Gulf Journal of Mathematics
Vol 23, Issue 1 (2026) 1-15

ISSN: 2309-4966
https://doi.org/10.56947/q3b9tb87

A CONVEX QUADRATIC PROGRAMMING BASED ON A NOVEL TYPE OF PARAMETERIZED KERNEL FUNCTION

GUEMMAZ ABDERRAHIM[1*] and BOUNIBANE BACHIR[2]

Abstract. This study introduces primal-dual interior-point methods for convex quadratic programming, based on a novel parameterized hyperbolic kernel function. Under certain conditions and by using simple analytical approaches, along with a specific choice of a key parameter, the proposed method achieves one of the best known iteration efficiencies for large-update methods. Numerical experiments are also presented to demonstrate the practical effectiveness of the approach.

Keywords. Quadratic programming, Kernel function, Interior-point methods, Iteration bounds.
2020 Mathematics Subject Classification. 90C34; 90C40.

1. Introduction

Convex Quadratic Programs (CQP) are optimization problems characterized by a convex quadratic objective function subject to linear constraints. They arise in diverse fields such as finance, agriculture, economics, optimal control, and geometric optimization, and frequently appear as subproblems in sequential quadratic programming frameworks.

Since the seminal work of Karmarkar [15], interior-point methods (IPMs) have been extensively developed for various convex optimization problems, including Linear Complementarity Problems (LCPs), Second-Order Cone Optimization (SOCO), Semidefinite Optimization (SDO), and more recently Convex Quadratic Semidefinite Optimization (CQSDO).

Primal-dual IPMs based on the logarithmic barrier function were introduced by Roos et al. [20]. A significant advancement was made by Bai et al. [3], who introduced analytical techniques for evaluating the complexity of primal-dual IPMs using a novel class of kernel functions. Their work established the complexity bounds $\mathcal{O}(\sqrt{n}\log\frac{n}{\varepsilon})$ for small-update methods and $\mathcal{O}(\sqrt{n}\log n\log\frac{n}{\varepsilon})$ for large-update methods, which remain the best-known bounds to date.

Kernel functions play a central role in both the analysis of IPMs and the formulation of search directions. Peng et al. [17] first introduced primal-dual

Date: Received: Mar 9, 2026; Revised: Apr 7, 2026; Accepted: Apr 23, 2026.
* Corresponding author

IPMs for linear programming based on self-regular kernel functions, achieving the same best-known iteration bounds. Subsequently, various kernel functions have been proposed for different optimization settings. For instance, Zhang [21] introduced a kernel function for convex quadratic semidefinite optimization, while Peyghami et al. [18, 19] developed kernel functions for linear optimization and convex quadratic semidefinite optimization, respectively. Achache [1] analyzed a kernel function for semidefinite optimization, and recently, Boudjellal et al. [7, 22] introduced bi-parameterized hyperbolic kernel functions for SDO.

El Ghami et al. [14] introduced the first kernel function with a trigonometric barrier term for linear optimization, later extended by Bouafia et al. [5]. Bouafia [6] further extended the approach to linear fractional programming.

Inspired by these developments, this work extends the primal-dual IPMs for linear optimization (LO) based on a new parameterized kernel function, originally introduced in [9], to convex quadratic programming (CQP). The kernel function is defined as:

$$\psi_a(t) = \frac{t^2-1}{2} - \int_1^t a^{\left(\frac{1}{x}-1\right)}dx, \tag{1.1}$$

While the properties of $\psi_a(t)$ were analyzed in [9] for LO, their application to CQP is nontrivial due to the presence of the matrix Q in the KKT conditions and the scaled system. The contribution of this paper is to demonstrate that the same kernel function yields the best-known iteration complexity $\mathcal{O}(\sqrt{n}\log n\log\frac{n}{\varepsilon})$ for large-update IPMs in the CQP setting. Numerical experiments are presented to illustrate the practical performance of the proposed algorithm.

The paper is organized as follows: Section 2 presents the problem statement and recalls the essential principles of interior-point methods applied to constrained quadratic programming. We look into a few aspects of the new kernel function (1.1) for primal-dual interior point techniques in Section 3. The complexity findings of our algorithm are then shown in Section 4, which also includes the best-known iteration bounds for small- and large-update approaches. The numerical outcomes of our experiments are shown in Section 5. Finally, some concluding remarks are given in Section 6.

The following notational conventions are used throughout this paper: $\mathbb{R}^n$, R^n_+ and $\mathbb{R}^n_{++}$ denotes the set of n-dimensional real space, nonnegative space and positive space, respectively. In the case of $x, s \in \mathbb{R}^n$, as well as $x \circ s = (x_1 s_1, ..., x_n s_n)^T$ and $x^T s$ denote the componentwise Hadamard product and the Euclidean scalar product of two vectors x and s, respectively. We denote by $\|x\|$ the 2-norm of x. For any f, $g : \mathbb{R}^n_{++} \to \mathbb{R}^n_{++}$ we have $f(x) = O(g(x))$ and $f(x) = \Theta(g(x))$ if $f(x) \leq c_1 g(x)$ and $c_2 g(x) \leq f(x) \leq c_3 g(x)$, respectively, for some positive constants c_1, c_2 and c_3.

