by lower-case letters. We write  $0_n$  when referring to the  $n$ -dimensional all-zeros vector, and the  $n$ -dimensional all-ones vector is denoted by  $e_n$ . When the dimension is obvious from the context, we may write  $0$  or  $e$ , respectively. Matrices are typically represented with upper-case letters. The identity of dimension  $n$  is denoted by  $I_{n \times n}$ , and  $0_{n \times m}$  represents the  $n \times m$ -dimensional all-zero matrix, again, dropping these subscripts when the dimension is obvious from the context. For a general  $n \times m$ -dimensional matrix  $H$ , we write  $H_{i \cdot}$  to refer to its  $i$ th row, and, similarly, denote the  $j$ th column by  $H_{\cdot j}$ . For the  $(i, j)$ th element of  $H$ , we write  $H_{ij}$  or  $H_{i,j}$ .

For real-valued functions  $f_1, f_2$ , and  $f_3$ , we write

$$f_1 = \mathcal{O}(f_2)$$

if there exists a positive number  $k_4$  such that  $f_1 \leq k_4 f_2$ . We write

$$f_1 = \tilde{\mathcal{O}}_{f_3}(f_2)$$

if there exists a positive number  $k_5$  such that  $f_1 \leq k_5 f_2 \times \text{poly log}(f_3)$ .

### 2.2. IPMs for LCQO

In this work, LCQO is defined as follows.

**Definition 1** (LCQO Problem). *For vectors  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ , and matrices  $A \in \mathbb{R}^{m \times n}$  and  $Q \in \mathbb{R}^{n \times n}$  with  $\text{rank}(A) = m \leq n$  and  $Q$  symmetric positive semidefinite, we define the primal and dual LCQO problems as:*

$$(P) \quad \begin{aligned} & \min c^T x + \frac{1}{2} x^T Q x, \\ & \text{s.t. } Ax = b, \\ & \quad x \geq 0, \end{aligned} \quad (D) \quad \begin{aligned} & \max b^T y - \frac{1}{2} x^T Q x, \\ & \text{s.t. } A^T y + s - Qx = c, \\ & \quad s \geq 0, \end{aligned} \quad (1)$$

where  $x \in \mathbb{R}^n$  is the vector of primal variables, and  $y \in \mathbb{R}^m$ ,  $s \in \mathbb{R}^n$  are vectors of the dual variables. Problem  $(P)$  is called the primal problem and  $(D)$  is called the dual problem.

Since  $A$  is of full row-rank,  $A$  does not contain any null rows, and we further make the following assumption on matrix  $A$ .

**Assumption 1.** *Matrix  $A$  has no all-zero columns.*

**Remark 1.** *Suppose that  $A$  has zero columns. Without a loss of generality, assume that the  $n$ th column is all-zero. Introducing a new variable  $x_{n+1}$ , we can rewrite the problem as*

$$\begin{aligned} & \min \begin{bmatrix} c \\ 0 \end{bmatrix}^T \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix}^T \begin{bmatrix} Q & 0_{n \times 1} \\ 0_{1 \times n} & 0 \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix}, \\ & \text{s.t. } \begin{bmatrix} A_{\cdot 1} & \cdots & A_{\cdot (n-1)} & 0_{m \times 1} & 0_{m \times 1} \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \\ & \quad x \geq 0, x_{n+1} \geq 0. \end{aligned}$$

The new LCQO problem is equivalent to the original one, and contains fewer all-zero columns. Iterating this procedure to eliminate each of the all-zero columns, we obtain a new LCQO problem satisfying Assumption 1 with no more than  $2n - m$  variables and  $n$  constraints in the worst case.

**Assumption 2.** There exists a solution  $(x, y, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$  such that

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

The set of *primal–dual feasible solutions* is defined as

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

and, similarly, the set of *interior feasible primal–dual solutions* is given by

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

By strong duality, the set of optimal solutions can be characterized as

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

where  $xs$  denotes the Hadamard, i.e., component-wise product of  $x$  and  $s$ . Let  $\epsilon > 0$ ; then, the set of  $\epsilon$ -approximate solutions to Problem (1) can be defined as

$$\mathcal{PD}_\epsilon := \left\{ (x, y, s) \in \mathcal{PD} : x^T s \leq n\epsilon \right\}. \quad (2)$$

Let  $X$  and  $S$  be diagonal matrices of  $x$  and  $s$ , respectively. Under Assumption 2, for all  $\mu > 0$ , the perturbed system of optimality conditions

$$\begin{aligned} Ax &= b, \\ A^T y + s - Qx &= c, \\ XSe &= \mu e, \\ (x, s) &\geq 0 \end{aligned} \quad (3)$$

has a unique solution  $(x(\mu), y(\mu), s(\mu))$ , and this set of solutions gives rise to the primal–dual central path

$$\mathcal{CP} := \left\{ (x, y, s) \in \mathcal{PD}^0 \mid x_i s_i = \mu \text{ for } i \in \{1, \dots, n\}; \text{ for } \mu > 0 \right\}.$$

IPMs apply Newton's method to solve system (3). At each iteration of infeasible IPMs, a candidate solution to the primal–dual LCQO pair in (1) is updated by solving the following linear system to find the Newton direction:

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}, \quad (4)$$

where

$$\begin{aligned} r_p &= b - Ax \\ r_d &= c - A^T y - s \\ r_c &= \sigma \mu e - XSe, \end{aligned}$$

are residuals, and  $\sigma \in (0, 1)$  is the barrier reduction parameter. If  $r_p = 0$  and  $r_d = 0$ , then the solution  $(x, y, s)$  exactly satisfies primal–dual feasibility. We can also define residuals in different ways as we will show later. Once the Newton direction is found, one can move along the direction but has to stay in a neighbourhood of the central path, which is defined as

$$\mathcal{N}_2(\theta) := \left\{ (x, y, s) \in \mathcal{PD}^0 \mid \|XSe - \mu e\|_2 \leq \theta \mu \right\}, \quad (5)$$

where  $\theta \in (0, 1)$ .

Until relatively recently, inexact solution approaches to solve the Newton linear system (4) had only been utilized in inexact infeasible IPMs (II-IPMs). For LCQO problems, ref. [6] proposes an II-IPM using an iterative method to solve the Newton systems and obtains a worst-case iteration complexity  $\mathcal{O}(n^2 \log(\frac{1}{\epsilon}))$ . On the other hand, feasible IPMs for LCQO problems enjoy  $\mathcal{O}(\sqrt{n} \log(\frac{1}{\epsilon}))$  iteration complexity [15–17]. In [5], the author provides a general inexact feasible IPM for LCQO problems but does not discuss how the sequence of iterates could be guaranteed to maintain primal–dual feasibility exactly when using inexact linear system solvers. This is a vital consideration, as the feasible neighborhood of the central path as outlined in (5) is a subset of the primal–dual feasible set; if primal and dual feasibility are not satisfied exactly at any point in the algorithm, the iterates leave this neighborhood and the method fails. Our work fills this gap by using a method inspired by the QIPMs of [13,14].

