Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 2, pp. 437–453
DOI : 10.47974/JIOS-1464

Interior-point method for CQP problems employing a novel trigonometric kernel function

I. Touil *
W. Chikouche †
Department of Mathematics
Laboratory of Pure and Applied Mathematics
Mohammed Seddik Ben Yahia University
Jijel
Algeria

Dj. Benterki §
Department of Mathematics
Laboratory of Fundamental and Numerical Mathematics
Ferhat Abbas University
Setif
Algeria

Abstract

We introduce a primal-dual interior-point approach to address convex quadratic programming dilemmas, utilizing a novel class of efficient parametric kernel functions. Using basic analytical tools, we establish that the developed algorithm has an iteration bound of $\mathcal{O}\left((p+1)n^{\frac{p+2}{2(p+1)}}\log\frac{n}{\varepsilon}\right)$ for the large-update method, where p is a parameter with $p \geq 2$. For the small-update method, we obtain the best known iteration bound, namely $\mathcal{O}\left(p^2\sqrt{n}\log\frac{n}{\varepsilon}\right)$ [5, 7]. Finally, we provide numerical experiments that illustrate the algorithm's effectiveness.

Subject Classification (2010): *90C20, 90C25, 90C51.*

Keywords: *Interior-point methods, Convex quadratic programming, Kernel functions, Iterations bound.*

* *E-mail:* **touilimene@univ-jijel.dz (Corresponding Author)**
† *E-mail:* **w_chikouche@yahoo.com**
§ *E-mail:* **djbenterki@univ-setif.dz**

1. Introduction

The given primal convex quadratic programming (CQP) problem in its standard form is as follows:

$$(PQ)\begin{cases}\min(c^T\overline{x}+\frac{1}{2}\overline{x}^TQ\overline{x})\\ A\overline{x}=b,\\ \overline{x}\geq 0,\end{cases}$$

The dual problem of (PQ) is written as follows:

$$(DQ)\begin{cases}\max(b^T\overline{y}-\frac{1}{2}\overline{x}^TQ\overline{x})\\ A^T\overline{y}+\overline{z}-Q\overline{x}=c,\\ \overline{z}\geq 0,\end{cases}$$

where the vectors $c,\overline{x}$ and $\overline{z}$ belong to $\mathbb{R}^n$, b and $\overline{y}$ are in $\mathbb{R}^m$, the matrix A is in $\mathbb{R}^{m\times n}$, and $Q\in\mathbb{S}_+^n$ such that $\mathbb{S}_+^n$ denotes the cone of symmetric positive semidefinite matrices.

Let's suppose, without compromising generality, that:

(H_1): A is a matrix that has full row rank, meaning that its rank, denoted as $\operatorname{rank}(A)$, is m such that $m<n$.

(H_2): Both problems (PQ) and (DQ) are strictly feasible, implying the existence of a solution (x,y,z) where

$$\begin{cases}Ax=b, & x>0,\\ A^Ty+z-Qx=c, & z>0.\end{cases}$$

We know that to find the optimal solution for both (PQ) and (DQ) it is sufficient to solve the following optimality conditions of the first order, commonly referred to as the KKT conditions.

$$\begin{cases}A\overline{x}=b, & \overline{x}\geq 0,\\ A^T\overline{y}+\overline{z}-Q\overline{x}=c, & \overline{z}\geq 0,\\ \overline{x}\,\overline{z}=0.\end{cases}\tag{1.1}$$

Interior-point methods (IPMs) offer a robust methodology for addressing problems in linear programming (LP). These methods have been effectively applied to a broader range of optimization problems, including Nonlinear Complementarity Programming (NCP) Problems [12], Convex Quadratic Programming (CQP) Problems [2, 6, 9], Semidefinite Programming (SDP) Problems [14, 15], Second-Order Cone Programming

(SOCP) Problems [3], and $P_*(k)-$Horizontal Linear Complementarity Programming ($P_*(k)$-HLCP) Problems [8].

Most IPMs rely on the logarithmic kernel function (KF) [13]. Various kernel functions (KFs) incorporating different types of barrier terms have been proposed and examined.

The concept of KFs is commonly used in various fields such as: machine learning [10, 11] and bioinformatics [1],... .

The fundamental notion of the IPMs involves introducing a positive parameter $\overline{\mu}$ into the condition known as the complementary condition, expressed as $\overline{x}\,\overline{z}=0.$ This leads to a nonlinear system expressed as follows

$$\begin{cases} A\overline{x}=b, & \overline{x}>0, \\ A^T\overline{y}+\overline{z}-Q\overline{x}=c, & \overline{z}>0, \\ \overline{x}\,\overline{z}=\overline{\mu}e, & \overline{\mu}>0. \end{cases} \tag{1.2}$$

where the vector $e\in\mathbb{R}^n$ such that all elements are equal to one. It's well-known that under assumptions (H_1) and (H_2), for each $\overline{\mu}>0$ the system (1.2) admits a unique solution denoted $(\overline{x}_{\overline{\mu}},\overline{y}_{\overline{\mu}},\overline{z}_{\overline{\mu}})$. The set of these solutions, denoted by $\mathcal{S}=\{(\overline{x}_{\overline{\mu}},\ \overline{y}_{\overline{\mu}},\ \overline{z}_{\overline{\mu}});\ \overline{\mu}>0\}$ is termed the central trajectory (also known as the $\overline{\mu}$-center) of problems (PQ) and (DQ). If $\overline{\mu}$ goes to 0, the central trajectory converges, providing optimal solutions for both problems (PQ) and (DQ).