2. Preliminaries and problem setting

The standard CQP problem (P) with its dual problem (D) are defined as

$$\min\left\{c^T x + \frac{1}{2}x^T Q x : Ax = b, x \geq 0\right\} \tag{P}$$

where $Q \in \mathbb{R}^{n\times n}$ is a given positive semidefinite matrix, $A \in \mathbb{R}^{m\times n}$ is a predefined matrix, $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. The dual problem of (P) is given by

$$\max\left\{b^T y - \frac{1}{2}x^T Q x : A^T y + s - Qx = c, s \geq 0\right\} \tag{D}$$

where $s \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. CQP reduces to LO if Q = 0; that is to say, CQP is the generalization of LO. In this study, we assume that (P) and (D) meet the following requirements. Interior-point-condition (IPC), i.e., there exist a point (x_0, y_0, s_0) such that

$$\left\{Ax_0 = b, A^T y_0 + s_0 - Qx_0 = c, x_0 \geq 0, s_0 \geq 0\right\} \tag{2.1}$$

$rank(A) = m < n$, i.e., the matrix A has full row rank. Determining optimal solutions for the primal (KKT) criteria for (P) and dual (D) problems is equivalent to solving the system of Karush-Kuhn-Tucker (KKT) optimality conditions

$$\begin{cases} Ax = b, \quad x \geq 0 \\ A^T y + s - Qx = c, \quad s \geq 0 \\ xs = 0 \end{cases} \tag{2.2}$$

The basic idea of primal-dual ($IPMs$) is to replace the complementarity condition in (2.2) with the parameterized equation $xs = \mu e$. This provides the next system.

$$\begin{cases} Ax = b, \quad x \geq 0 \\ A^T y + s - Qx = c, \quad s \geq 0 \\ xs = \mu e \end{cases} \tag{2.3}$$

where $\mu > 0$ and $e = (1, \ldots, 1)^T$ is the vector of all-ones. It is well known that for each $\mu > 0$, there exists a unique solution denoted by $(x(\mu), y(\mu), s(\mu))$, where $x(\mu)$ is known as the μ-center of (P) and $(y(\mu), s(\mu))$ is known as the μ-center of (D). As μ-centers approaches zero, the limit exists and converges to x and (y, s), which are the solutions of (P) and (D), respectively. The set of μ-centers is called the central path of (P) and (D). From a theoretical point of view, the (IPC) can be assumed without loss of generality. We may assume that $\mu_0 = 1$ and $x_0 = s_0 = e$ to simplify the theoretical contributions (see Roos et al. in [20]).

Now, to get the new feasible iterates we apply Newton's method in (2.3), we obtain Newton's direction. Due to ASSUMPTION (2), $(\Delta x, \Delta y, \Delta s)$ is the unique solution of the following linear system.

$$\begin{cases} A\Delta x = 0 \\ A^T \Delta y + \Delta s - Q\Delta x = 0 \\ s\Delta x + x\Delta s = \mu e - xs \end{cases} \tag{2.4}$$

At this stage, to streamline the analysis, we introduce the scaled vector v and the scaling directions (d_x, d_s) as follows

$$d_x = \frac{v\Delta x}{x}, \quad d_s = \frac{v\Delta s}{s} \quad and \quad v = \sqrt{\frac{xs}{\mu}}. \tag{2.5}$$

Consequently, we may describe the system (2.4) as the following equivalent system by using the notation (2.5).

$$\begin{cases} \overline{A}d_x = 0 \\ \overline{A}^T\Delta y + d_s - \overline{Q}d_x = 0 \\ d_x + d_s = v^{-1} - v \end{cases} \tag{2.6}$$

where $\overline{A} = AV^{-1}X$, $\overline{Q} = S^{-1}VQV^{-1}X$, $V = diag(v)$, $X = diag(x)$ and $S = diag(s)$. The new iterate is then computed as follows:

$$x_+ = x + \alpha\Delta x, \quad y_+ = y + \alpha\Delta y, \quad s_+ = s + \alpha\Delta s. \tag{2.7}$$

It is simple to verify that the last equation in (2.6) has a right-hand side equal to the negative gradient of the logarithmic barrier function $\Psi_l(v) : \mathbb{R}^n_{++} \to \mathbb{R}^n_+$, as defined by

$$\Psi_l(v) = \Psi_l(x, s; \mu) = \sum_{i=1}^{n} \psi_l(v_i) = \sum_{i=1}^{n} \left(\frac{v_i^2 - 1}{2} - \log v_i \right) \tag{2.8}$$

where the kernel function of the classical logarithmic barrier function is denoted by $\psi_l(t)$. Thus, we arrive at the following system

$$\begin{cases} \overline{A}d_x = 0 \\ \overline{A}^T\Delta y + d_s - \overline{Q}d_x = 0 \\ d_x + d_s = -\nabla\Psi_l(v) \end{cases} \tag{2.9}$$

As a result, in addition to proximity measure $\delta(v)$, the logarithmic barrier function $\Psi_l(v)$ is considered a proximity function to measure the distance between any iteration and the μ-center: The definition of $\delta(v) : \mathbb{R}^n_{++} \to \mathbb{R}^n_+$ is as follows

$$\delta(x, s; \mu) = \delta(v) = \frac{1}{2}\|d_x + d_s\| = \frac{1}{2}\|\nabla\Psi_l(v)\| \tag{2.10}$$

since d_x and d_s are orthogonal, due to (2.9), thus we obtain

$$d_x = d_s = 0 \Leftrightarrow \nabla\Psi_l(v) = 0 \Leftrightarrow v = e \Leftrightarrow \Psi_l(v) = 0$$

this says that the desired result $(x, s) = (x(\mu), s(\mu))$.

This paper aims to extend primal-dual interior-point methods $(IPMs)$ for linear optimization (LO), based on the parametric kernel function introduced in [9] , to convex quadratic programming (CQP). Additionally, it replaces $\psi_l(t)$ with a new kernel function $\psi_a(t)$ and $\Psi_l(v)$ defined in (2.8) with a new barrier function $\Psi_a(v)$, both of which will be defined in section 2,

It is abundantly evident from the foregoing justifications that the algorithm described above operates as follows: It starts with the parameters $\epsilon > 0$, $\tau \leq 1$, $0 < \theta < 1$, the parameter $\mu > 0$, a beginning point (x, y, s) and the scaled vector v (should be selected in a manner that the algorithm's necessary number of iterations is as low as feasible). To identify the search directions $(\Delta x, \Delta y, \Delta s)$ using the two systems (2.4) and (2.6), compute the step size α and then apply (2.7) to generate new iterations. As a result, in each outer iteration, we update the values of μ and v by the factor $1 - \theta$. If the criterion $\Psi_a(v) > \tau$ is fulfilled, we enter the inner iteration. We continue in this manner until μ is small enough and v agrees on $\Psi_a(v) \leq \tau$, at which time we claim that we have obtained the best solution to the convex quadratic programming.