### 2.3. Orthogonal Subspaces System

Assume that  $(x, y, s) \in \mathcal{PD}^0$ . To maintain the feasibility of the primal and dual variables, the first two linear equations in system (4) need to be solved with  $r_p = 0$  and  $r_d = 0$  exactly, which can be guaranteed if  $\Delta x$  lies in the null space of  $A$ , denoted as  $\text{Null}(A)$ , and  $\Delta s = Q\Delta x - A^T\Delta y$ . Accordingly, we can rewrite system (4) by representing  $\Delta x$  as a linear combination of basis elements of  $\text{Null}(A)$ . To achieve this, we partition  $A$  as  $A = [A_B \ A_N]$ , where  $A_B$  is a basis of  $A$ . Then, we construct the following matrix:

$$V = \begin{bmatrix} A_B^{-1} A_N \\ -I \end{bmatrix}.$$

Matrix  $V$  has a full column rank and satisfies  $AV = 0$ , i.e., the columns of  $V$  span the null space of  $A$ . Let  $\Delta x = V\lambda$ , where  $\lambda \in \mathbb{R}^{n-m}$  is the unknown coefficient vector used to determine  $\Delta x$ . Subsequently, we can rewrite system (4) by substituting  $\Delta x$  and  $\Delta s$  in the third equation as

$$SV\lambda + X(QV\lambda - A^T\Delta y) = r_c \Leftrightarrow [SV + XQV \quad -XA^T] \cdot \begin{bmatrix} \lambda \\ \Delta y \end{bmatrix} = r_c. \quad (6)$$

A similar system was proposed and called “Orthogonal Subspaces System” (OSS) in [13,14], and we use the same name in this work. The matrix in the OSS system (6) is of size  $n \times n$ , and it is nonsingular. Even if the OSS system is solved inexactly, primal and dual feasibility are preserved by computing  $\Delta x = V\lambda$  and  $\Delta s = QV\lambda - A^T\Delta y$ . Thus, we can conclude that any inexactness will only impact the third equation of (4), i.e.,  $r_p = 0$  and  $r_d = 0$ . This property of the OSS system is very convenient when analyzing the proposed inexact IPM, and allows us to obtain the best known iteration complexity for IPMs.

## 3. Inexact Feasible IPM with QLSAs

In this section, we propose our IF-QIPM for LCQO problems. We begin with the IF-IPM structure introduced by [5] and describe how to quantize it into an IF-QIPM. Then, we analyze the construction of the OSS system and conclude by analyzing the overall complexity of our IF-QIPM.

### 3.1. IF-IPM for LCQO

In [5], the author studies a general conceptual form IF-IPM for QCLO problems by assuming the feasibility of the primal and dual iterates, which induces the following system:

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ r_c \end{bmatrix}, \quad (7)$$

where  $r_c = \sigma\mu e - XSe$ , with  $\sigma \in (0, 1)$  being the reduction factor of the central path parameter  $\mu$ , i.e.,  $\mu^{new} = \sigma\mu$ . When system (7) is solved with  $r_c = \sigma\mu e - XSe$  inexactly

yielding an error  $r$ , if  $\|r\|_2 \leq \delta \|r_c\|_2$  for some  $\delta \in (0, 1)$ , the inexact IPM converges to an  $\epsilon$ -approximate solution to Problem (1) in at most  $\mathcal{O}(\sqrt{n} \log(1/\epsilon))$  iterations. As we mentioned earlier, it is not specified in [5] how to preserve primal and dual feasibility when system (7) is solved inexactly. Thus, it is presently not clear whether one could recover the convergence conditions described in [5] using inexact approaches, which are reliant on the assumption of primal-dual feasibility (see, e.g., system (7)).

Now, we present a general procedure of how to solve system (7) inexactly, while the inexactness error occurs only in the third equation of system (7). Let  $(\lambda, \Delta y)$  be an inexact solution for system (6) and  $r$  be the error at this solution, i.e.,

$$[SV + XQV \quad -XA^T] \cdot \begin{bmatrix} \lambda \\ \Delta y \end{bmatrix} = r_c + r.$$

The corresponding Newton step

$$\begin{aligned} \Delta x &= V\lambda \\ \Delta s &= Q\Delta x - A^T\Delta y \end{aligned}$$

satisfies

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ r_c + r \end{bmatrix}.$$

Recall that once  $(\lambda, \Delta y)$  is determined, then  $(\Delta x, \Delta s)$  is also (uniquely) determined. An interesting property is that, if  $(\lambda, \Delta y)$  and  $(\Delta x, \Delta y, \Delta s)$  can be deduced from each other, then the OSS system and system (7) yield the same error term  $r$ . Hence, the convergence conditions built upon system (7) can be directly examined using the residual  $r_c$  and error  $r$  of the OSS system. Let  $\epsilon_{OSS}$  be the target accuracy of the OSS system (6), i.e.,

$$\|(\lambda - \lambda^*, \Delta y - \Delta y^*)\|_2 \leq \epsilon_{OSS},$$

where  $(\lambda^*, \Delta y^*)$  is the accurate solution. According to [5], in order to guarantee that the IF-IPM converges, we must have

$$\begin{aligned} \|r\|_2 &= \left\| [SV + XQV \quad -XA^T] \cdot \begin{bmatrix} \lambda \\ \Delta y \end{bmatrix} - r_c \right\|_2 \\ &\leq \| [SV + XQV \quad -XA^T] \|_2 \epsilon_{OSS} \\ &\leq \delta \|r_c\|_2, \end{aligned}$$

where  $\delta \in (0, 1)$  is a constant parameter. Therefore, to ensure the convergence of the IF-IPM, it suffices to set

$$\epsilon_{OSS} \leq \delta \frac{\|r_c\|_2}{\| [SV + XQV \quad -XA^T] \|_2}.$$

The IF-IPM is presented in full detail in Algorithm 1. In each iteration, we build and solve system (6) classically. We solve system (6) to the accuracy just introduced above and then compute the feasible Newton step from the inexact solution and take a full Newton step.

**Algorithm 1** Short-step IF-IPM

---

```

1: Choose  $\epsilon > 0$ ,  $\delta \in (0, 1)$ ,  $\theta \in (0, 1)$ ,  $\beta \in (0, 1)$  and  $\sigma = (1 - \frac{\beta}{\sqrt{n}})$ .
2:  $k \leftarrow 0$ 
3: Choose initial feasible interior solution  $(x^0, y^0, s^0) \in \mathcal{N}(\theta)$ 
4: while  $(x^k, y^k, s^k) \notin \mathcal{PD}_\epsilon$  do
5:    $\mu^k \leftarrow \frac{(x^k)^T s^k}{n}$ 
6:    $\epsilon_{OSS}^k \leftarrow \delta \|r_c^k\|_2 / \left\| \begin{bmatrix} S^k V + X^k Q V^k & -X^k A^T \end{bmatrix} \right\|_2$ 
7:    $(\lambda^k, \Delta y^k) \leftarrow \text{solve system (6) with accuracy } \epsilon_{OSS}^k$ 
8:    $\Delta x^k = V \lambda^k$  and  $\Delta s^k = -A^T \Delta y^k$ 
9:    $(x^{k+1}, y^{k+1}, s^{k+1}) \leftarrow (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$ 
10:   $k \leftarrow k + 1$ 
11: end while
12: return  $(x^k, y^k, s^k)$ 

```

---

In the quantum-assisted IF-IPM, or IF-QIPM, we propose accelerating Step 7 using quantum subroutines. In the next sections, we investigate how to use quantum algorithms to build and solve the OSS system and obtain the Newton direction.

## 3.2. IF-QIPM for LCQO

The pseudocode of our IF-QIPM is presented in Algorithm 2. At each iteration of the IF-QIPM, we construct and solve system (6) and compute the Newton direction using quantum algorithms. To obtain an  $\epsilon_{OSS}$ -approximate solution of system (6), we first block encode system (8); see Appendix A. Then, we use quantum algorithms to solve for an  $\epsilon_{QLSA}$ -approximate solution of system (8). This solution is normalized but we can rescale it to obtain an  $\epsilon_{OSS}$ -approximate solution of system (6). Details are discussed later in this section.

