*AMO - Advanced Modeling and Optimization, Volume 15, Number 2, 2013*

# Feasible short-step interior point algorithm for linear complementarity problem based on kernel function

El Amir Djeffal[a,1] and Lakhdar Djeffal[a]

[a] *Department of Mathematics, Hadj Lakhdar University of Batna, Algeria*

**Abstract.** In this paper we deal with the study of the polynomial complexity analysis and numerical implementation for a short-step interior point algorithm for monotone linear complementarity problems $(LCP)$ based on karnel function. The analysis is based on a new class of search directions. We establish the global convergence of the algorithm. Furthermore, it is shown that the algorithm has $O(n^{2.5}L)$, iteration complexity. For its numerical tests some strategies are used and indicate that the algorithm is efficient.

**Keyword(s)**. Quadratic programming, Convex nonlinear programming, Interior point methods
AMS subject classification**.** 90C30, 90C51

## 1. Introduction

Let us consider the linear complementarity problem $(LCP)$: find vectors $x$ and $y$ in real space $\Re^n$ that satisfy the following conditions:

$$x \geq 0,\ y = Mx + q \geq 0 \text{ and } x^t y = 0, \tag{1.1}$$

where $q$ is a given vector in $\Re^n$ and $M$ is a given $n \times n$ real matrix. $LCP$ have important applications in mathematical programming and various areas of engineering. Interior-point methods $(IPMs)$ for solving Linear Optimization $(LO)$ problems were initiated by Karmarkar [2]. They not only have polynomial complexity, but are also highly efficient in practice. Feasible $IPMs$ start with a strictly feasible interior-point and maintain feasibility during the solution process. Feasible $IPMs$ require that the starting points satisfy exactly the equality constraints and are strictly positive, i.e., they lie in the interior of a region defined by constraints. Extending methods for $LO$ to $LCP$ has been successful in many cases. See, e.g.,[3, 4]. Recently, Peng et al. [5, 6] designed

[1] *Corresponding Author.*
*E-mail addresses: djeffal_elamir@yahoo.fr (El Amir Djeffal), lakdar_djeffal@yahoo.fr (Lakhdar Djeffal)*
*AMO-Advanced modeling and optimization. ISSN:1841-4311*

primal–dual feasible $IPMs$ by using self-regular functions for $LO$ and also extended the approach to $LCP$. In this paper we deal with the complexity analysis and the numerical implementation of a short-step interior point algorithm. This algorithm is based on the strategy of the central path and on a method for finding a new search directions, where we show that this short-step algorithm deserves the best current polynomial complexity namely $O(n^{2.5}L)$. The paper is organized as follows. In the next section, the statement of the problem is presented, we deal with the weighted vector introduced to ensure that the initial point $(x^0, y^0, \mu^0)$ verified $\delta(x^0, y^0, \mu^0) = 0$, (proximity measure define bellow). In Section $3$, we deal with the new search directions and the description of the algorithm. In Section $4$, we state its polynomial complexity. Section $5$ contains the numerical experiments. In Section $6$, a conclusion and remarks are given.

We use the classical notation. In particular, $\Re^n$ denotes the $n$-dimensional Euclidean space. Given $u, v \in \Re^n$, $u^t v = \sum_{i=1}^{n} u_i v_i$ is their inner product, and $\|u\| = \sqrt{u^t u}$ is the Euclidean norm. Given a vector $d \in \Re^n$, $D = diag(d)$ is the $n \times n$ diagonal matrix. $I$ is the identity matrix and $e$ is the identity vector.

## 2. Statement of the problem

The feasible set, the strictly feasible set and the solution set of $(1.1)$ are denoted, respectively by

$$\boldsymbol{F} = \left\{(x, y) \in \Re^{2n} : y = Mx + q, x \geq 0, y \geq 0\right\},$$

$$\boldsymbol{F}_{int} = \left\{(x, y) \in \boldsymbol{F} : x > 0, y > 0\right\},$$

and

$$\Omega = \left\{(x, y) \in \boldsymbol{F} : x \geq 0, y \geq 0, x^t y = 0\right\}.$$

In this paper, we assume that the following assumptions hold.
**Assumption 1**. $F_{int} \neq \emptyset$.
**Assumption 2**. $M$ is a positive semidefinite matrix.
In addition $(1.1)$, is equivalent to the following convex quadratic problem, see, e.g.,[7].

$$\min\left\{x^t y : x \geq 0, y \geq 0, y = Mx + q\right\}. \tag{2.1}$$

Hence, finding the solution of $(1.1)$ is equivalent to find the minimizer of $(2.1)$ with its objective value is zero.