Newton's method allows for obtaining the directions $\Delta\overline{x}$, $\Delta\overline{y}$, and $\Delta\overline{z}$ by solving:

$$\begin{cases} A\Delta\overline{x}=0, \\ A^T\Delta\overline{y}+\Delta\overline{z}-Q\Delta\overline{x}=0, \\ \overline{z}\Delta\overline{x}+\overline{x}\Delta\overline{z}=\overline{\mu}e-\overline{x}\,\overline{z}, \end{cases} \tag{1.3}$$

which contributes to the development of the traditional primal-dual path-following algorithm.

2. Refined search directions using kernel functions

In the present section, we define the following novel KF

$$\varphi(t)=\frac{2\pi(t^2-1)}{3^{\frac{p+5}{2}}}+\frac{1}{p}\left(\cot^p(l(t))-\frac{1}{3^{\frac{p}{2}}}\right),\text{for } t>0 \text{ and } p\geq 2, \tag{2.1}$$

where

$$l(t)=\frac{\pi t}{2t+1}. \tag{2.2}$$

Then, the three successive derivatives of φ are:

$$\varphi^{(1)}(t)=\frac{4\pi t}{3^{\frac{p+5}{2}}}-\frac{l^{(1)}(t)}{\sin^2 l(t)}\cot^{p-1}(l(t)), \tag{2.3}$$

$$\varphi^{(2)}(t)=\frac{4\pi}{3^{\frac{p+5}{2}}}+\frac{1}{\sin^2 l(t)}\Omega_1(t)>\frac{4\pi}{3^{\frac{p+5}{2}}}, \tag{2.4}$$

$$\varphi^{(3)}(t)=-\frac{1}{\sin^2 l(t)}(\Omega_2(s)(l^{(1)}(t))^3+\Omega_3(t)+(\cot^{p-1}(l(t)))l^{(3)}(t))<0, \tag{2.5}$$

where

$$\Omega_1(t)=(l^{(1)}(t))m(t)^2-l^{(2)}(t)[\cot^{p-1}(l(t))],$$

$$\Omega_2(t)=\cot^{p-3}(l(t))[(p-2)(p-1)+2p^2\cot^2(l(t))+(p+2)(p+1)\cot^4(l(t))],$$

$$\Omega_3(t)=-3m(t)l^{(2)}l^{(1)}(t)$$

and

$$m(t)=\cot^{p-2}(l(t))[(p+1)\cot^2(l(t))+(p-1)],$$

$$l^{(1)}(t)=\frac{\pi}{(2t+1)^2},\ l^{(2)}(t)=\frac{-4\pi}{(2t+1)^3}\text{ and }l^{(3)}(t)=\frac{24\pi}{(2t+1)^4}. \tag{2.6}$$

We define the vector $\overline{v}$ and the scaled search directions $d_{\overline{x}}$ and $d_{\overline{z}}$ as:

$$\overline{v}=\sqrt{\frac{\overline{x}\,\overline{z}}{\overline{\mu}}},\ d_{\overline{x}}=\overline{X}^{-1}\overline{V}\Delta\overline{x},\text{ and }d_{\overline{z}}=\overline{Z}^{-1}\overline{V}\Delta\overline{z}. \tag{2.7}$$

Then, system (1.3) is reduced to

$$\begin{cases}\overline{A}d_{\overline{x}}=0,\\ \overline{A}^T\Delta\overline{y}+d_{\overline{z}}-\overline{Q}d_{\overline{x}}=0,\\ d_{\overline{x}}+d_{\overline{z}}=\overline{v}^{-1}-\overline{v}=-\nabla\Phi_c(\overline{v}),\end{cases} \tag{2.8}$$

with $\overline{A}=\frac{1}{\overline{\mu}}A\overline{V}^{-1}\overline{X}$, $\overline{Q}=\frac{1}{\overline{\mu}}\overline{V}^{-1}\overline{X}Q\overline{V}^{-1}\overline{X}$ and $B=\text{diag}(b)$, where $B\in\{\overline{V},\overline{X},\overline{Z}\}$ and $b\in\{\overline{v},\overline{x},\overline{z}\}$.

Here $\Phi_c(\overline{v})=\sum_{i=1}^n\varphi_c(\overline{v}_i)$, represents the logarithmic barrier (proximity) function within the classical KF

$$\varphi_c(t)=\frac{1}{2}(t^2-1)-\log(t).$$

The main idea of IPMs for approximating the central trajectory involves substituting $\varphi_c(t)$ by any function $\varphi(t)$ strictly convex defined from $(0,+\infty)$ to $(0,+\infty)$ that achieves its minimum at $t=1$, where $\varphi(1)=0$.