**Algorithm 2** Short-step IF-QIPM

---

```

1: Choose  $\epsilon > 0$ ,  $\delta \in (0, 1)$ ,  $\theta \in (0, \theta_0)$ ,  $\beta \in (0, 1)$  and  $\sigma = (1 - \frac{\beta}{\sqrt{n}})$ .
2:  $k \leftarrow 0$ 
3: Choose initial feasible interior solution  $(x^0, y^0, s^0) \in \mathcal{N}(\theta)$ 
4: while  $(x^k, y^k, s^k) \notin \mathcal{PD}_\epsilon$  do
5:    $\mu^k \leftarrow \frac{(x^k)^T s^k}{n}$ 
6:    $\epsilon_{OSS}^k \leftarrow \delta \|r_c^k\|_2 / \left( \sqrt{2} \left\| \begin{bmatrix} S^k V + X^k Q V^k & -X^k A^T \end{bmatrix} \right\|_2 \right)$ 
7:    $(\lambda^k, \Delta y^k) \leftarrow \text{solve system (6) with accuracy } \epsilon_{OSS}^k \text{ quantumly}$ 
8:    $\Delta x^k = V \lambda^k$  and  $\Delta s^k = -A^T \Delta y^k$ 
9:    $(x^{k+1}, y^{k+1}, s^{k+1}) \leftarrow (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$ 
10:   $k \leftarrow k + 1$ 
11: end while
12: return  $(x^k, y^k, s^k)$ 

```

---

Here,  $\theta_0 < 1$  and its value will be discussed later. First, we introduce some notations to simplify the OSS system. In the  $k$ th iteration of Algorithm 2, let

$$M^k = [S^k V + X^k Q V^k \quad -X^k A^T], z^k = \begin{bmatrix} \lambda^k \\ \Delta y^k \end{bmatrix}.$$

Then, the OSS system can be rewritten as

$$M^k z^k = r_c^k.$$

As discussed in [14], to solve the OSS system (6) using quantum algorithms, we can first rewrite it as the normalized Hermitian OSS system

$$\frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} 0 & M^k \\ (M^k)^T & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ z^k \end{bmatrix} = \frac{1}{\sqrt{2}\|M^k\|_F} \cdot \begin{bmatrix} r_c^k \\ 0 \end{bmatrix}. \quad (8)$$

To use the QLSAs mentioned earlier, we need to turn the linear system (8) into a quantum linear system using the block encoding introduced in [18]. To this end, we first decompose the coefficient matrix in linear system (8) as

$$\frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} 0 & M^k \\ (M^k)^T & 0 \end{bmatrix} = \frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} 0 & 0 \\ (M^k)^T & 0 \end{bmatrix} + \frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} 0 & M^k \\ 0 & 0 \end{bmatrix}, \quad (9)$$

where

$$\begin{bmatrix} 0 & 0 \\ (M^k)^T & 0 \end{bmatrix} = \begin{bmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{(n-m) \times n} & V^T & 0_{(n-m) \times n} \\ 0_{m \times n} & 0_{m \times n} & -A \end{bmatrix} \times \left( \begin{bmatrix} 0_{n \times n} & 0_{n \times n} \\ S^k & 0_{n \times n} \end{bmatrix} + \begin{bmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & Q & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & I_{n \times n} \end{bmatrix} \begin{bmatrix} 0_{n \times n} & 0_{n \times n} \\ X^k & 0_{n \times n} \\ X^k & 0_{n \times n} \end{bmatrix} \right). \quad (10)$$

To compute matrix  $V$ , we need to find a basis matrix  $A_B$  of matrix  $A$  and we need to compute the inverse matrix  $A_B^{-1}$ . Both steps are nontrivial and can be expensive. However, we can reformulate the LCQO problem as follows:

$$\begin{aligned} & \min c^T x + \frac{1}{2} x^T Q x \\ \text{s.t. } & \begin{bmatrix} I & 0 & A \\ 0 & I & -A \end{bmatrix} \begin{bmatrix} x' \\ x'' \\ x \end{bmatrix} = \begin{bmatrix} b \\ -b \end{bmatrix} \\ & x \geq 0, x' \geq 0, x'' \geq 0. \end{aligned}$$

In this case, we have an obvious basis

$$A_B = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

and matrix  $V$  can be constructed efficiently

$$V = \begin{bmatrix} A_B^{-1} A_N \\ -I \end{bmatrix} = \begin{bmatrix} [I \ 0] [A] \\ [0 \ I] [-A] \\ -I \end{bmatrix} = \begin{bmatrix} A \\ -A \\ -I \end{bmatrix}.$$

Since matrix  $A$  has no all-zero rows, matrix  $V$  has no all-zero rows either. This property of the reformulation is useful in the analysis of the proposed IF-QIPM but we do not want to build the complexity analysis on the reformulated problem. Thus, without a loss of generality we may make the following assumption.

**Assumption 3.** *Matrix  $A$  is of the form  $A = [I \ A_N]$ .*

To simplify the analysis, we further assume that the input data are integers.

**Assumption 4.** *The input data of Problem (1) are integers.*

Based on the two assumptions above, we have the following lemma.

**Lemma 1.** Matrix  $V$  equals

$$V = \begin{bmatrix} A_N \\ -I \end{bmatrix}$$

and

$$\min_{i=1,\dots,n} \{\|V_i\|_2^2\} = \min\{1, \min_{i=1,\dots,m} \|(A_N)_{i\cdot}\|_2^2\} = 1,$$

where  $V_i$  and  $(A_N)_{i\cdot}$  are the  $i$ th row of  $V$  and  $A_N$ , respectively.

Now, we are ready to give  $\theta_0$  in our definition of the central path neighborhood; see (5). We set

$$\theta_0 = \min\left\{\frac{1}{3\sqrt{n}}, \frac{1}{4\|QVV^T\|_F + 1}\right\}. \quad (11)$$

We also define  $\omega^k$  as the maximum of the values of primal variables and dual slack variables in the  $k$ th iteration.

**Definition 2.** Let  $(x^k, y^k, s^k)$  be a candidate solution for Problem (1); then,

$$\omega^k = \max_{i \in \{1, \dots, n\}} \{x_i^k, s_i^k\}.$$

As is standard in the literature on quantum algorithms, in this work, we assume access to quantum random access memory (QRAM). Then, Step 7 of Algorithm 2 consists of three parts: (1). use block encoding to build system (8); (2). use QLSAs to solve system (8); (3). use quantum tomography algorithms (QTAs) to extract the classical solution. We use the block-encoding methods introduced in [18] to block-encode linear system (8).

**Proposition 1.** In the  $k$ th iteration of Algorithm 2, using the block-encoding methods introduced in [18] and the decomposition described in Equations (9) and (10), a

$$\left( \frac{\sqrt{\|V\|_F^2 + \|A\|_F^2} \sqrt{2}\omega^k}{\|M^k\|_F} (\sqrt{2}\|Q\|_F + \sqrt{2} + 1), \mathcal{O}(\text{poly log}(n)), \frac{\epsilon_{QLSA}}{\kappa_{M^k}^3} \right)$$

-block-encoding of the matrix in system (8) can be implemented efficiently and the complexity will be dominated by the complexity of the QLSA step. Here,  $\epsilon_{QLSA}$  is the accuracy required for the QLSA step and  $\kappa_{M^k}$  is the condition number of matrix  $M^k$ .

**Proof.** See Appendix A for proof.  $\square$

Provided access to QRAM, the complexity associated with block encoding the OSS system coefficient matrix and preparing a quantum state encoding the right hand side amounts to polylogarithmic overhead. The cost of these steps is therefore negligible when compared with the complexity contributed by QLSAs and QTAs, so we ignore it here. To bound the total complexity contributed by QLSAs and QTAs, we first need to analyze the accuracy of QLSA characterized by  $\epsilon_{QLSA}$ , the accuracy of QTA characterized by  $\epsilon_{QTA}$ , and their relationship.

In each iteration, we use a QLSA to solve the block-encoded version of system (8) and obtain an  $\epsilon_{QLSA}$ -approximate solution. Then, we use a QTA to extract an  $\epsilon_{QTA}$ -approximate solution from the quantum machine. In the context of QLSAs and QTAs, if  $\tilde{z}$  is an  $\epsilon$ -approximate solution of  $z$ , then  $\tilde{z}$  satisfies

$$\left\| \frac{\tilde{z}}{\|\tilde{z}\|_2} - \frac{z}{\|z\|_2} \right\|_2 \leq \epsilon$$

Observe that this definition of accuracy differs from the concept of  $\epsilon$ -approximate solutions defined in (2).