INPUT :
A proximity function $\Psi_a(v)$; an accuracy parameter $\varepsilon > 0$; a threshold parameter $\tau \geq 1$; a fixed barrier update parameter $0 < \theta < 1$; initial point (x_0, y_0, s_0) and the parameter $\mu_0 > 0$.
ITERATION :
BEGIN
$x = x_0$; $y = y_0$; $s = s_0$; $\mu = \mu_0$; $v = \sqrt{\frac{xs}{\mu}}$.
WHILE $(n\mu \geq \varepsilon)$ **DO**
BEGIN
update of μ and v : $\mu = (1-\theta)\mu$; $v = \frac{v}{\sqrt{1-\theta}}$.
WHILE $(\Psi_a(v) > \tau)$ **DO**
BEGIN
solve (2.6) via (2.4) to find search directions;
determine a step size $\alpha > 0$;
$(x, y, s) = (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$;
$v = \sqrt{\frac{xs}{\mu}}$.
END.
END.
END.

FIGURE 1. GENERIC INTERIOR-POINT ALGORITHM FOR *CQP*.

3. THE NEW KERNEL FUNCTION AND ITS PROPERTIES

In this section, we introduce a novel kernel function and analyze its key properties, which underpin the complexity analysis presented in subsequent discussions. First, we must recall the following definition.

Definition 3.1. $\psi(t) : \mathbb{R}_{++} \to \mathbb{R}_+$ is a kernel function if ψ is twice differentiable and satisfies the following conditions

Cond1. $\psi(1) = \psi'(1) = 0$
Cond2. $\psi''(t) > 0, \forall t > 0$.
Cond3. $\lim_{t \longrightarrow 0^+} \psi(t) = \lim_{t \longrightarrow +\infty} \psi(t) = +\infty$

The first three derivatives of (1.1) with respect to t are as follows:

$$\psi_a'(t) = t - a^{\left(\frac{1}{t}-1\right)}, \tag{3.1}$$

$$\psi_a''(t) = 1 + \frac{\log a}{t^2} a^{\left(\frac{1}{t}-1\right)}, \tag{3.2}$$

$$\psi_a'''(t) = -\frac{\log a\,(2t + \log a)}{t^4} a^{\left(\frac{1}{t}-1\right)}. \tag{3.3}$$

It is simple to confirm the three requirements of definition. 3.1 and determine that the function described by (1.1) is a kernel function by utilizing (1.1), (3.1) and (3.2). It is evident from the second criterion (**Con2**) that $\psi_a(t)$ is coercive and possesses the barrier quality. Additionally, (3.2) indicate that we have for all $t > 0$

$$\psi_a''(t) \geq 1 > 0. \tag{3.4}$$

We may now discuss the fundamental characteristics of (1.1).

Lemma 3.2. [10] *Let the function $\psi_a(t)$ be defined as in (1.1). Then, we have*

$$\begin{aligned} t\psi_a''(t) + \psi_a'(t) &> 0, \quad t < 1, && (3.5)\\ t\psi_a''(t) - \psi_a'(t) &> 0, \quad t > 1, && (3.6)\\ \psi_a'''(t) &< 0 \quad t > 0, && (3.7)\\ 2\left(\psi_a''(t)\right)^2 - \psi_a'(t)\psi_a'''(t) &> 0, \quad t < 1, && (3.8)\\ \psi_a''(t)\psi_a'(\beta t) - \beta\psi_a'(t)\psi_a''(\beta t) &> 0, \ t > 1, \ \beta > 1. && (3.9) \end{aligned}$$

In conclusion, we have $\psi_a(t)$ is an eligible kernel function. As an observation, we give some results about the new kernel function.

Lemma 3.3. [9] *Given $\psi_a(t)$, we have the following results confirmed.*

$$\begin{aligned} \frac{1}{2}(t-1)^2 &\leq \psi_a(t) \leq \frac{1}{2}\psi_a'(t)^2, \ t > 0, && (3.10)\\ \psi_a(t) &\leq \frac{1}{2}\psi_a''(1)(t-1)^2, \ t \geq 1, && (3.11) \end{aligned}$$

Lemma 3.4. [10] *Let $\beta \geq 1$. Then*

$$\psi_a(\beta t) \leq \psi_a(t) + \frac{1}{2}\left(\beta^2 - 1\right)t^2.$$

Defining

$$\psi_b(t) = -\int_1^t a^{\left(\frac{1}{t}-1\right)} < 0,$$

In this, we examine the complexity of the LO interior-point algorithm for large updates. The algorithm is analyzed using the norm-based proximity measure.

Lemma 3.5. [9] *Let $\varrho : [0, +\infty) \longrightarrow [1, +\infty)$ be the inverse function of $\psi_a(t)$ for $t \geq 1$ and $\rho : [0, +\infty) \longrightarrow (0, 1]$ the inverse function of $\frac{-1}{2}\psi_a'(t)$ for $(0, 1]$, we have:*

$$\begin{aligned} \sqrt{2s+1} &\leq \varrho(s) \leq \sqrt{2s} + 1 \ s \geq 0, && (3.12)\\ \rho(z) &\geq \frac{1}{1 + \frac{\log(1+2z)}{\log a}} \ z \geq 0. && (3.13) \end{aligned}$$

The following lemma establishes a relationship between (2.10) and $\Psi_a(v)$.