In this work, $\varphi_c(t)$ is replaced by the proposed KF $\varphi(t)$ defined in (2.1). Thus

$$\begin{cases} \overline{A}d_{\overline{x}}=0, \\ \overline{A}^T\Delta\overline{y}+d_{\overline{z}}-\overline{Q}d_{\overline{x}}=0, \\ d_{\overline{x}}+d_{\overline{z}}=-\nabla\Phi(\overline{v}). \end{cases} \tag{2.9}$$

Thanks to the assumption (H_1), system (2.9) possesses a unique solution, denoted by $(d_{\overline{x}},\Delta\overline{y},d_{\overline{z}})$. Since $d_{\overline{x}}$ and $d_{\overline{z}}$ being available, we can easily calculate the values of $\Delta\overline{x}$ and $\Delta\overline{z}$. By choosing a suitable displacement step $\overline{\alpha}$ through the Newton direction, the updated iterate $(\overline{x}^+,\overline{y}^+,\overline{z}^+)$ is formulated as

$$(\overline{x}^+,\overline{y}^+,\overline{z}^+):=(\overline{x},\overline{y},\overline{z})+\overline{\alpha}(\Delta\overline{x},\Delta\overline{y},\Delta\overline{z}), \tag{2.10}$$

where $\overline{\alpha}\in(0,1)$ ensures that $\overline{x}^+$ and $\overline{z}^+$ are strictly greater than zero.

We can summarize the aforementioned generic method in the following interior-point algorithm (IPA):

IPA for CQP

Initial data

$\overline{\tau}=\sqrt{n}\geq 1$ is a threshold value;

$\overline{\varepsilon}=10^{-8}>0$ is a precision value;

$0<\overline{\theta}<1$ is a fixed adjustment to the barrier;

(x,y,z) is a strictly feasible point, $\mu=1$ such that $\Psi(x,z;\mu)=\Phi(v)\leq\overline{\tau}$;

Begin

$\overline{x}:=x;\overline{y}:=y;\overline{z}:=z;\overline{\mu}:=\mu;$

Repeat until $\overline{x}^T\overline{z}=n\overline{\mu}\geq\overline{\varepsilon}$

  $\overline{\mu}:=(1-\overline{\theta})\overline{\mu};$

  Repeat until $\Psi(\overline{x},\overline{z};\overline{\mu})=\Phi(\overline{v})>\overline{\tau}$

    Solve (2.9) and select an appropriate displacement step $\overline{\alpha}$;

    Update $(\overline{x},\overline{y},\overline{z}):=(\overline{x},\overline{y},\overline{z})+\overline{\alpha}(\Delta\overline{x},\Delta\overline{y},\Delta\overline{z});\overline{v}:=\sqrt{\frac{\overline{x}\,\overline{z}}{\overline{\mu}}};$

  End Repeat

End Repeat

End

2.1 *Characteristics of the new kernel function*

Several properties of the KF defined in (2.1) are developed in the following lemmas:

Lemma 2.1: *Suppose* $l(t)$ *represents the function defined in* (2.2). *Hence,* $\forall t \in (0,+\infty)$ *one has*

$$l(t) \in \left(0, \frac{\pi}{2}\right), \tag{2.11}$$

$$\cot(l(t)) \in (0,+\infty) \tag{2.12}$$

and

$$\cot(l(t)) - \frac{t^{-1}}{(p+1)\pi} \in (0,+\infty). \tag{2.13}$$

Proof: For (2.11), $l(t)$ strictly increases over $(0,+\infty)$ since $l^{(1)}(t) > 0$ for all $t \in (0,+\infty)$ and by taking the limit when t goes to zero and to $+\infty$, we obtain

$$\lim_{t\to 0^+} l(t) = 0 \text{ and } \lim_{t\to+\infty} l(t) = \frac{\pi}{2}.$$

which proves (2.11).

The inequality (2.12) is then a direct consequence.

For (2.13), let $k(t) = \cot(l(t)) - \frac{t^{-1}}{(p+1)\pi}$, since $\sin(l(t)) \le l(t)$, then

$$k^{(1)}(t) = -\frac{l^{(1)}(t)}{\sin^2(l(t))} + \frac{t^{-2}}{(1+p)\pi} \le \frac{-p\pi^2 t^2}{(1+p)\pi t^2 (2t+1)^2 \sin^2(l(t))} < 0, \forall t \in (0,+\infty).$$

Hence, $k(t)$ decreases over $(0,+\infty)$. Furthermore, $\lim_{t\to+\infty}\left(\cot(l(t)) - \frac{t^{-1}}{(1+p)\pi}\right) = 0$, this gives (2.13).

Lemma 2.2: *Given the KF* $\varphi(t)$ *defined in* (2.1), *it follows that for all* $t > 0$, *one has:*

(i) $t\varphi^{(2)}(t) + \varphi^{(1)}(t) > 0.$

(ii) $t\varphi^{(2)}(t) - \varphi^{(1)}(t) > 0.$

(iii) The function $\varphi^{(2)}(t)$ decreases monotonically.

(iv) $\varphi^{(1)}(\lambda t)\varphi^{(2)}(t) - \lambda\varphi^{(2)}(\lambda t)\varphi^{(1)}(t) > 0$, for all $t > 1$ and $\lambda > 1$.

Proof: For (i) and (ii), using (2.3), (2.4), (2.6) and (2.13) of Lemma 2.1, we get

$$t\varphi^{(2)}(t)+\varphi^{(1)}(t)=\frac{8\pi t}{3^{\frac{p+5}{2}}}+\frac{(1+p)\pi^2 t\cot^{p-1}(l(t))}{(2t+1)^4\sin^2(l(t))}\left(\cot(l(t))+\frac{4t^2-1}{\pi(1+p)t}\right)$$
$$+\left(\frac{l^{(1)}(t)}{\sin(l(t))}\right)^2(p-1)t\cot^{p-2}(l(t))\geq 0.$$

$$t\varphi^{(2)}(t)-\varphi^{(1)}(t)=\frac{t}{\sin^2 l(t)}[m(t)(l^{(1)}(t))^2+\cot^{p-1}(l(t))(l^{(1)}(t)l(t)-tl^{(2)}(t))]\geq 0.$$

(iii) is a direct consequence from (2.5).