Similar to [12,13], the QLSA we use is proposed by [19] and the QTA we use is proposed by [20]. Following the argument in Section 2 in [12], we can establish the relationship among  $\epsilon_{QLSA}$ ,  $\epsilon_{QTA}$ , and  $\epsilon_{OSS}^k$  as

$$\epsilon_{QLSA} = \epsilon_{QTA} = \frac{1}{2} \cdot \frac{\sqrt{2}\|M^k\|_F}{\|r_c^k\|_2} \epsilon_{OSS}^k, \quad (12)$$

where  $\epsilon_{OSS}^k$  is defined as the  $\ell_2$  norm of the residual when solving system (8) inexactly in the  $k$ th iteration. This coefficient is also used to rescale the solution. According to [12], we rescale the normalized solution obtained from QLSA and QTA by

$$\frac{\|r_c^k\|_2}{\sqrt{2}\|M^k\|_F}$$

to obtain the  $\epsilon_{OSS}^k$ -approximate solution for system (6). Here, we did not add superscript to  $\epsilon_{QLSA}$  and  $\epsilon_{QTA}$ , and the reason shall be revealed later. Let

$$\begin{bmatrix} \tilde{0}^k \\ \tilde{z}^k \end{bmatrix}$$

be an inexact solution for system (8) in the  $k$ th iteration. Then, the norm of residual of system (8), which is  $\epsilon_{OSS}^k$ , and the norm of residual of system (6), which is  $\|M^k\tilde{z}^k - r_c^k\|_2$ , satisfies

$$\begin{aligned} \epsilon_{OSS}^k &= \left\| \frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} 0 & M^k \\ (M^k)^T & 0 \end{bmatrix} \begin{bmatrix} \tilde{0}^k \\ \tilde{z}^k \end{bmatrix} - \frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} r_c^k \\ 0 \end{bmatrix} \right\|_2 \\ &= \left\| \frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} M^k\tilde{z}^k \\ (M^k)^T\tilde{0}^k \end{bmatrix} - \frac{1}{\sqrt{2}\|M^k\|_F} \begin{bmatrix} r_c^k \\ 0 \end{bmatrix} \right\|_2 \\ &\geq \left\| \frac{1}{\sqrt{2}\|M^k\|_F} M^k\tilde{z}^k - \frac{1}{\sqrt{2}\|M^k\|_F} r_c^k \right\|_2 \\ &\geq \frac{1}{\sqrt{2}\|M^k\|_F} \|M^k\tilde{z}^k - r_c^k\|_2. \end{aligned}$$

Recall that the error arising from the OSS system (6) is the same as the error in the full Newton system (7); then, we can directly use the convergence condition in [5], i.e.,

$$\|M^k\tilde{z}^k - r_c^k\|_2 \leq \delta \|r_c^k\|_2.$$

We can require

$$\|M^k\tilde{z}^k - r_c^k\|_2 \leq \sqrt{2}\|M^k\|_F \epsilon_{OSS}^k \leq \delta \|r_c^k\|_2$$

and it follows that

$$\epsilon_{OSS}^k \leq \frac{\delta \|r_c^k\|_2}{\sqrt{2}\|M^k\|_F}.$$

Then, choosing

$$\epsilon_{OSS}^k = \frac{\delta \|r_c^k\|_2}{\sqrt{2}\|M^k\|_F} \text{ and } \epsilon_{QLSA} = \epsilon_{QTA} = \frac{\|M^k\|_F \epsilon_{OSS}^k}{\sqrt{2}\|r_c^k\|_2} = \frac{\delta}{2}$$

ensures the convergence of the IF-QIPM. The complexities for each step are also available now. Using the QLSA from [19] and QTA from [20], we have the complexity for QLSA and QTA:

$$T_{QLSA} = \tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}} \left( \kappa_{M^k} \frac{\omega^k}{\|M^k\|_F} \right),$$

$$T_{QTA} = \tilde{\mathcal{O}}_n \left( \frac{n}{\epsilon_{QTA}} T_{QLSA} \right) = \tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}} \left( n \kappa_{M^k} \frac{\omega^k}{\|M^k\|_F} \right).$$

Since we have  $\epsilon_{QTA} = \frac{\delta}{2}$  and  $\delta \in (0, 1)$  is a constant parameter, we omit  $\epsilon_{QTA}$  in the Big-O notation. Note that the complexity of the block-encoding procedure is dominated by that of QLSA and QTA and thus we ignore the complexity contributed by block encoding. In Step 8, the complexity contributed by computing Newton step from OSS solution is  $\mathcal{O}(n^2)$ . The total complexity for the  $k$ th iteration of IF-QIPM will be

$$\mathcal{O}(T_{QTA} + n^2) = \tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}} \left( \frac{n \omega^k \kappa_{M^k}}{\|M^k\|_F} + n^2 \right). \quad (13)$$

### 3.2.1. Bound for $\omega^k / \|M^k\|_F$

In this section, all of the quantities that we consider are from the  $k$ th iteration. For simplicity, we omit the superscript  $k$  in this section unless we need it. Using the property of trace, we have

$$\begin{aligned} \|M\|_F^2 &= \text{tr}(M^T M) \\ &= \text{tr}((SV + XQV)(SV + XQV)^T + XA^T AX) \\ &= \text{tr}((SV + XQV)(SV + XQV)^T) + \text{tr}(XA^T AX) \\ &= \text{tr}(SVV^T S) + \text{tr}(XQVV^T S) + \text{tr}(SVV^T QX) + \text{tr}(XQVV^T QX) + \text{tr}(XA^T AX). \end{aligned}$$

For the non-symmetric term, due to the cyclic invariant property of trace, we have

$$\text{tr}(XQVV^T S) = \text{tr}(SXQVV^T).$$

Recalling the central path neighborhood that we defined in (5), we define a matrix  $E$  such that

$$E = \frac{1}{\mu\theta}(XS - \mu I). \quad (14)$$

It is obvious that  $E$  is a diagonal matrix and satisfies

$$\|Ee\|_2 < 1,$$

which leads to

$$|\text{tr}(E)| \leq \|Ee\|_1 \leq \sqrt{n} \|E\|_F = \sqrt{n} \|Ee\|_2 < \sqrt{n} \text{ and } I - E \succ 0 \text{ and } I + E \succ 0.$$

With this, we can have

$$\begin{aligned} \text{tr}(XQVV^T S) &= \text{tr}(SXQVV^T) \\ &= \text{tr}((\theta\mu E + \mu I)QVV^T) \\ &= \text{tr}(\theta\mu E QVV^T) + \text{tr}(\mu QVV^T). \end{aligned}$$

For the second term, we know that  $Q$  and  $V^T Q V$  are both positive semidefinite. Thus, we can have

$$\text{tr}(Q V V^T) = \text{tr}(V^T Q V) \geq 0$$

because of the cyclic invariant property of trace. According to the Cauchy–Schwarz inequality, we have

$$\text{tr}(E Q V V^T)^2 \leq \|E\|_F^2 \|Q V V^T\|_F^2.$$

Thus, we have

$$\text{tr}(E Q V V^T) \geq -\|Q V V^T\|_F.$$

Thus, we have

$$\begin{aligned} \text{tr}(X Q V V^T S) &= \text{tr}(\theta \mu E Q V V^T) + \text{tr}(\mu Q V V^T) \\ &\geq \mu \left( \text{tr}(Q V V^T) - \theta \|Q V V^T\|_F \right) \\ &\geq -\theta \mu \|Q V V^T\|_F \\ &\geq -\frac{\mu}{4}, \end{aligned}$$

where the last inequality holds due to condition (11). Thus, we can bound  $\|M\|_F$  by

$$\begin{aligned} \|M\|_F^2 &= \text{tr}(S V V^T S) + \text{tr}(X Q V V^T S) + \text{tr}(S V V^T Q X) \\ &\quad + \text{tr}(X Q V V^T Q X) + \text{tr}(X A^T A X) \\ &\geq \text{tr}(S V V^T S) + \text{tr}(X Q V V^T Q X) + \text{tr}(X A^T A X) - \frac{\mu}{2}. \end{aligned}$$