Lemma 3.6. [10] *Let $\delta(v)$ be defined as in (2.10). Then we have*

$$\delta(v) \geq \sqrt{\frac{\Psi_a(v)}{2}}.$$

We now provide the theorem, which provides a prediction of the effect of a μ-update on the value of $\Psi(v)$.

Theorem 3.7. [3] *Let $\varrho(s) : [0, +\infty[\rightarrow [1, +\infty[$ be the inverse function of $\psi(t)$ defined in (1.1) for all $t \geq 1$. Then we have for any positive vector v and any $\beta \geq 1$ that*

$$\Psi(\beta v) \leq n\psi\left(\beta\varrho\left(\frac{\Psi(v)}{n}\right)\right). \tag{3.14}$$

We compute the upper bound for the effect of a μ-update on the value of $\Psi_a(v)$ in the following lemma.

Lemma 3.8. [9] *Let $0 \leq \theta < 1$ and $v_+ = \frac{v}{\sqrt{1-\theta}}$. If $\Psi(v) \leq \tau$ then we have:*

$$\Psi(v_+) \leq \frac{1 + \log a}{1-\theta}\left(\theta\sqrt{n} + \sqrt{2\tau}\right)^2, \tag{3.15}$$

Denote

$$\bar{\Psi}_0 = (1 + \log a)\frac{n\theta^2 + 2\tau + 2\theta\sqrt{2n\tau}}{1-\theta} \tag{3.16}$$

We'll utilize $\bar{\Psi}_0$ for the upper bounds of $\Psi(v)$ for large-update methods throughout the algorithm.

Remark 3.9. For large-update methods we take $\tau = O(n)$ and $\theta = \Theta(1)$.

4. COMPLEXITY ANALYSIS

In this section, we derive the iteration bounds for large-update methods based on our kernel function (1.1). Before analyzing the reduction of the proximity function during an inner iteration, a default step size α must first be determined. Each iteration of Algorithm.1 generates updated primal-dual variables (x_+, y_+, s_+), while the parameter μ remains constant throughout the inner iteration. Using (2.5) and (2.7), we establish the relationship between successive iterates and quantify the decrease in the proximity function.

$$x_+ = x + \alpha\Delta x = x\left(e + \alpha\frac{\Delta x}{x}\right) = x\left(e + \alpha\frac{d_x}{v}\right) = \frac{x}{v}(v + \alpha d_x)$$
$$s_+ = s + \alpha\Delta s = s\left(e + \alpha\frac{\Delta s}{s}\right) = s\left(e + \alpha\frac{d_s}{v}\right) = \frac{s}{v}(v + \alpha d_s).$$

So, we have

$$v_+ = \sqrt{\frac{x_+ s_+}{\mu}} = \sqrt{(v + \alpha d_x)(v + \alpha d_s)}, \tag{4.1}$$

from (3.6) and (4.1), it is clear that

$$\Psi_a(v_+) = \Psi_a\left(\sqrt{(v + \alpha d_x)(v + \alpha d_s)}\right) \quad \leq \quad \frac{1}{2}\left(\Psi_a(v + \alpha d_x) + \Psi_a(v + \alpha d_s)\right).$$

For $\alpha > 0$, we define

$$f(\alpha) = \quad \Psi_a(v_+) - \Psi_a(v) \tag{4.2}$$
$$f_1(\alpha) = \quad \frac{1}{2}\left(\Psi_a(v + \alpha d_x) + \Psi_a(v + \alpha d_s)\right) - \Psi_a(v). \tag{4.3}$$

Taking the first two derivatives of $f_1(\alpha)$ with respect to α, we have:

$$f_1'(\alpha) = \frac{1}{2}\sum_{i=1}^{n}\left(\psi_a'(v_i + \alpha d_{x_i})d_{x_i} + \psi_a'(v_i + \alpha d_{s_i})d_{s_i}\right) \tag{4.4}$$

and

$$f_1^{''}(\alpha) = \frac{1}{2}\sum_{i=1}^{n}\left(\psi_a^{''}(v_i + \alpha d_{x_i})d_{x_i}^2 + \psi_a^{''}(v_i + \alpha d_{s_i})d_{s_i}^2\right). \tag{4.5}$$

Therefore

$$f(\alpha) \leq f_1(\alpha), \quad f(0) = f_1(0) = 0$$

and by using the second equation in (2.9) and (2.10), we have

$$f_1^{'}(0) = \frac{1}{2}\langle \nabla\Psi_a(v), (d_x + d_s)\rangle = -\frac{1}{2}\|\nabla\Psi_a(v)\|^2 = -2\delta(v)^2. \tag{4.6}$$

notation We denote $v_{\min} = \min_{i\in\{1,\dots,n\}} v_i$, $\quad \delta = \delta(v)$ and $\quad \Psi_a = \Psi_a(v)$.

We assume that $\Psi_a \geq \tau \geq 1$ for the remainder of this study. Using Lemma 3.6, we obtain $\delta \geq \sqrt{\tau} \geq 1$. We shall reiterate the four axioms we will employ before presenting our results (for proof, it is possible to resort to [3]).

Lemma 4.1. [3] *Let $f_1(\alpha)$ be defined as in (4.3) and δ as in (2.10). Then*

$$f_1^{''}(\alpha) \leq 2\delta^2\psi_a^{''}\left(v_{min} - 2\alpha\delta\right). \tag{4.7}$$

Lemma 4.2. [3] *If α satisfies the inequality*

$$-\psi_a^{'}(v_{min} - 2\alpha\delta) + \psi_a^{'}(v_{min}) \leq 2\delta, \tag{4.8}$$

then, we obtain $f_1^{'}(\alpha) \leq 0$.