For (iv), applying Lemma 2.4 from , and given that $\varphi(t)$ satisfies conditions (ii) and (iii) above, (iv) follows. Thus, the proof is now concluded.

Lemma 2.3: *Regarding* $\varphi(t)$, *we observe:*

(i) $\frac{(\sqrt{2\pi}(t-1))^2}{3^{\frac{p+5}{2}}}\leq\varphi(t)\leq\frac{3^{\frac{p+5}{2}}}{8\pi}(\varphi^{(1)}(t))^2,\ \forall t\in]0,+\infty[.$

(ii) $\varphi(t)\leq\frac{(\sqrt{2\pi}(t-1))^2}{3^{\frac{p+11}{2}}}(63+(4p-2)\pi\sqrt{3}),\ \forall t\in[1,+\infty[.$

Proof: For (i), since $\varphi^{(2)}(t)>\frac{4\pi}{3^{\frac{p+5}{2}}}$ from (2.4), we have

$$\varphi(t)=\int_1^t\int_1^\varsigma\varphi^{(2)}(\zeta)d\zeta d\varsigma\geq\frac{4\pi}{3^{\frac{p+5}{2}}}\int_1^t\int_1^\varsigma d\zeta d\varsigma=\frac{(\sqrt{2\pi}(t-1))^2}{3^{\frac{p+5}{2}}},$$

and

$$\varphi(t)=\int_1^t\int_1^\varsigma\varphi^{(2)}(\zeta)d\zeta d\varsigma\leq\frac{3^{\frac{p+5}{2}}}{4\pi}\int_1^t\int_1^\varsigma\varphi^{(2)}(\varsigma)\varphi^{(2)}(\zeta)d\zeta d\varsigma=\frac{3^{\frac{p+5}{2}}}{8\pi}(\varphi^{(1)}(t))^2.$$

For (ii), considering $\varphi(t)$ and $\varphi^{(1)}(t)$ equal to zero at $t=1$, $\varphi^{(2)}(1)=\frac{4\pi}{3^{\frac{p+11}{2}}}(63+(4p-2)\,\pi\sqrt{3})$, $\varphi^{(3)}(t)<0$, and utilizing Taylor's development, we obtain:

$$\varphi(t)\leq\frac{(\sqrt{2\pi}(t-1))^2}{3^{\frac{p+11}{2}}}(63+(4p-2)\pi\sqrt{3}).$$

Lemma 2.4: *Let* $\overline{\rho}$ *from* $[0,+\infty)$ *to* $[1,+\infty)$ *and* $\overline{\rho}$ *from* $(0,+\infty)$ *to* $(0,1]$ *denote the inverse functions of the functions* $\varphi(t)$ *and* $-\frac{1}{2}\varphi^{(1)}(t)$, *respectively. Then, one has*

(i) $\quad 1+3^{\frac{p+11}{4}}\sqrt{\frac{s}{2\pi(63+(4p-2)\pi\sqrt{3})}} \leq \overline{\rho}(s) \leq 1+3^{\frac{p+5}{4}}\sqrt{\frac{s}{2\pi}}, \ \forall s \in [0,+\infty[.$

(ii) $\quad \cot(l(t)) \leq \left(\frac{18s}{\pi}+\frac{4}{3^{\frac{p+1}{2}}}\right)^{\frac{1}{p+1}}, \ s=-\frac{1}{2}\varphi^{(1)}(t) \in [0,+\infty[.$

Proof: For item (i), set $s=\varphi(t)\geq 0$ where t in $[1,+\infty[$, then $\overline{\rho}(s)=t$. Using item (i) and then item (ii) of Lemma 2.3, we obtain

$s=\varphi(t) \geq \left(\frac{\sqrt{2\pi}(t-1)}{3^{\frac{p+5}{4}}}\right)^2$, this implies $t \leq 1+3^{\frac{p+5}{4}}\sqrt{\frac{s}{2\pi}}$,

$s \leq \frac{(\sqrt{2\pi}(t-1))^2}{3^{\frac{p+11}{2}}}(63+(4p-2)\pi\sqrt{3})$, this implies $t \geq 1+3^{\frac{p+11}{4}}\sqrt{\frac{s}{2\pi(63+(4p-2)\pi\sqrt{3})}}$.

For (ii), let $s=-\frac{1}{2}\varphi^{(1)}(t)$, then $\overline{\rho}(s)=t$, for t in $]0,1]$, so, by (2.3), we obtain

$$2s=-\varphi^{(1)}(t)=\frac{-4\pi t}{3^{\frac{p+5}{2}}}+\frac{l^{(1)}(t)}{\sin^2 l(t)}\cot^{p-1}(l(t)),$$

which is equivalent to

$$\cot^{p-1}(l(t))(1+\cot^2(l(t)))=\frac{(2t+1)^2}{\pi}\left(2s+\frac{4\pi t}{3^{\frac{p+5}{2}}}\right),$$

then, we have

$$\cot^{p+1}(l(t)) \leq \frac{18s}{\pi}+\frac{4}{3^{\frac{p+1}{2}}}.$$

Now, we calculate an approximation of how adjusting the barrier parameter $\overline{\mu}$ impacts the proximity function's value throughout an iteration. This analysis begins with a fundamental theorem applicable to all KFs $\varphi(t)$ meeting the criteria of strict convexity and fulfilling conditions (ii) and (iii) outlined in Lemma 2.2.