In order to introduce an interior point method to solve $(2.1)$, we associate with it the following barrier minimization problem

$$\min\left\{f_{\mu r}(x, y) : y = Mx + q, x > 0, y > 0\right\}, \tag{2.2}$$

where $f_{\mu r}(x, y) = x^t y - \mu \sum_{i=1}^{n} r_i \log(x_i y_i),\ \mu > 0$ be the barrier parameter and $r = (r_1, ..., r_n) \in \Re^n_+$ is a weighted vector introduced to ensure that the initial point $(x^0, y^0)$ verified $\delta(x^0 y^0, \mu^0) = 0$ (proximity measure define bellow), if $r_i = 1, i = 1, ..., n,$ then the weighted central path coincides with the classical one. Hence, this approach can be seen as a generalization of central path methods.

The problem $(2.2)$ is a convex optimization problem and then its first order optimality conditions are:

$$\begin{cases} Mx + q = y, \\ xy = \mu r,\ x > 0, y > 0. \end{cases} \tag{2.3}$$

If the Assumptions 1 and 2 hold then for a fixed $\mu > 0$, the problem $(2.2)$ and the system $(2.3)$ have a unique solution [7] denoted as $(x(\mu), y(\mu))$, with $x(\mu) > 0$ and $y(\mu) > 0$. We call $(x(\mu), y(\mu))$, with $\mu > 0$, the $\mu$-centers of $(2.3)$. The set of the $\mu$-centers defines the so-called the central path of $(1.1)$.

In the next section, we introduce a method for tracing the central path based a new class of search directions.

## 3. A new search directions

Now, the basic idea behind this approach is to replace the non linear equation:

$$\frac{xy}{\mu r} = e,$$

in $(2.3)$ by an equivalent equation

$$\psi(\frac{xy}{\mu r}) = \psi(e),$$

where $\psi$, is a real valued function on $[0, \infty)$ and differentiable on $[0, \infty)$ such that $\psi(t)$ and $\psi'(t) > 0$, for all $t > 0$. Then the system $(2.3)$ can be written as the following equivalent form:

$$\begin{cases} Mx + q = y,\ x > 0, y > 0 \\ \psi(\frac{xy}{\mu r}) = \psi(e). \end{cases} \tag{3.1}$$

Suppose that we have $(x, y) \in F_{int}$. Applying Newton's method for the system $(2.3)$, we obtain a new class of search directions:

$$\begin{cases} M\triangle x = \triangle y, \\ \frac{y}{\mu r}\psi'(\frac{xy}{\mu r})\triangle x + \frac{x}{\mu r}\psi'(\frac{xy}{\mu r})\triangle y = \psi(e) - \psi(\frac{xy}{\mu r}). \end{cases} \tag{3.2}$$

Now, the following notations are useful for studying the complexity of the proposed algorithm. The vectors

$$v = \sqrt{\frac{xy}{\mu r}},\ d = \sqrt{xy^{-1}},$$

these notations lead to

$$\frac{d^{-1}x}{\sqrt{\mu r}} = \frac{dy}{\sqrt{\mu r}} = v.$$

Denote by

$$d_x = \frac{d^{-1}\triangle x}{\sqrt{\mu r}},\ d_y = \frac{d\triangle y}{\sqrt{\mu r}}, \tag{3.3}$$

and hence, we have

$$\mu r v(d_x + d_y) = y\triangle x + x\triangle y, \tag{3.4}$$

and

$$d_x d_y = \frac{\triangle x \triangle y}{\mu r} \tag{3.5}$$

So using $(3.3)$ and $(3.4)$, the system $(3.2)$ becomes

$$\begin{cases} \overline{M} d_x = d_y, \\ d_x + d_y = p_v, \end{cases}$$

where $\overline{M} = MDM$ with $D = diag(d)$
and

$$p_v = \frac{\psi(e) - \psi(v^2)}{v\psi'(v^2)}.$$

We shall consider the following function:

$$\psi(t) = \frac{1}{2}(t^2 - 1),\ \text{with } \psi^{'}(t) = t \text{ for all } t > 0.$$

Hence, the Newton directions in $(3.2)$ is

$$\begin{cases} M\triangle x = \triangle y, \\ d_x + d_y = \frac{1}{2}(v^{-3} - v), \end{cases} \tag{3.6}$$

with

$$p_v = \frac{1}{2}(v^{-3} - v),$$

and we define for all vector $v$ the following proximity measure by

$$
\begin{aligned}
\delta(xy, \mu) &= \frac{\|p_v\|_2}{2}, \\
&= \left\|v^{-3} - v\right\|_2, \\
&= \left\|\left(\sqrt{\frac{xy}{\mu r}}\right)^{-3} - \sqrt{\frac{xy}{\mu r}}\right\|_2.
\end{aligned}
$$

Now, the generic short-step primal-dual algorithm to solve $LCP$ has the following form

### 3.1. Algorithm

**Begin algorithm**

**Input:**

an accuracy parameter $\varepsilon > 0$,

an update parameter $\theta$, $0 < \theta < 1$ (default $\theta = \frac{1}{2\sqrt{n}}$),

a strictly feasible point $(x^0, y^0)$ and $\mu^0 = \frac{(x^0)^t \ Ry^0}{n}$.