Lemma 4.3. [3] *Let $\rho(s) : [0, +\infty[\rightarrow]0, 1]$ denote the inverse function of $-\frac{1}{2}\psi_a^{'}(t)$. The largest step size α that satisfies (4.8) is given by*

$$\overline{\alpha} := \frac{\rho(\delta) - \rho(2\delta)}{2\delta}.$$

Lemma 4.4. [9] *Let $\overline{\alpha}$ be as defined in (4.3). Then*

$$\bar{\alpha} \geq \frac{1}{8\delta\log a\left[1 + \frac{\log(1+4\delta)}{\log a}\right]^2}.$$

The default step size $\alpha^\star$ ($\alpha^\star \leq \overline{\alpha}$) for Algorithm1 is defined as follows

$$\alpha^\star = \frac{1}{8\delta\left[1 + \frac{\log(1+4\delta)}{\log a}\right]^2\log a}, \tag{4.9}$$

Role of the parameter a. From (4.9), we observe that $\alpha^\star$ is inversely proportional to $\log a$. Larger values of a lead to smaller step sizes, which may increase the number of iterations. Therefore, a must be chosen carefully to balance the reduction rate and the iteration complexity.

Now, we use $\alpha^\star$, to indicate the reduction in the proximity function Ψ_a during an inner iteration. We arrived at the results shown below to do this.

Lemma 4.5. [17] *Let $h(t)$ be a twice differentiable convex function with $h(0) = 0$, $h^{'}(0) < 0$ and let $h(t)$ attain its global minimum at its stationary point $t^* > 0$. If $h^{''}(t)$ is increasing with respect to t, then one has for any $t \in [0, t^*]$*

$$h(t) \leq \frac{th^{'}(0)}{2}.$$

Lemma 4.6. [23] *If the step size α satisfies $\alpha \leq \overline{\alpha}$, then*

$$f(\alpha) \leq -\alpha\delta^2. \tag{4.10}$$

The following theorem is derived from Lemma (4.6) and (4.9).

Theorem 4.7. [3] *With $\alpha^\star$ being the default step size, as given by (4.9), one has*

$$f(\alpha^\star) \leq -\frac{\delta^2}{\psi_a''\left(\rho(2\delta)\right)}. \tag{4.11}$$

Lemma 4.8. [9] *Let $\alpha^\star$ defined as in (4.9). Then we have*

$$f(\alpha^\star) \quad \leq \quad -\frac{\sqrt{\Psi}}{16\log a\left[1+\frac{\log\left(1+2\sqrt{\Psi_0}\right)}{\log a}\right]^2}. \tag{4.12}$$

At this stage, we count how many inner iterations Algorithm 1 requires to achieve the situation $\Psi_a \leq \tau$. Let Ψ_0 denote the value of Ψ_a after μ-update and subsequent values in the same outer iteration are denoted as Ψ_{a_k}, $k = 1, \ldots, K$, where K denotes the total number of inner iterations in the outer iteration and we use the following lemma to do so.

Lemma 4.9. [17] *Suppose that a sequence $\left\{t^k > 0,\ k = 0, 1, 2, ..., K\right\}$ is satisfying the following inequality:*

$$t_{k+1} \leq t_k - \eta t_k^{1-\gamma}, \quad k = 0, 1, 2, ..., K-1,$$

where $\eta > 0$ and $\gamma \in (0, 1]$. Then $K \leq \left\lceil \frac{t_0^\gamma}{\eta\gamma} \right\rceil$.

Using Lemma 4.9 for $t_k = \Psi_{h_k}$, we may obtain the proper values of η and $\gamma \in (0, 1]$.

$$\eta = \frac{1}{16\log a\left(1+\frac{\log\left(1+2\sqrt{\Psi_0}\right)}{\log a}\right)^2}, \quad \gamma = \frac{1}{2}.$$

Theorem 4.10. [9] *Let $\bar{\Psi}_0$ be defined as in (3.16) and let L be the total number of inner iterations in the outer iteration for large-update methods. We have*

$$L \leq 32\log a\left(1+\frac{\log\left(1+2\sqrt{\bar{\Psi}_0}\right)}{\log a}\right)^2 \bar{\Psi}_0^{\frac{1}{2}},$$

The number of outer iterations is bounded above by $\frac{\log\frac{n}{\epsilon}}{\theta}$ (see [17] Lemma II.17, page116). By multiplying the number of outer iterations by the number

of inner iterations, we get an upper bound for the total number of iterations for large -update methods, which is

$$\left\lceil 32 \log a \left(\frac{\log a + \log\left(1 + 2\sqrt{\bar{\Psi}_0}\right)}{\log a} \right)^2 \bar{\Psi}_0^{\frac{1}{2}} \frac{1}{\theta} \log \frac{n}{\epsilon} \right\rceil .$$

For small-update methods with $\tau = O(1)$ and $\theta = \Theta\left(\frac{1}{\sqrt{n}}\right)$, we have:

$$\mathcal{O}\left(\sqrt{n} \log \frac{n}{\varepsilon}\right)$$

iteration complexity, which is independent of the parameter a, and improves upon the large-update bound by a factor of $\log n$.

Remark 4.11. To achieve the best-known iteration bound for large-update methods with $\tau = \mathcal{O}(n)$ and $\theta = \Theta(1)$:
The parameter a must be chosen such that $\log a = \Theta(\log n)$. This ensures that the factor $\left(1 + \frac{\log(1+2\sqrt{\bar{\Psi}_0})}{\log a}\right)^2$ is of order $\mathcal{O}(1)$ and that $\sqrt{\bar{\Psi}_0} = \mathcal{O}(\sqrt{n} \log a)$. A convenient choice satisfying these requirements is

$$a = 1 + 2\left(\frac{n\theta + 2\tau + 2\sqrt{2n\tau}}{2(1-\theta)}\right)^{\frac{1}{2}} \tag{4.13}$$

With this selection, the iteration bound reduces to

$$\mathcal{O}\left(\sqrt{n} \log n \log \frac{n}{\varepsilon}\right),$$

which matches the currently best-known iteration bound for large-update IPMs.