Lemma 2.5: ([4]). *For* $\overline{v}$ *in* $\mathbb{R}^n_{++}$ *and* λ *in* $(1,+\infty)$, *one has*

$$\Phi(\lambda\overline{v}) \leq n\varphi\left(\lambda\overline{\rho}\left(\frac{\Phi(\overline{v})}{n}\right)\right).$$

Corollary 2.1: *Suppose* $\overline{\theta}$ *is such that* $\overline{\theta} \in (0,1)$. *Then*

$$\Phi(\overline{v}^+) = \Phi(\lambda\overline{v}) \le \frac{n\overline{\theta}+3^{\frac{p+5}{2}}n\overline{\tau}+2(3^{\frac{p+5}{4}})n\sqrt{2\pi\overline{\tau}}}{1-\overline{\theta}},\ \lambda = \sqrt{\frac{1}{1-\overline{\theta}}} > 1,$$

if

$$\Phi(\overline{v}) \le \overline{\tau}.$$

Proof: Using (2.1), Lemma 2.5, and (i) of Lemma 2.4, we have

$$\Phi(\lambda\overline{v}) \le \frac{2n\pi}{3^{\frac{p+5}{2}}}\left(\lambda^2\left(1+3^{\frac{p+5}{4}}\sqrt{\frac{\Phi(\overline{v})}{2\pi}}\right)^2 - 1\right).$$

Since $\lambda = \frac{1}{\sqrt{1-\overline{\theta}}}$, and based on the assumption that $\Phi(\overline{v}) \le \overline{\tau}$ just before the update of $\overline{\mu}$, we obtain

$$\Phi(\overline{v}^+) \le \frac{n\overline{\theta}+3^{\frac{p+5}{2}}n\overline{\tau}+2(3^{\frac{p+5}{4}})n\sqrt{2\pi\overline{\tau}}}{1-\overline{\theta}} := \Phi_0, \tag{2.14}$$

where Φ_0 serves as an upper limit for $\Phi(\overline{v}^+)$ throughout the algorithm's execution.

The norm-based proximity measure, denoted $\delta(\overline{v})$, is defined as:

$$\delta(\overline{v}) = \frac{1}{2}\|\nabla\Phi(\overline{v})\| = \frac{1}{2}\|d_{\overline{x}} + d_{\overline{z}}\|, \text{ for any } \overline{v} > 0. \tag{2.15}$$

The Lemma below provides a lower bound for $\delta(\overline{v})$ in relation to the proximity function $\Phi(\overline{v})$.

Lemma 2.6: *For any positive vector $\overline{v}$, one has*

$$\delta(\overline{v}) \ge \sqrt{\frac{2\pi}{3^{\frac{p+5}{2}}}\Phi(\overline{v})}.$$

Proof: Utilizing (2.15) and from (i) of Lemma 2.3, one has

$$\delta^2(\overline{v}) = \left(\frac{1}{2}\|\nabla\Phi(\overline{v})\|\right)^2 = \frac{1}{4}\sum_{i=1}^{n}(\varphi^{(1)}(\overline{v}_i))^2 \ge \frac{2\pi}{3^{\frac{p+5}{2}}}\sum_{i=1}^{n}\varphi(\overline{v}_i) = \frac{2\pi}{3^{\frac{p+5}{2}}}\Phi(\overline{v}),$$

which concludes the proof.

3. Examination of the algorithm's complexity

Here, our objective is to calculate a default displacement step $\overline{\alpha}$ that keeps the updated iterate $(\overline{x}^+, \overline{y}^+, \overline{s}^+)$ feasible and ensures sufficient decrease of the barrier function Φ.

While performing an inner iteration, it's important to note that the parameter $\overline{\mu}$ remains fixed. Therefore, using (2.7) in (2.10), we obtain

$$\overline{x}^{+} = \frac{\overline{x}}{\overline{v}}(\overline{v} + \overline{\alpha} d_{\overline{x}}), \quad \overline{z}^{+} = \frac{\overline{z}}{\overline{v}}(\overline{v} + \overline{\alpha} d_{\overline{z}}).$$

Therefore, the new vector $\overline{v}^{+}$ is determined by

$$\overline{v}^{+} = \sqrt{\frac{\overline{x}^{+}\overline{z}^{+}}{\overline{\mu}}} = \sqrt{(\overline{v} + \overline{\alpha} d_{\overline{x}})(\overline{v} + \overline{\alpha} d_{\overline{z}})}.$$

Due to the property of exponential convexity of $\varphi(t)$, we deduce

$$\Phi(\overline{v}^{+}) \leq \frac{\Phi(\overline{v}+\overline{\alpha} d_{\overline{x}}) + \Phi(\overline{v}+\overline{\alpha} d_{\overline{z}})}{2}.$$

Let us define $g(\overline{\alpha}) = \Phi(\overline{v}^{+}) - \Phi(\overline{v})$.

For the sake of simplicity, in the rest of this section, we denote $\delta := \delta(\overline{v})$.