$\sigma = \dfrac{\|X^0Y^0e\|}{\sqrt{n}}$ , $r = \dfrac{X^0Y^0e}{\sigma}.R = diag(r_i)$

$k = 0$

**While** $(n\mu^k) > \varepsilon$ **do**

$1^\circ$) Compute $(\triangle x, \triangle y)$,

$2^\circ$) Update $(x^{k+1}, y^{k+1}) = (x^k, y^k) + (\triangle x, \triangle y)$

$3^\circ$) Set $\mu^{k+1} = (1-\theta)\mu^k = (1-\theta)\frac{x^k Ry^k}{n}$ and $k = k+1$.

**End While.**

**End algorithm.**

## 4. Complexity analysis

Let

$$p_v = d_x + d_y, \; q_v = d_x - d_y,$$

and, we have

$$d_x = \frac{1}{2}(p_v + q_v), \; d_y = \frac{1}{2}(p_v - q_v),$$

hence,

$$d_x d_y = \frac{1}{4}(p_v^2 - q_v^2) \text{ and } \|q_v\| \leq \|p_v\|.$$

This last result follows directly from the equality

$$\|p_v\|^2 = \|q_v\|^2 + 4d_x^t d_y,$$

since,

$$d_x^t d_y = d_x^t \overline{M} d_x \geq 0, \text{because } \overline{M} \text{ is positive semidefinite.}$$

We have

$$\delta(v, \mu) \geq \|q_v\|.$$

In the following lemma, we state a condition which ensures the feasibility of the full Newton step.

Let

$$\begin{aligned} x^+ &= x + \triangle x, \\ y^+ &= y + \triangle y, \end{aligned}$$

be the new iterate after a full Newton step.

**Lemma 1.** *Let $(x, y)$ is a strictly feasible iteration. If $e + d_x d_y > 0$ then*

$$(x^+, y^+) = (x + \triangle x, y + \triangle y)$$

*Proof.* Let $0 < \alpha \leq 1$ is step lenght.

We define:

$$x(\alpha) = x + \alpha\triangle x, \ y(\alpha) = y + \alpha\triangle y,$$

we have

$$\begin{aligned} x(\alpha)y(\alpha) &= (x + \alpha\triangle x)(y + \alpha\triangle y) \\ &= xy + \alpha(x\triangle y + s\triangle x) + \alpha^2\triangle x\triangle y \\ &= xy + \alpha(\mu r - xy) + \alpha^2\triangle x\triangle y. \end{aligned}$$

We assume that $e + d_x d_y > 0$,
we deduce $\mu r + \triangle x\triangle y > 0$, which is equivalent to $\triangle x\triangle y > -\mu r$, by substitution we obtain

$$\begin{aligned} x(\alpha)s(\alpha) &> xs + \alpha(\mu r - xs) - \alpha^2\mu r \\ &= (1-\alpha)xy + (\alpha - \alpha^2)\mu r \\ &= (1-\alpha)xy + \alpha(1-\alpha)\mu r. \end{aligned}$$

Since

$xy > 0$ and $\mu r > 0$, it follows that $x(\alpha)y(\alpha) > 0$ for all $\alpha \in ]0, 1]$. □

Now for convenience,we may write

$$(v^+)^2 = \frac{x^+y^+}{\mu r} = e + d_x d_y.$$

**Lemma 2.** *If* $\delta(xy, \mu) < 1$,*Then* $x^+ > 0$ *and* $y^+ > 0$.

*Proof.* In the Lemma 1 , we have $(x^+, y^+)$ are strictly feasible if $(e+d_xd_y) > 0$. So $(e+d_xd_y) > 0$ holds if $(1 + (d_xd_y)_i) > 0$ for all $i \in \Re^n$.