5. Numerical Results

In this section, we present numerical results to confirm the effectiveness of our proposed kernel function $\psi_a(t)$. Due to space limitations, the detailed problem data for Problems 1–5 are provided in the Appendix. The reader is referred to the original sources cited below for further details.

Experimental setup. The experiments were conducted in Dev-Cpp 5.11 TDM-GCC 4.9.2 on a PC with the following parameters:

- Accuracy parameter: $\varepsilon = 10^{-6}$
- Threshold parameter: $\tau = \sqrt{n}$
- Barrier update parameter: $\theta \in \{0.15,\ 0.3,\ 0.6,\ 0.95\}$
- Practical step size: $\alpha_{\text{pra}} = \rho \min(\alpha_x, \alpha_s)$ with $\rho \in (0, 1)$

We denote by I_t and T the number of iterations and the computational time (in seconds), respectively. Our goal is to compare the performance of our kernel function $\psi_a(t)$ with other existing kernel functions listed in Table 1.

TABLE 1. KERNEL FUNCTIONS USED IN NUMERICAL COMPARISON.

The kernel function	Ref
$\psi_a(t) = \frac{t^2-1}{2} - \int_1^t a^{(\frac{1}{x}-1)}dx, \quad a \geq e$	new
$\psi_h(t) = t^2 - 1 + \frac{1}{p}\left(\frac{\cosh^p(t-1)}{\tanh(1)\cosh^p(1)t^p} - \frac{1}{\tanh(1)}\right), \quad p \geq 4$	[11]
$\psi_1(t) = \frac{t^2-1}{2} - \tanh^p(1)\int_1^t \coth^p(x)\exp\left(\frac{\cosh(x)-\cosh(1)}{\cosh^p(1)-1}\right)dx, \quad p \geq 1$	[8]
$\psi_l(t) = \frac{t^2-1}{2} - \log(t)$	[13]
$\psi_2(t) = t^2 - 1 - \log(t) + \frac{t^{-p}-1}{p}, \quad p > 0$	[3]
$\psi_3(t) = (p+1)t^2 - \frac{1}{t^p} - (p+2)t, \quad p > 4$	[12]
$\psi_4(t) = t^2 - 1 - \frac{t^{-2p+1}-1}{-2p+1} - \frac{t^{-p+1}-1}{-p+1}, \quad p > 1$	[4]
$\psi_5(t) = p\frac{t^2-1}{2} + \frac{4}{\pi}\left(e^{p\left(\tan\left(\frac{\pi}{2+2t}\right)\right)^{-1}} - 1\right), \quad p \geq 1$	[16]

Test problems. The following five test problems were considered:

- **Problem 1.** (Example 3, [2]) See Appendix for detailed data.
- **Problem 2.** (Example 3, [7]) See Appendix for detailed data.
- **Problem 3.** (Example 4, [7]) See Appendix for detailed data.
- **Problem 4.** (Example 5, [7]) For $n = 2m$, we have $Q(i,j) = 0$ for all i, j. See Appendix for details.

Discussion of numerical results. From Tables 2 and 3, we observe the following:

- Our kernel function ψ_a consistently achieves the lowest iteration counts across all test problems, particularly for small θ values ($\theta = 0.15, 0.3$). This confirms its fast convergence in practice.
- The kernel function ψ_h shows competitive computational times, sometimes slightly better than ψ_a, but with slightly higher iteration counts.
- The classical logarithmic kernel ψ_l requires the largest number of iterations and computational time, highlighting the effectiveness of the proposed kernel functions.
- As the problem size increases ($m = 100, 200, 500, 1000$), the advantage of ψ_a becomes more pronounced, especially in terms of iteration count.
- For large-update methods ($\theta = 0.6, 0.95$), all kernel functions converge quickly, but ψ_a and ψ_h remain the most efficient.

These numerical results support the theoretical complexity bound $\mathcal{O}(\sqrt{n}\log n \log\frac{n}{\varepsilon})$ derived in Section 4 and demonstrate the practical effectiveness of the proposed kernel function ψ_a for convex quadratic programming.

6. CONCLUDING REMARKS AND FUTURE RESEARCH

In this paper, we extended the primal-dual interior-point algorithm based on eligible kernel functions, originally developed for linear optimization (LO) in [9],

TABLE 2. Numerical results for Problems 1, 2, and 3 using various kernel functions

Problem	Kernel	$\theta = 0.15$		$\theta = 0.3$		$\theta = 0.6$		$\theta = 0.95$	
		I_t	T	I_t	T	I_t	T	I_t	T
Problem 1	ψ_a	**117**	**0.76**	60	**0.6**4	**30**	**0.63**	**17**	0.58
	ψ_h	118	**0.76**	**59**	0.65	**30**	**0.63**	17	0.54
	ψ_1	**117**	**0.71**	64	0.67	32	0.64	**17**	**0.52**
	ψ_l	127	0.94	69	0.91	37	0.83	21	0.75
	ψ_2	119	0.81	61	0.79	31	0.69	19	0.67
	ψ_3	122	0.76	62	0.72	31	0.65	19	0.59
	ψ_4	126	0.88	67	0.81	37	0.81	20	0.80
	ψ_5	122	0.80	64	0.75	35	0.73	19	0.70
Problem 2	ψ_a	**131**	0.86	75	**0.61**	41	0.57	**25**	0.46
	ψ_h	**131**	**0.83**	**74**	**0.61**	42	**0.55**	26	**0.43**
	ψ_1	135	0.92	76	0.71	37	0.65	26	0.54
	ψ_l	143	1.05	88	0.96	53	0.74	37	0.61
	ψ_2	133	0.90	77	0.75	45	0.61	28	0.57
	ψ_3	140	1.04	87	0.89	48	0.70	31	0.59
	ψ_4	132	0.89	78	0.72	43	0.68	28	0.58
	ψ_5	139	0.93	80	0.74	47	0.72	35	0.64
Problem 3	ψ_a	**137**	1.13	**83**	1.05	43	0.88	30	0.85
	ψ_h	138	**1.09**	**83**	0.98	48	**0.81**	29	**0.73**
	ψ_1	137	1.10	86	0.77	43	0.88	27	0.75
	ψ_l	149	1.34	93	1.17	59	1.02	34	0.98
	ψ_2	139	1.26	87	1.09	53	0.99	31	0.94
	ψ_3	143	1.36	89	1.13	58	1.07	32	0.99
	ψ_4	141	1.19	84	1.06	52	0.94	31	0.88
	ψ_5	148	1.24	85	1.12	54	1.05	32	0.94