Since  $X Q V V^T Q X \succeq 0$ , we have

$$\|M\|_F^2 \geq \text{tr}(S V V^T S) + \text{tr}(X A^T A X) - \frac{\mu}{2}.$$

Since  $X$  and  $S$  are both positive diagonal matrices, we have

$$\begin{aligned} \|M\|_F^2 &\geq \text{tr}(S V V^T S) + \text{tr}(X A^T A X) - \frac{\mu}{2} \\ &= \sum_i s_i^2 (V V^T)_{ii} + \sum_i x_i^2 (A^T A)_{ii} - \frac{\mu}{2} \\ &\geq \omega^2 - \frac{\mu}{2}. \end{aligned}$$

As we mentioned in the very beginning of this section, at each iteration,  $\omega$  is indeed  $\omega^k$ , but the superscript is ignored here. Now, we aim to find a bound for  $\mu$  so we can further bound  $\|M\|_F^2$ . Since  $\omega$  is the upper bound for the magnitude of the primal and dual slack variables, we have

$$\omega^2 \geq x_i s_i.$$

Recall the definition of matrix  $E$ ; see (14). Thus, we have

$$\omega^2 \geq x_i s_i = \mu + \theta \mu E_{ii} \geq \mu - \theta \mu = (1 - \theta) \mu.$$

Thus,

$$\|M\|_F^2 \geq \omega^2 - \frac{\mu}{2} \geq \omega^2 - \frac{1}{2} \frac{\omega^2}{1 - \theta} \geq \omega^2 - \frac{1}{2} \frac{\omega^2}{1 - 1/3} = \frac{\omega^2}{4},$$

where the last inequality follows from the bound for  $\theta$ ; see (11). Thus, we have

$$\frac{\omega}{\|M\|_F} \leq 2 = \mathcal{O}(1).$$

### 3.2.2. Bound for $\kappa_{M^k}$

Similar to the previous section, we ignore the superscript  $k$  unless we need it. We will start with a general result and then work on the matrix  $M^k$ . The following lemma is a well-known result regarding condition numbers of matrices and can be proven using Courant–Fischer–Weyl min–max principle [21].

**Lemma 2.** *For any full row rank matrix  $P \in \mathbb{R}^{m \times n}$  and symmetric positive definite matrix  $D \in \mathbb{R}^{n \times n}$ , their condition number satisfies*

$$\kappa(PDP^T) \leq \kappa(D)\kappa(PP^T).$$

Next, we analyze the matrix in the OSS system (8). Specifically, we focus on  $M^T M$  since we are interested in the spectral property of the OSS system (8). Using the matrix  $E$  defined in (14), we have the following decomposition:

$$\begin{aligned} M^T M &= \begin{bmatrix} V^T(S + XQ)^T(S + XQ)V & -V^T(S + XQ)^T X A^T \\ -A X(S + XQ)V & A X^2 A^T \end{bmatrix} \\ &= \begin{bmatrix} V^T(S + XQ)^T(S + XQ)V & -V^T \mu(\theta E) A^T - V^T Q^T X^2 A^T \\ -A \mu(\theta E) V - A X^2 Q V & A X^2 A^T \end{bmatrix} \\ &= \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} (S + XQ)^T(S + XQ) & -\mu\theta E - Q X^2 \\ -\mu\theta E - X^2 Q & X^2 \end{bmatrix} \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix}^T. \end{aligned}$$

The second equality holds because

$$\begin{aligned} -V^T S X A^T - V^T Q^T X^2 A^T &= -V^T \mu(I + \theta E) A^T - V^T Q^T X^2 A^T \\ &= -V^T \mu(\theta E) A^T - V^T Q X^2 A^T, \end{aligned}$$

as  $AV = 0$  and  $Q$  is symmetric. Then, plugging (14) into the first diagonal block of the decomposition we obtained earlier, we have

$$\begin{aligned} M^T M &= \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix} \left( \begin{bmatrix} S^2 + 2\mu Q + \mu\theta(EQ + QE) + Q X^2 Q & -\mu\theta E - Q X^2 \\ -\mu\theta E - X^2 Q & X^2 \end{bmatrix} \right) \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix}^T \\ &= \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix} \left( \begin{bmatrix} S^2 + 2\mu Q + \mu\theta(EQ + QE) & -\mu\theta E \\ -\mu\theta E & 0 \end{bmatrix} + \begin{bmatrix} Q X^2 Q & -Q X^2 \\ -X^2 Q & X^2 \end{bmatrix} \right) \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix}^T \\ &= \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix} \left( \begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix} \begin{bmatrix} S^2 + 2\mu Q & -\mu\theta E \\ -\mu\theta E & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ -Q & I \end{bmatrix} \right) \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix}^T \\ &\quad + \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix} \left( \begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & X^2 \end{bmatrix} \begin{bmatrix} I & 0 \\ -Q & I \end{bmatrix} \right) \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix}^T \\ &= \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix} \begin{bmatrix} S^2 + 2\mu Q & -\mu\theta E \\ -\mu\theta E & X^2 \end{bmatrix} \begin{bmatrix} I & 0 \\ -Q & I \end{bmatrix} \begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix}^T. \end{aligned}$$

The first two matrices are nonsingular, so we can apply the Lemma 2, and thus we only need to study the middle matrix. Denote the middle matrix by  $\Psi$ . Observe that  $\Psi$  is almost the same as its counterpart in [14]. Subsequently, we have the following result regarding the spectral property of  $M^k$ .

**Lemma 3.** When  $(x, y, s) \in \mathcal{N}(\theta)$  and  $\theta \in \left(0, \min\left\{\frac{1}{3\sqrt{n}}, \frac{1}{4\|QVV^T\|_F + 1}\right\}\right)$ , the condition number of matrix  $M^k$  satisfies

$$\kappa_{M^k} = \mathcal{O}\left(\frac{(\omega^k)^2 + \mu^k \sigma_{\max}(Q)}{\mu^k} \kappa_{VAQ}\right),$$

where  $\kappa_{VAQ}$  is the condition number of the matrix  $\begin{bmatrix} V^T & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix}$ .

**Proof.** The proof is in Appendix B.  $\square$

Putting all of these together, we have the complexity for our IF-QIPM for LCQO problems.

**Theorem 1.** The IF-QIPM for LCQO problems stops with the final duality gap less than  $\epsilon$  in at most  $\mathcal{O}(\sqrt{n} \log(1/\epsilon))$  IPM iterations and, in each IPM iteration, the Newton direction can be obtained with complexity  $\tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}}\left(n\left(\frac{\bar{\omega}^2}{\epsilon} + \sigma_{\max}(Q)\right)\kappa_{VAQ} + n^2\right)$ , where  $\bar{\omega} = \max_k \omega^k$ .

**Proof.** The complexity bound for the IPM iterations comes from the result in [5]. According to (13), the complexity for obtaining the Newton direction is

$$\tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}}\left(\frac{n\omega^k \kappa_{M^k}}{\|M^k\|_F} + n^2\right).$$

Combining this with the result in Sec. 3.2.1, the bound in Lemma 3, and  $\mu^k \geq \epsilon$ , we have

$$\tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}}\left(\frac{n\omega^k \kappa_{M^k}}{\|M^k\|_F} + n^2\right) = \tilde{\mathcal{O}}_{n, \bar{\omega}, \frac{1}{\epsilon}}\left(n\left(\frac{\bar{\omega}^2}{\epsilon} + \sigma_{\max}(Q)\right)\kappa_{VAQ} + n^2\right).$$

$\square$

#### 4. Application in Support Vector Machine Problems

In this section, we discuss how to use our IF-QIPM to solve SVM problems. We show that our algorithm can solve  $\ell_1$ -norm soft margin SVM problems faster than any existing classical or quantum algorithms with respect to dimension.