We have

$$\begin{aligned}(1 + (d_xd_y)_i) &\geq (1 - |(d_xd_y)_i| \,, \text{ for all } i \in \Re^n \\ &\geq (1 - \delta^2).\end{aligned}$$

Thus $(e + d_xd_y) > 0$ if $\delta(xy, \mu) < 1$. □

In the next lemma we proved the local quadratic convergence for our algorithm

**Lemma 3.** *Let* $\delta = \delta(xy, \mu) < 1$ *then*

$$\delta(x^+y^+, \mu^+) \leq \frac{\delta^2}{\sqrt{2(1 - \delta^2)}}.$$

*Proof.* letting $\alpha = 1$,

we have

$$\begin{aligned}4\delta_+^2 &= \left\|(v^+)^{-1} - v^+\right\|^2 \\ &= \left\|(v^+)^{-1}(e - (v^+)^2)\right\|^2,\end{aligned}$$

where $(v^+)^2 = (e - d_xd_y)$ and $(v^+)^{-1} = \frac{1}{\sqrt{(e+d_xd_y)}}$, then it follows that

$$\begin{aligned}4\delta_+^2 &= \left\|\frac{d_xd_s}{\sqrt{(e + d_xd_y)}}\right\|_2^2 \\ &= \left\|\frac{d_xd_s}{\sqrt{(e + d_xd_y)}}\right\|_2^2 \\ &\leq \frac{\|d_xd_y\|_2^2}{(1 - \|d_xd_y\|_\infty)}.\end{aligned}$$

We deduce that

$$4\delta_+^2 \leq \frac{2\delta_+^4}{(1-\delta_+^2)}.$$

This proves the lemma. □

**Lemma 4.** *Let* $\delta(xy,\mu) < \frac{1}{\sqrt{2}}$ *and* $\mu^+ = (1-\theta)\mu,\ 0 < \theta < 1$. *Then*

$$\delta^2(x^+y^+,\mu^+) \leq (1-\theta)\delta_+^2 + \frac{\theta^2(n+1)}{4(1-\theta)} + \frac{\theta}{2}.$$

*Furthermore, if* $\delta \leq \frac{1}{\sqrt{2}},\ \theta = \frac{1}{2\sqrt{n}}$ *and* $n \geq 2$, *then we have* $\delta(x^+y^+,\mu^+) \leq \frac{1}{\sqrt{2}}$.

*Proof.* Let $v^+ = \sqrt{\frac{x^+y^+}{\mu^+r}}$ and $\mu^+ = (1-\theta)\mu$, then

$$\begin{aligned}
4\delta^2(x^+y^+,\mu^+) &= \left\|\sqrt{\frac{\mu^+r}{x^+y^+}} - \sqrt{\frac{x^+y^+}{\mu^+r}}\right\|_2^2 \\
&= \left\|\sqrt{1-\theta}(v^+)^{-1} - \frac{1}{\sqrt{1-\theta}}v^+\right\|_2^2 \\
&= \left\|\sqrt{1-\theta}\left((v^+)^{-1} - v^+\right) - \frac{\theta}{\sqrt{1-\theta}}v^+\right\|_2^2 \\
&= (1-\theta)\left\|(v^+)^{-1} - v^+\right\|_2^2 + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta\left((v^+)^{-1} - v^+\right)^t v^+ \\
&= (1-\theta)\left\|(v^+)^{-1} - v^+\right\|_2^2 + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta(v^+)^{-t}v^+ + {v^+}^t v^+ \\
&= 4(1-\theta)\delta_+^2 + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta n + 2\theta\left\|v^+\right\|_2^2
\end{aligned}$$

since,
$(v^+)^{-t}v^+ = n$ and $(v^+)^t v^+ = \|v^+\|_2^2$, and we have $\delta_+^2 \leq \frac{\delta_+^4}{2(1-\delta_+^2)}$.Then

$$4\delta^2(x^+y^+,\mu^+) \leq 4(1-\theta)\frac{\delta^4}{2(1-\delta_+^2)} + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta n + 2\theta\left\|v^+\right\|_2^2,$$

since,

$$x^+s^+ = \mu r(e + d_x d_y),$$

and if $\delta < \frac{1}{\sqrt{2}}$, it follows

$$\left\|v^+\right\|^2 = \frac{x^+y^+}{\mu r} = e + d_x d_y \leq 1 + n.$$

Consequently,

$$4\delta^2(x^+y^+,\mu^+) \leq 4(1-\theta)\frac{\delta^4}{2(1-\delta_+^2)} + \frac{\theta^2}{1-\theta}\left\|v^+\right\|_2^2 - 2\theta n + 2\theta(n+1),$$

and

$$\delta^2(x^+y^+,\mu^+) \leq (1-\theta)\delta_+^4 + \frac{\theta^2(n+1)}{4(1-\theta)} + \frac{\theta}{2}.$$

The last statement the proof goes as follows. If $\delta < \frac{1}{\sqrt{2}}$ , then $\delta_+^2 = \frac{1}{4}$ and this yields the following upper bound for $\delta(x^+y^+,\mu^+)$ as:

$$\delta^2(x^+y^+,\mu^+) \leq \frac{(1-\theta)}{4} + \frac{\theta^2(n+1)}{4(1-\theta)} + \frac{\theta}{2}.$$