to convex quadratic programming (CQP). By employing Algorithm 1 with the newly proposed kernel function (1.1), and through a specific selection of its parameter a, we demonstrated that the algorithm achieves the best-known complexity bound for large-update methods, namely $O\left(\sqrt{n}\log n\log\frac{n}{\varepsilon}\right)$. Additionally, we presented numerical experiments on five test problems to illustrate the practical effectiveness of the proposed kernel function. A promising direction for future research lies in extending this complexity analysis -based on kernel functions with novel barrier terms- to primal-dual interior-point methods for convex quadratic semidefinite optimization.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to [any funding sources, institutions, or individuals who provided support or valuable insights during this research]. Their contributions were instrumental in the development and completion of this work. https://www.aimsciences.org/article/doi/10.3934/naco.2022003

TABLE 3. Numerical results for Problem 4 with different values of m using various kernel functions

m	Kernel	$\theta = 0.15$		$\theta = 0.3$		$\theta = 0.6$		$\theta = 0.95$	
		I_t	T	I_t	T	I_t	T	I_t	T
100	ψ_a	**295**	**25.72**	**209**	**19.75**	**150**	13.34	**88**	**8.59**
	ψ_h	297	36.93	228	20.23	165	**12.60**	94	9.35
	ψ_1	306	27.80	237	20.70	163	13.40	92	9.47
	ψ_l	349	41.75	266	22.93	178	14.52	108	9.80
	ψ_2	320	29.53	228	19.84	**150**	14.63	89	10.43
	ψ_3	307	27.64	216	27.77	164	15.36	97	9.51
	ψ_4	384	26.02	295	20.40	172	15.15	107	9.38
	ψ_5	357	30.27	236	21.61	177	14.49	100	10.32
200	ψ_a	**335**	**31.85**	**221**	**21.94**	151	15.92	**92**	**8.95**
	ψ_h	336	37.92	229	24.46	167	**15.62**	99	9.95
	ψ_1	346	33.91	248	24.16	168	15.90	97	10.12
	ψ_l	386	40.51	275	26.45	188	17.28	116	11.45
	ψ_2	358	34.28	239	23.12	162	16.38	98	11.08
	ψ_3	345	32.75	230	30.15	170	17.95	105	10.68
	ψ_4	412	31.22	310	24.88	185	17.45	115	10.95
	ψ_5	389	34.98	254	25.32	189	17.12	112	11.42
500	ψ_a	**375**	**37.95**	233	**23.69**	**152**	**16.55**	**97**	**9.30**
	ψ_h	376	38.75	**230**	28.68	166	17.60	104	10.55
	ψ_1	386	39.82	258	27.61	167	17.80	100	9.85
	ψ_l	422	39.09	284	23.97	197	20.58	124	12.85
	ψ_2	382	40.99	289	31.05	153	18.02	102	10.49
	ψ_3	416	38.10	278	29.84	166	18.37	101	10.91
	ψ_4	394	38.11	235	32.10	154	19.88	108	11.01
	ψ_5	411	39.82	272	28.10	163	20.35	122	11.97
1000	ψ_a	**412**	**45.20**	255	**28.15**	**168**	**19.20**	**108**	**10.85**
	ψ_h	415	46.30	**252**	33.45	182	20.15	115	12.10
	ψ_1	425	47.10	278	32.20	184	20.50	112	11.95
	ψ_l	465	46.80	305	28.75	215	23.40	138	14.20
	ψ_2	430	47.95	312	35.80	172	20.80	115	12.25
	ψ_3	458	44.65	298	34.20	185	21.15	114	12.68
	ψ_4	435	44.85	262	36.90	172	22.60	120	12.85
	ψ_5	452	46.20	298	32.45	182	23.15	134	13.75

REFERENCES

[1] M.Achache: Complexity analysis of an interior point algorithm for the semidefinite optimization based on a new kernel function with a double barrier term. Acta Mathematica Sinica, English Series, 31 (3): 543-556, (2015). https://doi.org/10.1007/s10114-015-1314-4.

[2] M.Achache, N.Tabchouche: A full-Newton step feasible interior-point algorithm for monotone horizontal linear complementarity problems, Optimization Letters. 1-19: (2018). https://doi.org/10.1007/s11590-018-1328-9.

[3] Y.Q.Bai, M.El.Ghami, C. Roos: A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization, SIAM J. Optim, 101-128 (2004). https://doi.org/10.1137/S1052623403423114.

[4] A.Benhadid, F. Merahi: Complexity analysis of an interior-point algorithm for linear optimization based on a new parametric kernel function with a double barrier term. NACO. 13(2), 224-238, (2022). https://www.aimsciences.org/article/doi/10.3934/naco.2022003

[5] M.Bouafia, D.Benterki, A.Yassine: An efficient primal-dual interior point method for linear programming problems based on a new kernel function with a trigonometric barrier term. J. Optim. Theory Appl. 170, 528–545, (2016). https://doi.org/10.1007/s10957-016-0895-0.

[6] M.Bouafia, A.Yassine: Extension of primal-dual interior point method based on a kernel function for linear fractional problem, Numer. Anal. Approx. Theory. 52(2), 162–171, (2023). https://doi.org/10.33993/jnaat522-1349.