The ordinary SVM problem works on a linearly separable dataset, in which the data points have binary labels. The ordinary SVM aims to find a hyperplane correctly separating the data points with a maximum margin. However, in practice, the data points are not necessarily linearly separable. To allow for mislabelling, the concept of a soft margin SVM was introduced in [22]. Let  $\{(\phi_i, \zeta_i) \in \mathbb{R}^m \times \{-1, +1\} | i = 1, \dots, n\}$  be the set of data points,  $\Phi$  be a matrix with the  $i$ th column being  $\phi_i$ , and  $Z$  be a diagonal matrix with the  $i$ th diagonal element being  $\zeta_i$ . The SVM problem with an  $\ell_1$ -norm soft margin can be formulated as below.

$$\begin{aligned} \min_{(\xi, w, t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}} \quad & \frac{1}{2} \|w\|_2^2 + C \|\xi\|_1 \\ \text{s.t.} \quad & \zeta_i (\langle w, \phi_i \rangle + t) \geq 1 - \xi_i, \quad i = 1, \dots, n \\ & \xi_i \geq 0, \quad i = 1, \dots, n. \end{aligned} \tag{15}$$

Here,  $(w, t)$  determines a hyperplane and  $C$  is a penalty parameter. In [9], the authors rewrote the SVM problem as a second-order conic optimization (SOCO) problem and used the quantum algorithm that they proposed to solve the resulting SOCO problem. They claim the complexity of their algorithm has  $\mathcal{O}(n^2)$  dependence on the dimension, which is better than any classical algorithm. However, the algorithm in [9] is invalid. Their

algorithm is an inexact infeasible-QIPM (II-QIPM), while they used the IPM complexity for the feasible-QIPM, which ignores at least  $\mathcal{O}(n^{1.5})$  dependence on  $n$ . They also missed the symmetrization of the Newton step, which is necessary for SOC problems and makes their Newton step invalid.

Aside from [9], some pure quantum algorithms for SVM problems are also proposed. In [23], the authors propose a pure quantum algorithm for SVM problems. They claim the complexity is  $\mathcal{O}(\kappa_{\text{eff}}^3 \epsilon^{-3} \log(mn))$ , where  $\kappa_{\text{eff}}$  is the condition number of a matrix involving the kernel matrix and  $\epsilon$  is the accuracy. In the worst case,  $\kappa_{\text{eff}} = \mathcal{O}(m)$ . Their complexity is worse than ours regarding the dependence of dimension and accuracy. In addition, their algorithm does not provide classical solutions. Namely, the solution is in the quantum machine and we cannot read or use it in a classical computer. However, our algorithm produces a classical solution.

To convert the problem into standard-form LCQO, we introduce  $(w_+, w_-) \in \mathbb{R}_+^m \times \mathbb{R}_+^m$ ,  $(t_+, t_-) \in \mathbb{R}_+ \times \mathbb{R}_+$ , and a slack variable  $\rho \in \mathbb{R}_+^n$ . Then, we can obtain the following formulation:

$$\begin{aligned} & \min_{w_+, w_-, t_+, t_-, \xi, \rho} \frac{1}{2} \|w_+ - w_-\|_2^2 + C\|\xi\|_1 \\ & \text{s.t. } \zeta_i(\langle w_+ - w_-, \phi_i \rangle + t_+ - t_-) + \xi_i - \rho_i = 1, i = 1, \dots, n \\ & \quad (\xi, w_+, w_-, t_+, t_-, \rho) \geq 0. \end{aligned}$$

This is a standard-form LCQO problem with non-negative variables  $(w_+, w_-, t_+, t_-, \xi, \rho) \in \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n$  and parameters

$$\begin{aligned} c &= \begin{bmatrix} 0_{2m+2} \\ Ce_n \\ 0_n \end{bmatrix} \\ Q &= \begin{bmatrix} I_{m \times m} & -I_{m \times m} & 0_{m \times (2+2n)} \\ -I_{m \times m} & I_{m \times m} & 0_{m \times (2+2n)} \\ 0_{(2+2n) \times m} & 0_{(2+2n) \times m} & 0_{(2+2n) \times (2+2n)} \end{bmatrix} \\ A &= [Z\Phi^T \quad -Z\Phi^T \quad Z \quad -Z \quad I_{n \times n} \quad -I_{n \times n}] \\ b &= e. \end{aligned}$$

Thus, we can use the proposed IF-QIPM for LCQO problems to solve the  $\ell_1$ -norm soft margin SVM problems and obtain an  $\epsilon$ -approximate solution with complexity

$$\tilde{\mathcal{O}}_{m,n,\bar{\omega},\frac{1}{\epsilon}} \left( (m+n)^{1.5} \left( \frac{\bar{\omega}^2}{\epsilon} + \sigma_{\max}(Q) \right) \kappa_{VAQ} + (m+n)^{2.5} \right).$$

This dependence on dimension is better than any existing quantum or classical algorithm.

## 5. Discussion

In this work, we present an IF-QIPM for LCQO problems by combining the IF-IPM framework proposed in [5] and the OSS system introduced in [14]. Our algorithm has  $n^{1.5}$  dependence on  $n$ , which is better than any existing algorithms for LCQO problems. The dependence on the accuracy is polynomial, which is worse than classic IPMs. Iterative refinement techniques might help to improve the dependence on the accuracy but they are beyond the discussion of this work.

**Author Contributions:** Conceptualization, Z.W. and T.T.; methodology, Z.W.; supervision, X.Y. and T.T.; validation, Z.W., Mohammadhossein Mohammadisiahroudi, B.A., X.Y. and T.T.; writing—original draft, Z.W.; writing—review and editing, Z.W., M.M., B.A., X.Y. and T.T. All authors have read and agreed to the published version of the manuscript.

**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.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The funder had no role in the design of the study; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

IF-IPM	Inexact Feasible Interior Point Method
IF-QIPM	Inexact Feasible Quantum Interior Point Methods
IPM	Interior Point Method
LCQO	Linearly Constrained Quadratic Optimization
LO	Linear Optimization
OSS	Orthogonal Subspace System
QIPM	Quantum Interior Point Method
QLSA	Quantum Linear System Algorithm
QTA	Quantum Tomography Algorithm
SOCO	Second-Order Conic Optimization
SVM	Support Vector Machine

## Appendix A. Block Encoding of the OSS System

In this section, we ignore the superscript  $k$  for simplicity. As described in Equation (9), we first block encode each of the matrices involved in (10). We assume that  $V$ ,  $A$ ,  $S$ , and  $X$  are given and are stored in a quantum accessible data structure (we ignore the complexity to store the classical information into the quantum machine). For the first matrix

$$M_1 = \begin{bmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{(n-m) \times n} & V^T & 0_{(n-m) \times n} \\ 0_{m \times n} & 0_{m \times n} & -A \end{bmatrix},$$

a

$$\left( \sqrt{\|V\|_F^2 + \|A\|_F^2}, \mathcal{O}(\text{poly log}(n)), \epsilon_1 \right)$$

-block-encoding of  $M_1$  can be implemented efficiently according to Lemma 50 from [18].

The second matrix

$$M_2 = \begin{bmatrix} 0_{n \times n} & 0_{n \times n} \\ S & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} \end{bmatrix}$$

is both one-row-sparse and one-column-sparse. By the definition of  $\omega$ , each element of  $M_2/\omega$  has an absolute value of at most 1. According to Lemma 48 in [18], a

$$(1, \mathcal{O}(\text{poly log}(n)), \epsilon_2)$$

-block-encoding of  $M_2/\omega$  can be implemented efficiently.

The third matrix

$$M_3 = \begin{bmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & Q & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & I_{n \times n} \end{bmatrix}$$

can be decomposed into

$$M_3 = \begin{bmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & Q & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \end{bmatrix} + \begin{bmatrix} 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & 0_{n \times n} & I_{n \times n} \end{bmatrix}.$$

Then, we can block encode the two matrices first, and then apply a linear combination to obtain  $M_3$ . In fact, a

$$(\|Q\|_F, \mathcal{O}(\text{poly log}(n)), \epsilon_3)$$

-block-encoding of the left matrix can be implemented efficiently according to Lemma 50 from [18] and a

$$(1, \mathcal{O}(\text{poly log}(n)), \epsilon_3)$$

-block-encoding of the right matrix can be implemented efficiently according to Lemma 48 in [18]. With the state-preparation cost of the linear combination coefficient vector  $(1, 1)$  neglected, a

$$(\|Q\|_F + 1, \mathcal{O}(\text{poly log}(n)), (\|Q\|_F + 1)\epsilon_3)$$

-block-encoding of  $M_3$  can be implemented efficiently according to Lemma 52 from [18].