Now, taking $\theta = \frac{1}{2\sqrt{n}}$ then $\theta^2 = \frac{1}{4n}$ if follows that

$$\delta^2(x^+y^+,\mu^+) \leq \frac{\frac{(n+1)}{4n}}{4(1-\theta)} + \frac{(1-\theta)}{4} + \frac{\theta}{2},$$

since $\frac{(n+1)}{4n} \leq \frac{3}{8}$ for all $n \geq 2$ then we have

$$\delta^2(x^+y^+,\mu^+) \leq \frac{3}{32(1-\theta)} + \frac{\theta+1}{4}.$$

Now for $n \geq 2$, we have $0 < \theta \leq \frac{1}{2\sqrt{2}}$ and since the function $f(\theta) = \frac{3}{32(1-\theta)} + \frac{\theta+1}{4}$ is continuous and monotonic increasing on $0 < \theta \leq \frac{1}{2\sqrt{2}}$,
consequently,

$$f(\theta) \leq f(\frac{1}{2\sqrt{2}}) < \frac{1}{2}, \text{ for all } 0 < \theta \leq \frac{1}{2\sqrt{2}}.$$

Hence,

$$\delta(x^+y^+,\mu^+) < \frac{1}{2}.$$

□

The following theorem gives an upper bound for the total number of iteration for our algorithm.

**Theorem 5.** *Let $\varepsilon > 0$ be an accuracy parameter. The algorithm has a complexity bound of $O(\sqrt{n}L)$ iterations and the total complexity bound of the algorithm is:*
*$O(n^{2.5}L)$, where $L = \log\frac{\mu^0}{\varepsilon}$.*

*Proof.* We have

$$\mu^k = (1-\theta)^k \mu^0,$$

thus,

$$(1-\theta)^k \mu^0 \leq \varepsilon.$$

Now taking logarithms of $(1-\theta)^k \mu^0 \leq \varepsilon$, we may write

$$\log((1-\theta)^k \mu^0) \leq \log \varepsilon$$

equivalent

$$k \log(1-\theta) \leq \log \varepsilon - \log \mu^0,$$

using the fact that $\log(1-\theta) \leq \theta$, for $0 \leq \theta \leq 1$, then the above inequality holds if

$$k \geq \frac{1}{\theta}\left[\log \frac{\mu^0}{\varepsilon}\right].$$

Let $L = \log \frac{\mu^0}{\varepsilon}$, then at most $k = 2\sqrt{n} \log \frac{\mu^0}{\varepsilon} = O(\sqrt{n} \log \frac{\mu^0}{\varepsilon}) = O(\sqrt{n}L)$ iterations in the algorithm, we can obtain $\varepsilon$-solution of (2.3). However, in every step, the complexity bound of computing the linear system is $O(n^2)$. Therefore, the total complexity bound of the algorithm is $O(n^{2.5}L)$. □

# 5. Numerical implementation

In this section, we deal with the numerical implementation of this algorithm applied to some problems of monotone $LCPs$. Here we used $(x^0, y^0)$ to denote the feasible starting point of the algorithm, $\delta(x^0 y^0, \mu^0) < \frac{1}{2}$, the proximity condition, $(x^*, y^*)$ the optimal solution of $LCP$ and **Iter** means the iterations number produced by the algorithm. $z^*$ denotes the value of the objective function at $(x^*, y^*)$ and $\mu^*$ denotes the value when the algorithm terminates. The implementation is manipulated in **DEV C++**. Our tolerance is $\varepsilon = 10^{-6}$. For the update parameter we have vary $0 < \theta < 1$. Finally we note that the linear system of Newton in $(3.2)$ is solved thanks to the Gauss elimination procedure.

**Problem 1.**

$$M = \begin{pmatrix} 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ -2 & -1 & 0 & 0 & 0 \\ -1 & -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{pmatrix}, q = \begin{pmatrix} -4 & -5 & 8 & 7 & 3 \end{pmatrix},$$

The feasible starting point is

$x^0 = \begin{pmatrix} 2 & 2 & 2 & 2 & 2 \end{pmatrix}, y^0 = \begin{pmatrix} -4 & -5 & 8 & 7 & 3 \end{pmatrix}$