Lemma 3.1: ([15, Lemmas 4.4 and 4.7]) *The maximum of the displacement step* $\overline{\alpha}^{*}$ *is*

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

Furthermore

$$\overline{\alpha}^{*} \geq \frac{1}{\varphi^{(2)}(\overline{\rho}(2\delta))},$$

and

$$g(\overline{\alpha}) \leq -\overline{\alpha}\delta^{2}, \forall \overline{\alpha} \leq \overline{\alpha}^{*}.$$

The following corollary demonstrates that the proximity function decreases during the inner iteration.

Corollary 3.1: *Consider setting* $\hat{\alpha} = \frac{1}{\varphi^{(2)}(\overline{\rho}(2\delta))}$, *as the default displacement step. Assuming* $\delta \geq 1$, *we obtain:*

$$g(\hat{\alpha}) \leq -\frac{\delta^{2}}{\varphi^{(2)}(\overline{\rho}(2\delta))} \leq -\frac{\Phi(\overline{v})^{\frac{p}{2(p+1)}}}{61(6+p\pi^{2})}. \tag{3.1}$$

Proof: Firstly, we search an upper bound for $\varphi^{(2)}(t)$ in relation to δ. For $t \in]0,1]$, with $t = \overline{\rho}(2\delta)$, one has:

$$0 < l(t) \leq \frac{\pi}{3}, \cot(l(t)) \geq \frac{1}{\sqrt{3}}, l^{(1)}(t) \leq \pi \text{and} -l^{(2)}(t) < 4\pi.$$

Using the above inequalities in (2.4), and (ii) of Lemma 2.4, we get

$$\varphi^{(2)}(\bar{\rho}(2\delta)) \leq \left[\frac{4\pi}{3\sqrt{3}}\left(1+\frac{3\sqrt{3}}{8\pi}(p\pi^2+2)\right)\right]\left(\frac{36\delta}{\pi}+\frac{4}{3^{\frac{p+1}{2}}}\right)^{1+\frac{1}{p+1}},$$

Hence, according to Lemma 3.1, by applying the preceding inequality for $\hat{\alpha}=\frac{1}{\varphi^{(2)}(\bar{\rho}(2\delta))}\in[0,\tilde{\alpha}],$ we obtain

$$g(\hat{\alpha}) \leq -\delta^2\hat{\alpha} \leq -\frac{\delta^{\frac{p}{p+1}}}{\left[\frac{4\pi}{3\sqrt{3}}\left(1+\frac{3\sqrt{3}}{8\pi}(p\pi^2+2)\right)\right]\left(454\times\frac{4}{\pi^2}\right)}.$$

Using Lemma 2.6, we have

$$g(\hat{\alpha}) \leq -\frac{\left(\sqrt{\frac{2\pi\Phi(\bar{v})}{3^{\frac{p+5}{2}}}}\right)^{1-\frac{1}{p+1}}}{\left[\frac{4\pi}{3\sqrt{3}}\left(1+\frac{3\sqrt{3}}{8\pi}(p\pi^2+2)\right)\right]\left(454\times\frac{4}{\pi^2}\right)} \leq -\frac{\Phi(\bar{v})^{\frac{1}{2}-\frac{1}{2(p+1)}}}{61(6+p\pi^2)}.$$

This completes the proof.

We aim to calculate the required inner iterations number to restore the condition $\Phi(\bar{v})\leq\bar{\tau}$ after updating $\bar{\mu}$. Let Φ_0 denotes the value of $\Phi(\bar{v})$ after updating $\bar{\mu}$ and Φ_i, where $i=1,\ldots,N,$ with N representing the total number of inner iterations in the outer iteration, denotes the subsequent values in the same outer iteration. From definition of the function $g(\bar{\alpha})$ and in accordance with equation (3.1), we obtain

$$\Phi_{i+1} \leq \Phi_i - \frac{\Phi_i^{\frac{p}{2(p+1)}}}{61(6+p\pi^2)}, \text{ for } i=1,\ldots,N-1.$$

By setting $\beta=\frac{1}{61(6+p\pi^2)}$ and $\gamma=\frac{2+p}{2(p+1)},$ the following lemma is derived.

Lemma 3.2: *Let N be defined above. Then*

$$N \leq \frac{\Phi_0^{\gamma}}{\beta\gamma} = \left(\frac{122(p\pi^2+6)(1+p)}{2+p}\right)\Phi_0^{\frac{2+p}{2(1+p)}} =: \bar{N}. \tag{3.2}$$

Next, we establish the iterations bound for both small and large update IPMs.

Theorem 3.1: *The developed algorithm requires at most*

$$\left(\left(\frac{122(6+p\pi^2)(p+1)}{p+2}\right)\Phi_0^{\frac{p+2}{2(p+1)}}\right)\left(\frac{\log\frac{n}{\varepsilon}}{\bar{\theta}}\right), \tag{3.3}$$

iterations to obtain an $\varepsilon-$*solution.*

Proof: One knows that the outer iterations number such that $n\overline{\mu} = \overline{x}^T\overline{z} \le \overline{\varepsilon}$ is limited by $\frac{1}{\theta}(\log\frac{n}{\overline{\varepsilon}})$. Hence, the lemma is proved by multiplying this number by the number $\overline{N}$ obtained in (3.2).

For the large update method, we have $\overline{\tau} = \mathcal{O}(n)$ and $\overline{\theta} = \Theta(1)$. The iteration bound of the developed algorithm using the proximity function derived from the new KF is $\mathcal{O}\left((1+p)n^{\frac{2+p}{2(1+p)}}\log\frac{n}{\varepsilon}\right)$.