The fourth matrix

$$M_4 = \begin{bmatrix} 0_{n \times n} & 0_{n \times n} \\ X & 0_{n \times n} \\ X & 0_{n \times n} \end{bmatrix}$$

is one-row-sparse and two-columns-sparse. After being scaled by  $\frac{1}{\omega}$ , each element of  $M_4/\omega$  has an absolute value of at most 1. According to Lemma 48 in [18], a

$$(\sqrt{2}, \mathcal{O}(\text{poly log}(n)), \epsilon_4)$$

-block-encoding of  $M_4/\omega$  can be implemented efficiently.

For the matrix multiplication  $M_3 M_4/\omega$ , a

$$(\sqrt{2}\|Q\|_F + \sqrt{2}, \mathcal{O}(\text{poly log}(n)), (\|Q\|_F + 1)(\sqrt{2}\epsilon_3 + \epsilon_4))$$

-block-encoding can be implemented efficiently according to Lemma 53 from [18].

For the linear combination  $M_2/\omega + M_3 M_4/\omega$ , the cost for the state-preparation of the coefficient vector  $(1, 1)$  is negligible and thus a

$$\left( \sqrt{2}\|Q\|_F + \sqrt{2} + 1, \mathcal{O}(\text{poly log}(n)), (\sqrt{2}\|Q\|_F + \sqrt{2} + 1)(\epsilon_3 + \frac{1}{\sqrt{2}}\epsilon_4) \right)$$

-block-encoding can be implemented efficiently according to Lemma 52 from [18].

For the matrix multiplication of  $M_1(M_2/\omega + M_3 M_4/\omega)$ , a

$$\begin{aligned} & \left( \sqrt{\|V\|_F^2 + \|A\|_F^2}(\sqrt{2}\|Q\|_F + \sqrt{2} + 1), \right. \\ & \quad \mathcal{O}(\text{poly log}(n)), \\ & \quad \left. \sqrt{\|V\|_F^2 + \|A\|_F^2}(\sqrt{2}\|Q\|_F + \sqrt{2} + 1)(\epsilon_3 + \frac{1}{\sqrt{2}}\epsilon_4) + (\sqrt{2}\|Q\|_F + \sqrt{2} + 1)\epsilon_1 \right) \end{aligned}$$

-block-encoding can be implemented efficiently according to Lemma 53 from [18].

Finally, considering that the complexity of the state-preparation of the vector

$$\left( \frac{\omega}{\sqrt{2}\|M\|_F}, \frac{\omega}{\sqrt{2}\|M\|_F} \right)$$

can be neglected, a

$$\begin{aligned} & \left( \frac{\sqrt{\|V\|_F^2 + \|A\|_F^2} \sqrt{2}\omega}{\|M\|_F} (\sqrt{2}\|Q\|_F + \sqrt{2} + 1), \right. \\ & \quad \left. \mathcal{O}(\text{poly log}(n)) \right), \\ & \frac{\sqrt{\|V\|_F^2 + \|A\|_F^2} \sqrt{2}\omega}{\|M\|_F} (\sqrt{2}\|Q\|_F + \sqrt{2} + 1)^2 \left( \sqrt{\|V\|_F^2 + \|A\|_F^2} (\epsilon_3 + \frac{1}{\sqrt{2}}\epsilon_4) + \epsilon_1 \right) \end{aligned}$$

-block-encoding of the coefficient matrix of system (8) can be implemented efficiently according to Lemma 52 from [18]. We can choose

$$\begin{aligned} \epsilon_1 &= \frac{\epsilon_{QLSA}}{\kappa_M^3} \frac{1}{2\mathcal{K}} \\ \epsilon_2 &= \frac{\epsilon_1}{2\sqrt{\|V\|_F^2 + \|A\|_F^2}} \\ \epsilon_3 &= \epsilon_2 \\ \epsilon_4 &= \sqrt{2}\epsilon_2, \end{aligned}$$

where  $\mathcal{K}$  depends on the initial data

$$\mathcal{K} = \sqrt{2}\sqrt{\|V\|_F^2 + \|A\|_F^2}(\sqrt{2}\|Q\|_F + \sqrt{2} + 1)^2.$$

Now, considering that the complexity for all of the block-encoding algorithms that we have used so far has poly-logarithmic dependence on the dimension and accuracy, and that, for  $i = 1, 2, 3, 4$

$$\mathcal{O}\left(\text{poly log}\left(\frac{1}{\epsilon_i}\right)\right) = \mathcal{O}(\text{poly log}(\kappa_M)),$$

the complexity for block encoding will be dominated by the complexity for QLSA because QLSA has linear dependence on  $\kappa_M$ , we can ignore the complexity of block encoding.

## Appendix B. Spectral Analysis for Matrix $\Psi$

In this section, we provide the spectral analysis for the matrix

$$\Psi = \begin{bmatrix} S^2 + 2\mu Q & -\mu\theta E \\ -\mu\theta E & X^2 \end{bmatrix}. \quad (\text{A1})$$

Just like in the previous section, for simplicity, we ignore the superscript  $k$ . We can perform the following decomposition:

$$\begin{bmatrix} S^2 + 2\mu Q & -\mu\theta E \\ -\mu\theta E & X^2 \end{bmatrix} = \begin{bmatrix} S^2 & -\mu\theta E \\ -\mu\theta E & X^2 \end{bmatrix} + \begin{bmatrix} 2\mu Q & 0 \\ 0 & 0 \end{bmatrix}.$$

Let us use the following notation:

$$\Psi_1 = \begin{bmatrix} S^2 & -\mu\theta E \\ -\mu\theta E & X^2 \end{bmatrix}$$

$$\Psi_2 = \begin{bmatrix} 2\mu Q & 0 \\ 0 & 0 \end{bmatrix}.$$

It can be proven that  $\Psi_1$  is positive definite. The majority of the proof of this conclusion comes from the paper [14]. For the reader's convenience, we provide the complete proof here.

Matrix  $\Psi_1$  is a block diagonal matrix, with all four blocks being diagonal matrices. Thus, we can easily compute the eigenvalues using the characteristic polynomial

$$\det(\Psi_1 - qI) = \det \left( (X^2 - qI)(S^2 - qI) - \theta^2 \mu^2 E^2 \right)$$

$$= \prod_{i=1}^n \left( (x_i^2 - q)(s_i^2 - q) - \theta^2 \mu^2 E_{ii}^2 \right).$$

Clearly,  $\det(\Psi_1 - qI) = 0$  gives  $n$  quadratic equations and each quadratic equation gives two eigenvalues. The two eigenvalues from the  $i$ th quadratic equation are

$$q_{i+} = \frac{1}{2} \left( (x_i^2 + s_i^2) + \sqrt{(x_i^2 + s_i^2)^2 - 4x_i^2 s_i^2 + 4\theta^2 \mu^2 E_{ii}^2} \right)$$

and

$$q_{i-} = \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 - 4x_i^2 s_i^2 + 4\theta^2 \mu^2 E_{ii}^2} \right).$$

Recalling the definition of  $E$  in (14), we can write

$$q_{i-} = \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 - 4x_i^2 s_i^2 + 4(x_i s_i - \mu)^2} \right)$$

$$= \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 + 4(x_i s_i - \mu + x_i s_i)(x_i s_i - \mu - x_i s_i)} \right)$$

$$= \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 - 4\mu(2x_i s_i - \mu)} \right)$$

$$= \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 - 4\mu(2\theta\mu E_{ii} + \mu)} \right).$$