$\delta(x^0 y^0, \mu^0) = 2.517539 > \frac{1}{2}$, then the classical method is diverge.
The numerical results with this problem are summarized in the table below:

| | | |
|---|---|---|
| | **Results of the Algorithm** | |
| | $\delta(x^0 y^0, \mu^0) = 0.000000 < \frac{1}{2}$ | |
| $\theta = 0.15$ | **iter** | 87 |
| | $x^*$ | $\begin{pmatrix} 2.999995 & 2.000002 & 1.000001 & 2.000001 & 0.000005 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.000003 & 0.000007 & 0.000009 & 0.000002 & 0.999998 \end{pmatrix}$ |
| | $z^*$ | 0.000041 |
| | $\mu^*$ | 0.000009 |
| $\theta = 0.20$ | **iter** | 65 |
| | $x^*$ | $\begin{pmatrix} 2.999994 & 2.000002 & 1.000001 & 2.000000 & 0.000005 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.000003 & 0.000008 & 0.000010 & 0.000002 & 0.999998 \end{pmatrix}$ |
| | $z^*$ | 0.000045 |
| | $\mu^*$ | 0.000010 |
| $\theta = 0.60$ | **iter** | 24 |
| | $x^*$ | $\begin{pmatrix} 2.999994 & 2.000002 & 1.000001 & 2.000001 & 0.000004 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.000003 & 0.000006 & 0.000008 & 0.000002 & 0.999998 \end{pmatrix}$ |
| | $z^*$ | 0.000037 |
| | $\mu^*$ | 0.000008 |
| $\theta = 0.95$ | **iter** | 19 |
| | $x^*$ | $\begin{pmatrix} 2.999993 & 2.000003 & 1.000002 & 2.000000 & 0.000004 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.000002 & 0.000006 & 0.000012 & 0.000002 & 0.999997 \end{pmatrix}$ |
| | $z^*$ | 0.000036 |
| | $\mu^*$ | 0.000008 |

**Problem 2.**

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 0.8 & 0.32 & 1.128 & 0.0512 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0.32 & 0.128 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0.32 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.128 & -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0512 & -1.128 & -0.32 & -0.8 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$q = \begin{pmatrix} -0.0256 & -0.064 & -0.16 & 5.59 & -1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$
The feasible starting point is

$x^0 = \begin{pmatrix} 0.18 & 0.18 & 0.18 & 0.18 & 0.25 & 3 & 4 & 5 & 6 & 9 \end{pmatrix},$

$y^0 = \begin{pmatrix} 21.0032 & 11.008 & 12.52 & 12.8 & 8 & 0.46 & 0.676 & 0.6184 & 0.41536 & 0.336144 \end{pmatrix}.$

$\delta(x^0y^0, \mu^0) = 2.278802 > \frac{1}{2}$, then the classical method is diverge.

The numerical results with this problem are summarized in the table below:

| | | **Results of the Algorithm** |
|---|---|---|
| | | $\delta(x^0y^0, \mu^0) = 0.000000 < \frac{1}{2}$ |
| $\theta = 0.15$ | **iter** | 84 |
| | $x^*$ | $\begin{pmatrix} 0.000036 & 0.886520 & 0.000003 & 0.000001 & 0.000001 \\ 0.000005 & 0.000088 & 0.000039 & 0.000013 & 8.035358 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.385925 & 0.000008 & 2.411364 & 12.028297 & 7.035360 \\ 0.999892 & 0.113452 & 0.290770 & 0.716269 & 0.000001 \end{pmatrix}$ |
| | $z^*$ | 0.000085 |
| | $\mu^*$ | 0.000009 |
| $\theta = 0.20$ | **iter** | 64 |
| | $x^*$ | $\begin{pmatrix} 0.000036 & 0.886520 & 0.000003 & 0.000001 & 0.000001 \\ 0.000005 & 0.000088 & 0.000039 & 0.000013 & 8.035358 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.385925 & 0.000008 & 2.411364 & 12.028297 & 7.035360 \\ 0.999892 & 0.113452 & 0.290770 & 0.716269 & 0.000001 \end{pmatrix}$ |
| | $z^*$ | 0.000095 |
| | $\mu^*$ | 0.000010 |
| $\theta = 0.60$ | **iter** | 24 |
| | $x^*$ | $\begin{pmatrix} 0.000024 & 0.886521 & 0.000002 & 0.000001 & 0.000001 \\ 0.000003 & 0.000058 & 0.000026 & 0.000009 & 8.035394 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.385887 & 0.000005 & 2.411360 & 12.028322 & 7.035394 \\ 0.999927 & 0.113460 & 0.290773 & 0.716283 & 0.000001 \end{pmatrix}$ |
| | $z^*$ | 0.000065 |
| | $\mu^*$ | 0.000007 |
| $\theta = 0.95$ | **iter** | 19 |
| | $x^*$ | $\begin{pmatrix} 0.000047 & 0.886521 & 0.000003 & 0.000001 & 0.000000 \\ 0.000001 & 0.000032 & 0.000025 & 0.000004 & 8.035417 \end{pmatrix}$ |
| | $y^*$ | $\begin{pmatrix} 0.385854 & 0.000002 & 2.411361 & 12.028336 & 7.035416 \\ 0.999859 & 0.113441 & 0.290765 & 0.716257 & 0.000000 \end{pmatrix}$ |
| | $z^*$ | 0.000048 |
| | $\mu^*$ | 0.000005 |

**Problem 3.** Let $M \in \Re^{n\times n}$ and $q \in \Re^n$are given by:

$$M = \begin{pmatrix} 1 & 2 & 2 & . & . & . & 2 \\ 0 & 1 & 2 & . & . & . & 2 \\ 0 & 0 & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & 2 \\ 0 & 0 & 0 & . & . & 0 & 1 \end{pmatrix}, q = \begin{pmatrix} -1 & . & . & . & -1 \end{pmatrix},$$

Case 1: $n = 10$.