The expression $(p+1)n^{\frac{p+2}{2(p+1)}}$ reaches its minimum at $p = \frac{1}{2}(\log n - 2)$ and is then equal to $\frac{e\sqrt{n}\log n}{2}$. Therefore, we derive the iteration bound as:

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

The above iteration bound coincides with the bound of the large update IPMs in [5], which is based on trigonometric KFs as well for linear programming.

If $\overline{\tau} = \mathcal{O}(1)$ and $\overline{\theta} = \Theta\left(\frac{1}{\sqrt{n}}\right)$, then the iteration bound of the developed IPA for the small update IPM is:

$$\mathcal{O}\left(p^2\sqrt{n}\log\frac{n}{\varepsilon}\right).$$

This matches the currently best-known complexity bound for small-update IPMs.

4. Numerical experiments

In the present part, we perform numerical experiments to show the effectiveness of the developed algorithm. The following examples are taken from the literature, for instance, see [6], and were implemented using MATLAB R2008b on an Intel Core i3 (1.80 GHz) processor with 4.00 GB of RAM. We set $\overline{\tau} = \sqrt{n}$ and $\overline{\theta}$ takes values from the set $\{0.01, 0.1, 0.5, 0.9\}$ for the large update method, while for the small update method, we set $\overline{\tau} = 1$ and $\overline{\theta} = \frac{1}{2\sqrt{n}}$. For all experiments, we have considered $\overline{\varepsilon} = 10^{-8}$ and $p \in \{2, \frac{1}{2}(\log n - 2)\}$, where n represents the size of the example.

Example 4.1: For $m = 2$ and $n = 3$, we take:

$$A = \begin{pmatrix} -1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix},\ b = \begin{pmatrix} 1 \\ 2 \end{pmatrix},\ c = \begin{pmatrix} -2 \\ -4 \\ 0 \end{pmatrix}, Q = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

we have the following initialization:

$$x = \begin{pmatrix} 0.3262 \\ 1.3261 \\ 0.3477 \end{pmatrix}, \quad y = \begin{pmatrix} 0 \\ -2.0721 \end{pmatrix} \text{ and } z = \begin{pmatrix} 0.7247 \\ 0.7247 \\ 2.0722 \end{pmatrix}.$$

The optimal solution obtained is:

$$\overline{x}^* = \begin{pmatrix} 0.5 \\ 1.4999 \\ 0.0000 \end{pmatrix}, \overline{y}^* = \begin{pmatrix} 0 \\ -0.9997 \end{pmatrix} \text{ and } \overline{z}^* = \begin{pmatrix} 0.0000 \\ 0.0000 \\ 0.9998 \end{pmatrix}.$$

Example 4.2: For $m = 2$ and $n = 2m$, we take:

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{pmatrix}, \ b = \begin{pmatrix} 2 \\ 5 \end{pmatrix}, \ c = \begin{pmatrix} -4 \\ -6 \\ 0 \\ 0 \end{pmatrix}, Q = \begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

we have the following initialization:

$$x = \begin{pmatrix} 0.9683 \\ 0.5775 \\ 0.4543 \\ 1.1444 \end{pmatrix}, \quad y = \begin{pmatrix} -0.9184 \\ -1.1244 \end{pmatrix} \text{ and } z = \begin{pmatrix} 0.7612 \\ 0.9141 \\ 0.9185 \\ 1.1244 \end{pmatrix}.$$

The optimal solution obtained is:

$$\overline{x}^* = \begin{pmatrix} 1.1290 \\ 0.7742 \\ 0.0968 \\ 0.0000 \end{pmatrix}, \overline{y}^* = \begin{pmatrix} -0.0001 \\ -1.0322 \end{pmatrix} \text{ and } \overline{z}^* = \begin{pmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \\ 1.0322 \end{pmatrix}.$$

In the following tables, we list the total number of inner (noted In) and outer (noted Out) iterations and the cpu time (noted CPU) corresponding to the following values of the displacement step $\overline{\alpha} \in \{0.5, 0.6, 0.7\}$ and practical step size, denoted $\overline{\alpha}_p$, defined in [6, 16]. **Bold font** is used to highlight the smallest number of inner and outer iterations achieved.

Table 1

Inner and outer iteration counts and CPU time for Example 1 for $p = 2$.

	$\bar{\theta}/\bar{\alpha}$	0.5	0.6	0.7	$\bar{\alpha}_p$
In Out CPU	0.01	1943 1943 2.3870	1943 1943 2.5336	1943 1943 2.3455	1943 1943 2.2534
In Out CPU	0.1	186 186 0.3720	186 186 0.3620	186 186 0.3638	186 186 0.3686
In Out CPU	0.5	131 29 0.1663	108 29 0.1607	92 29 0.1583	70 29 0.1699
In Out CPU	0.9	134 9 0.1451	110 9 0.1394	93 9 0.1318	**70** 9 0.1411

Table 2

Inner and outer iteration counts and CPU time for Example 2 for $p = 2$.

	$\bar{\theta}/\bar{\alpha}$	0.5	0.6	0.7	$\bar{\alpha}_p$
In Out CPU	0.01	1971 1971 2.4397	1971 1971 2.5771	1971 1971 2.6577	1971 1971 2.5476
In Out CPU	0.1	188 188 0.4024	188 188 0.3886	188 188 0.3780	188 188 0.3840
In Out CPU	0.5	132 29 0.1757	109 29 0.1741	92 29 0.1653	70 29 0.1744
In Out CPU	0.9	136 9 0.1559	112 9 0.1455	95 9 0.1359	**71** 9 0.1521

Table 3

Inner and outer iteration counts and CPU time for Example 1 for $p = \frac{1}{2}(\log n - 2)$.