One can verify that the square root always exists because

$$(x_i^2 + s_i^2)^2 - 4\mu(2x_i s_i - \mu) \geq 4(x_i s_i)^2 - 4\mu(2x_i s_i) + 4\mu^2$$

$$= 4(x_i s_i - \mu)^2$$

$$\geq 0.$$

With  $\theta \in \left(0, \min\left\{\frac{1}{3\sqrt{n}}, \frac{1}{4\|QVV^T\|_F + 1}\right\}\right)$ , we have

$$\begin{aligned}
q_{i-} &\geq \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 - 4\mu(2\theta\mu E_{ii} + \mu)} \right) \\
&\geq \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 - 4\mu(-2\mu \frac{1}{3\sqrt{n}} + \mu)} \right) \\
&= \frac{1}{2} \left( (x_i^2 + s_i^2) - \sqrt{(x_i^2 + s_i^2)^2 - \frac{4}{3}\mu^2} \right) \\
&= \frac{1}{2} \frac{\frac{4}{3}\mu^2}{(x_i^2 + s_i^2) + \sqrt{(x_i^2 + s_i^2)^2 - \frac{4}{3}\mu^2}} \\
&\geq \frac{1}{2} \frac{\frac{4}{3}\mu^2}{(x_i^2 + s_i^2) + \sqrt{(x_i^2 + s_i^2)^2}} \\
&= \frac{\mu^2}{3(x_i^2 + s_i^2)} \\
&> 0.
\end{aligned}$$

This means that matrix  $\Psi_1$  is positive definite and its eigenvalues coincide with its singular values because  $\Psi_1$  is also real and symmetric. Analogously, we have

$$\begin{aligned}
q_{i+} &= \frac{1}{2} \left( (x_i^2 + s_i^2) + \sqrt{(x_i^2 + s_i^2)^2 - 4\mu(2\theta\mu E_{ii} + \mu)} \right) \\
&\leq \frac{1}{2} \left( (x_i^2 + s_i^2) + (x_i^2 + s_i^2) + 2\mu\sqrt{(2\theta E_{ii} + 1)} \right) \\
&\leq \frac{1}{2} \left( (x_i^2 + s_i^2) + (x_i^2 + s_i^2) + 2\mu\sqrt{2} \right) \\
&= (x_i^2 + s_i^2) + \sqrt{2}\mu.
\end{aligned}$$

Thus, the condition number of  $\Psi$  satisfies

$$\begin{aligned}
\kappa(\Psi) &\leq \frac{\sigma_{\max}(\Psi_1) + \sigma_{\max}(\Psi_2)}{\sigma_{\min}(\Psi_1) + \sigma_{\min}(\Psi_2)} \\
&= \frac{\max_i q_{i+} + \sigma_{\max}(\Psi_2)}{\min_j q_{j-} + \sigma_{\min}(\Psi_2)} \\
&\leq \frac{\max_i \{x_i^2 + s_i^2\} + \sqrt{2}\mu + 2\mu\sigma_{\max}(Q)}{\min_j \frac{\mu^2}{3(x_i^2 + s_i^2)}} \\
&= \frac{3 \max_i \{x_i^2 + s_i^2\} \left( \max_i \{x_i^2 + s_i^2\} + \sqrt{2}\mu + 2\mu\sigma_{\max}(Q) \right)}{\mu^2} \\
&\leq \frac{3\omega^2 \left( \omega^2 + \sqrt{2}\mu + 2\mu\sigma_{\max}(Q) \right)}{\mu^2},
\end{aligned}$$

where the last inequality comes from the definition of  $\omega$ . Since  $\omega^2 \geq x_i s_i \geq (1 - \theta)\mu$ , we have

$$\kappa(\Psi) = \mathcal{O}\left(\frac{\omega^2(\omega^2 + \mu\sigma_{\max}(Q))}{\mu^2}\right).$$

Using Lemma 2, we can also bound the condition number of matrix  $M$  by

$$\begin{aligned}\kappa_M &= \sqrt{\kappa(M^T M)} \\ &\leq \sqrt{\kappa(\Psi)} \kappa_{VAQ} \\ &= \mathcal{O}\left(\frac{(\omega^2 + \mu\sigma_{\max}(Q))}{\mu} \kappa_{VAQ}\right).\end{aligned}$$

## References

1. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: New York, NY, USA, 1999.
2. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; Haussler, D., Ed.; Association for Computing Machinery: NY, NY, USA, 1992; pp. 144–152.
3. Roos, C.; Terlaky, T.; Vial, J.P. *Theory and Algorithms for Linear Optimization: An Interior Point Approach*; John Wiley & Sons: New York, NY, USA, 1997.
4. Pólik, I.; Terlaky, T. Interior point methods for nonlinear optimization. In *Nonlinear Optimization*; Gianni Di Pillo, F.S., Ed.; Springer: New York, NY, USA, 2010; pp. 215–276.
5. Gondzio, J. Convergence analysis of an inexact feasible interior point method for convex quadratic programming. *SIAM J. Optim.* **2013**, *23*, 1510–1527. [\[CrossRef\]](#)
6. Lu, Z.; Monteiro, R.D.; O’Neal, J.W. An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming. *SIAM J. Optim.* **2006**, *17*, 287–310. [\[CrossRef\]](#)
7. Bunch, J.R.; Parlett, B.N. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM J. Numer. Anal.* **1971**, *8*, 639–655. [\[CrossRef\]](#)
8. Schuld, M.; Sinayskiy, I.; Petruccione, F. Prediction by linear regression on a quantum computer. *Phys. Rev. A* **2016**, *94*, 022342. [\[CrossRef\]](#)
9. Kerenidis, I.; Prakash, A.; Szilágyi, D. Quantum algorithms for second-order cone programming and support vector machines. *Quantum* **2021**, *5*, 427. [\[CrossRef\]](#)
10. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Kerenidis, I.; Prakash, A. A quantum interior point method for LPs and SDPs. *ACM Trans. Quantum Comput.* **2020**, *1*, 1–32. [\[CrossRef\]](#)
12. Mohammadisahroudi, M.; Fakhimi, R.; Terlaky, T. Efficient use of quantum linear system algorithms in interior point methods for linear optimization. *arXiv* **2022**, arXiv:2205.01220.
13. Augustino, B.; Nannicini, G.; Terlaky, T.; Zuluaga, L.F. Quantum interior point methods for semidefinite optimization. *arXiv* **2021**, arXiv:2112.06025.
14. Mohammadisahroudi, M.; Fakhimi, F.; Wu, Z.; Terlaky, T. *An Inexact Feasible Interior Point Method for Linear Optimization with High Adaptability to Quantum Computers*; Technical Report: 21T-006; Department of ISE, Lehigh University: Bethlehem, PA, USA, 2021.
15. Kojima, M.; Mizuno, S.; Yoshise, A. A polynomial-time algorithm for a class of linear complementarity problems. *Math. Program.* **1989**, *44*, 1–26. [\[CrossRef\]](#)
16. Monteiro, R.D.; Adler, I. Interior path following primal-dual algorithms. part II: Convex quadratic programming. *Math. Program.* **1989**, *44*, 43–66. [\[CrossRef\]](#)
17. Goldfarb, D.; Liu, S. An  $O(n^3L)$  primal interior point algorithm for convex quadratic programming. *Math. Program.* **1990**, *49*, 325–340. [\[CrossRef\]](#)
18. Gilyén, A.; Su, Y.; Low, G.H.; Wiebe, N. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. *arXiv* **2018**, arXiv:1806.01838.
19. Chakraborty, S.; Gilyén, A.; Jeffery, S. The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation. *arXiv* **2018**, arXiv:1804.01973.
20. van Apeldoorn, J.; Cornelissen, A.; Gilyén, A.; Nannicini, G. Quantum tomography using state-preparation unitaries. *arXiv* **2022**, arXiv:2207.08800.
21. Horn, R.A.; Johnson, C.R. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 2012.
22. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [\[CrossRef\]](#)
23. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [\[CrossRef\]](#) [\[PubMed\]](#)