The feasible starting point is

$x^0 = \begin{pmatrix} 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 1.0009 \end{pmatrix},$

$y^0 = \begin{pmatrix} 1.0171 & 1.0153 & 1.0135 & 1.0108 & 1.0099 & 1.0081 & 1.0063 & 1.0045 & 1.0027 & 0.0009 \end{pmatrix}.$

$\delta(x^0y^0, \mu^0) = 0.032154 < \frac{1}{2}$, then the classical method is converge.

The numerical results with this problem are summarized in the table below:

| | **Results of the Algorithm** $(r = e)$ |
|---|---|
| | $\delta(x^0y^0, \mu^0) = 0.032154 < \frac{1}{2}$ |
| $\theta = 0.15$ | **iter** 31 <br> $x^* \begin{pmatrix} 0.000009 & 0.000009 & 0.000009 & 0.000009 & 0.000009 \\ 0.000009 & 0.000009 & 0.000009 & 0.000009 & 1.000009 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.000168 & 1.000151 & 1.000133 & 0.999215 & 1.000097 \\ 1.000080 & 1.000062 & 1.000044 & 1.000027 & 0.000009 \end{pmatrix}$ <br> $z^*$ 0.000089 <br> $\mu^*$ 0.000009 |
| $\theta = 0.20$ | **iter** 23 <br> $x^* \begin{pmatrix} 0.000009 & 0.000009 & 0.000009 & 0.000009 & 0.000009 \\ 0.000009 & 0.000009 & 0.000009 & 0.000009 & 1.000010 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.000180 & 1.000161 & 1.000142 & 0.999223 & 1.000104 \\ 1.000085 & 1.000066 & 1.000047 & 1.000028 & 0.000009 \end{pmatrix}$ <br> $z^*$ 0.000095 <br> $\mu^*$ 0.000009 |
| $\theta = 0.60$ | **iter** 9 <br> $x^* \begin{pmatrix} 0.000007 & 0.000007 & 0.000007 & 0.000007 & 0.000007 \\ 0.000007 & 0.000007 & 0.000007 & 0.000007 & 1.000007 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.000129 & 1.000115 & 1.000102 & 0.999188 & 1.000074 \\ 1.000061 & 1.000047 & 1.000034 & 1.000020 & 0.000007 \end{pmatrix}$ <br> $z^*$ 0.000068 <br> $\mu^*$ 0.000007 |
| $\theta = 0.95$ | **iter** 7 <br> $x^* \begin{pmatrix} 0.000007 & 0.000007 & 0.000007 & 0.000007 & 0.000007 \\ 0.000007 & 0.000007 & 0.000007 & 0.000007 & 1.000007 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.000138 & 1.000124 & 1.000109 & 0.999194 & 1.000080 \\ 1.000065 & 1.000050 & 1.000036 & 1.000021 & 0.000007 \end{pmatrix}$ <br> $z^*$ 0.000073 <br> $\mu^*$ 0.000007 |

Case 2: $n = 15$.

The feasible starting point is

$$x^0 = \begin{pmatrix} 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 0.0009 \\ & & & 0.0009 & 0.0009 & 0.0009 & 0.0009 & 1.0009 & & \end{pmatrix},$$

$$y^0 = \begin{pmatrix} 1.0261 & 1.0243 & 1.0225 & 1.0198 & 1.0189 & 1.0171 & 1.0153 & 1.0135 & 1.0108 & 1.0099 \\ & & & 1.0081 & 1.0063 & 1.0045 & 1.0027 & 0.0009 & & \end{pmatrix}.$$

$\delta(x^0y^0, \mu^0) = 0.059169 < \frac{1}{2}$, then the classical method is converge.