	$\bar{\theta} / \bar{\alpha}$	0.5	0.6	0.7	$\bar{\alpha}_p$
In Out CPU	0.01	1943 1943 2.2908	1943 1943 2.5501	1943 1943 2.3035	1943 1943 2.3365
In Out CPU	0.1	186 186 0.3389	186 186 0.3435	186 186 0.3381	186 186 0.3419
In Out CPU	0.5	42 29 0.1451	33 29 0.1398	29 29 0.1425	29 29 0.1555
In Out CPU	0.9	35 9 0.1117	26 9 0.1149	22 9 0.1083	**14** 9 0.1233

Table 4

Inner and outer iteration counts and CPU time for Example 2 for $p = \frac{1}{2}(\log n - 2)$.

	$\bar{\theta} / \bar{\alpha}$	0.5	0.6	0.7	$\bar{\alpha}_p$
In Out CPU	0.01	1971 1971 2.0717	1971 1971 2.4858	1971 1971 2.3345	1971 1971 2.3462
In Out CPU	0.1	188 188 0.3607	188 188 0.3530	188 188 0.3488	188 188 0.3535
In Out CPU	0.5	47 29 0.1476	37 29 0.1443	29 29 0.1426	29 29 0.1607
In Out CPU	0.9	40 9 0.1231	31 9 0.1133	26 9 0.1132	**17** 9 0.1236

4.1 *Comments*

From Tables 1-4, we conclude that:

- For both examples, the algorithm using the proposed KF runs and yields the optimal solution for all different values of the displacement step.
- Numerical tests confirm the efficiency of the parameter p and demonstrate that the smallest number of iterations was attained in all experiments when $p = \frac{1}{2}(\log n - 2)$.

References

[1] E. I. Abd El-Latif and M. I. Moussa, "Information hiding using artificial DNA sequences based on Gaussian kernel function," *Journal of Information and Optimization Sciences,* vol. 40, no. 6, pp. 1181-1194 (2019). doi: 10.1080/02522667.2017.1413041.

[2] M. Achache, "A new primal-dual path-following method for convex quadratic programming," *Computational & Applied Mathematics,* vol. 25, no. 1, pp. 97-110 (2006).

[3] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Mathematical Programming*, Series B, vol. 95, pp. 3–51 (2003).

[4] Y. Q. Bai, M. E. L. Ghami, and C. Roos, "A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization," *SIAM Journal on Optimization,* vol. 15, pp. 101-128 (2004).

[5] M. Bouafia, D. Benterki, and A. Yassine, "An efficient primal-dual interior-point method for linear programming problems based on a new kernel function with a trigonometric barrier term," *Journal of Optimization Theory and Applications,* vol. 170, pp. 528-545 (2016).

[6] N. Boudjellal, H. Roumili, and D. Benterki, "A primal-dual interior-point algorithm for convex quadratic programming based on a new parametric kernel function," *Optimization* (2020). doi: 10.1080/02331934.2020.1751156.

[7] S. Guerdouh, W. Chikouche, and I. Touil, "An efficient primal-dual interior-point algorithm for linear optimization problems based on a novel parameterized kernel function with a hyperbolic barrier term," (2021). halshs-03228790.

[8] F. S. Hafshejani and A. F. Jahromi, "An interior point method for P*(k)–horizontal linear complementarity problem based on a new

proximity function," *Journal of Applied Mathematics and Computation* (2019). doi: 10.1007/s12190-019-01284-9.

[9] F. S. Hafshejani and Z. Moaberfard, "An interior-point algorithm for linearly constrained convex optimization based on kernel function and application in non-negative matrix factorization," *Optimization Engineering*, vol. 21, pp. 1019–1051 (2020). doi: 10.1007/s11081-020-09514-x.

[10] F. S. Hafshejani and Z. Moaberfard, "A new trigonometric kernel function for support vector machine," *Iran Journal of Computer Science*, vol. 6, no. 2, pp. 1-9 (2022). doi: 10.1007/s42044-022-00130-9.

[11] E. H. Houssein, A. Hamad, A. E. Hassanien, and A. A. Fahmy, "Epileptic detection based on whale optimization enhanced support vector machine," *Journal of Information and Optimization Sciences,* vol. 40, no. 3, pp. 699-723 (2019). doi: 10.1080/02522667.2018.1453671.

[12] M. A. Noor, "On the nonlinear complementarity problem," *Journal of Mathematical Analysis and Applications*, vol. 123, no. 2, pp. 455-460 (1987).

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

[14] I. Touil and D. Benterki, "A primal-dual interior-point method for the semidefinite programming problem based on a new kernel function," *Journal of Nonlinear Functional Analysis*, vol. 2019, Article ID 25 (2019).

[15] I. Touil and W. Chikouche, "Primal-dual interior point methods for semidefinite programming based on a new type of kernel functions," *Filomat*, vol. 34, no. 12, pp. 3957-3969 (2020).

[16] I. Touil and W. Chikouche, "Polynomial-time algorithm for linear programming based on a kernel function with hyperbolic-logarithmic barrier term," *PJM*, vol. 11, pp. 127-135 (2022).

Received February, 2023