[7] N.Boudjellal, H.Roumili, Dj.Benterki: Complexity Analysis of Interior Point Methods for Convex Quadratic Programming Based on a Parameterized Kernel Function, Bol. Soc. Paran. Mat, (2022). https://doi.org/10.5269/bspm.47772.

[8] Y.Bouhenache, W.Chikouche, I.Touil, S.F-Hafshejani: Complexity analysis of primal-dual interior-point methods for convex quadratic programming based on a new twice parameterized kernel function. Jornal of Mathematical Modeling. 12(2), 247-265, (2024). https://doi.org/10.22124/jmm.2024.25394.2257.

[9] B. Bounibane, R. Chalekh: Kernel function-based primal-dual interior point methods for linear optimization. Palestine Journal of Mathematics, 13(4), 261–277, (2024). https://api.semanticscholar.org/CorpusID:279239371

[10] B. Bounibane, R. Chalekh: Complexity analysis of primal-dual interior-point methods for semidefinite optimization based on a new type of kernel functions, Stat., Optim. Inf. Comput. 12, 1745–1761. (2024). https://doi.org/10.19139/soic-2310-5070-1927.

[11] R.Chalekh, EL.A.Djeffal: Complexity analysis of an interior-point algorithm for CQP based on a new parametric kernel function. Statistics, Optimization and Information Computing. 12, 153-166, (2024). https://doi.org/10.19139/soic-2310-5070-1761.

[12] El.A.Djeffal, M.Laouar: A primal-dual interior-point method based on a new kernel function for linear complementarity problem. Asian-European Journal of Mathematics, (2019). https://api.semanticscholar.org/CorpusID:126040737

[13] M.El.Ghami, I.D.Ivanov, C.Roos, T.Steihaug: A polynomial-time algorithm for LO based on generalized logarithmic barrier functions, Int. J. Appl. Math., 21, 99–115, (2008). https://doi.org/10.1016/j.cam.2008.05.027.

[14] M.El.Ghami, Z.A.Guennoun, S.Bouali, T.Steihaug: Interior point methods for linear optimization based on a kernel function with a trigonometric barrier term. J. Comput. Appl. Math. 236, 3613–3623, (2012). https://doi.org/10.1016/j.cam.2011.05.036.

[15] N.Karmarkar: A new polynomial-time algorithm for linear programming. In Proceeding of the 16th Annual ACM. Symposium on theory of computing. 303-311, (1984). https://doi.org/10.1007/BF02579150

[16] Z.Moaberfardi, S.F-Hafshejani, A.Fakharzadeh: An interior-point method for lineair optimization based on a trigonometric kernel function. J. Nonlinear Funct. Anal. (2019). https://doi.org/10.23952/jnfa.2019.46.

[17] J.Peng, C.Roos, T.Terlaky: Self-regularity - a new paradigm for primal-dual interior-point algorithmss. Princeton series in applied mathematics, (2002). https://api.semanticscholar.org/CorpusID:118159820

[18] M.R.Peyghami, S.F.Hafshejani: An interior point algorithm for solving convex quadratic semidefinite optimization problems using a new kernel function. Iranian J Math Sci Inform. 12, 131–152, (2017). https://doi.org/10.7508/IJMSI.2017.01.011.

[19] M.R.Peyghami, S.F.Hafshejani: Complexity analysis of an Interior point algorithm for linear optimisation based on a new proximity function. Numerical-Algorithms. 67, 33-48, (2014). https://doi.org/10.1007/s11075-013-9772-1.

[20] C.Roos, T.Terlaky, J.-Ph.Vial: Theory and Algorithms for Linear Optimization. An Interior-Point Approach. John Wiley, Sons, Chichester, UK, (1997), 2nd Ed, Springer, (2006). https://api.semanticscholar.org/CorpusID:32992663.

[21] M.W.Zhang: A large-update interior-point algorithm for convex quadratic semidefinite optimization based on a new kernel function. Acta Mathematica Sinica, English Series. 28, 2313-2321, (2012). https://api.semanticscholar.org/CorpusID:122495002.

[22] Y. Bouhenache, W. Chikouche, I. Toul, and S. F. Hafshenjani: Complexity analysis of primal-dual interior-point methods for convex quadratic programming based on a new twice parameterized kernel function. Journal of Mathematical Modeling. 12 , 247-265, (2024). https://doi.org/10.22124/jmm.2024.25394.2257.

[23] A. Guemmaz, B. Bounibane, and El. A. Djeffal: Primal-Dual interior point method for LCP based on a new kernel function with a logarithmic barrier term. Nonlinear Studies. 33(1) , 377-388, (2026). https://www.nonlinearstudies.com/index.php/nonlinear/index

[24] Z. Dib, A. Leulmi: A new numerical scheme for solving linear optimization problems. Gulf Journal of Mathematics. 20(1), 303-312, (2025). https://doi.org/10.56947/gjom.v20i.2916

[25] N. Kacem, A. Leulmi: An explicit step length for solving an optimization problem. Gulf Journal of Mathematics. 19(2), 398-411, (2025). https://doi.org/10.56947/gjom.v19i2.2801

[26] F. Leulmi, S. Leulmi, and A. M. Bekkouche: Analysis and numerical estimation of the step size in optimization problems. Gulf Journal of Mathematics. 21(2), 275-287, (2025). https://doi.org/10.56947/gjom.v21i2.3623.

[1] Laboratory of Science for Mathematics, Computer Science and Engineering Applications, Department of Mathematics, University Center of Barika, Amdoukal Road, Barika, 05001, Algeria.

Email address: abderrahim.guemmaz@cu-barika.dz.

[2] Department of Mathematics, University of Batna 2, Algeria.

Email address: b.bounibane@univ-batna2.dz