The numerical results with this problem are summarized in the table below:

| | **Results of the Algorithm** $(r = e)$ |
|---|---|
| | $\delta(x^0y^0, \mu^0) = 0.032154 < \frac{1}{2}$ |
| $\theta = 0.15$ | **iter** 15 <br> $x^* \begin{pmatrix} 0.000097 & 0.000097 & 0.000097 & 0.000097 & 0.000097 \\ 0.000097 & 0.000097 & 0.000097 & 0.000097 & 0.000097 \\ 0.000097 & 0.000097 & 0.000097 & 0.000097 & 1.000097 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.002816 & 1.002622 & 1.002428 & 1.001334 & 1.002040 \\ 1.001846 & 1.001652 & 1.001458 & 1.000363 & 1.001069 \\ 1.000875 & 1.000680 & 1.000486 & 1.000292 & 0.000097 \end{pmatrix}$ <br> $z^*$ 0.001458 <br> $\mu^*$ 0.000097 |
| $\theta = 0.20$ | **iter** 12 <br> $x^* \begin{pmatrix} 0.000084 & 0.000084 & 0.000084 & 0.000084 & 0.000084 \\ 0.000084 & 0.000084 & 0.000084 & 0.000084 & 0.000084 \\ 0.000084 & 0.000084 & 0.000084 & 0.000084 & 1.000084 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.002448 & 1.002280 & 1.002111 & 1.001042 & 1.001773 \\ 1.001605 & 1.001436 & 1.001267 & 1.000198 & 1.000929 \\ 1.000760 & 1.000591 & 1.000423 & 1.000253 & 0.000085 \end{pmatrix}$ <br> $z^*$ 0.001268 <br> $\mu^*$ 0.000085 |
| $\theta = 0.60$ | **iter** 5 <br> $x^* \begin{pmatrix} 0.000061 & 0.000061 & 0.000061 & 0.000061 & 0.000061 \\ 0.000060 & 0.000060 & 0.000060 & 0.000060 & 0.000060 \\ 0.000060 & 0.000060 & 0.000060 & 0.000060 & 1.000060 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.001750 & 1.001629 & 1.001508 & 1.000486 & 1.001265 \\ 1.001145 & 1.001024 & 1.000903 & 0.999882 & 1.000662 \\ 1.000541 & 1.000421 & 1.000300 & 1.000180 & 0.000060 \end{pmatrix}$ <br> $z^*$ 0.000906 <br> $\mu^*$ 0.000060 |
| $\theta = 0.95$ | **iter** 4 <br> $x^* \begin{pmatrix} 0.000060 & 0.000059 & 0.000059 & 0.000059 & 0.000059 \\ 0.000059 & 0.000058 & 0.000058 & 0.000058 & 0.000058 \\ 0.000058 & 0.000057 & 0.000057 & 0.000057 & 1.000057 \end{pmatrix}$ <br> $y^* \begin{pmatrix} 1.001688 & 1.001569 & 1.001450 & 1.000432 & 1.001214 \\ 1.001097 & 1.000980 & 1.000863 & 0.999847 & 1.000631 \\ 1.000515 & 1.000400 & 1.000286 & 1.000171 & 0.000057 \end{pmatrix}$ <br> $z^*$ 0.000874 <br> $\mu^*$ 0.000058 |

# 6. Conclusion

In this paper, we have proposed a feasible short-step interior point algorithm for solving monotone linear complementarity problem. The algorithm deserves the best wellknown theoretical iteration bound $O(n^{2.5}L)$ when the starting point is $(x^0, y^0)$ is strictly feasible and verified proximity measure condition. This choice of initial point can be done by the technique of Djamel Benterki [1]. For the numerical tests we vary the parameter $\theta$, and we note that each problem addressed when the parameter $\theta$ crosses we get the good numerical behavior. Future research might extended the algorithm for other optimization problems.

**Acknowledgements**

The authors thank the referees for their careful reading and their precious comments. Their help is much appreciated.

# References

[1] D. Benterki, A. Leulmi. An improving procedure of the interior projective method for linear programming. Applied Mathematics and Computation $199$ $(2008)$ $811-819$

[2] N.K. Karmarkar, A new polynomial-time algorithm for linear programming, in: Proceedings of the 16th Annual ACM Symposium on Theory of Computing, $1984,\ pp.\ 302-311$.

[3] J. Ji, F.A. Potra, S. Huang, Predictor–corrector method for linear complementarity problems with polynomial complexity and superlinear convergence, J. Optim. Theory Appl. $85$ $(1)$ $(1995)$ $187-199$.

[4] F.A. Potra, A superlinearly convergent predictor–corrector method for degenerate $LCP$ in a wide neighborhood of the central path with $O(\sqrt{n}L)$ iteration complexity, Math. Program. $100$ $(2)$ $(2004)$ $317-337$.

[5] J. Peng, C. Roos, T. Terlaky, Self-regular functions and new search directions for linear and semidefinite optimization, Math. Program. $93$ $(1)$ $(2002)$ $129-171$.

[6] J. Peng, C. Roos, T. Terlaky, Self-Regularity. A New Paradigm for Primal–Dual Interior-Point Algorithms, Princeton University Press, $2002$.

[7] S.J. Wright, Primal-dual Interior-Point Methods, SIAM, Philadelphia, USA, $1997$.