People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Ferhat Abbas University, Sétif 1
Faculty of Sciences
Department of Mathematics

# Doctoral thesis

**Field: Mathematics and Computer Science**
**Option: Mathematics**
**Specialty: Optimization and Control**

**Theme:**

# Complementarity problems and Interior-point methods

Presented by

Mr. **Welid GRIMES**

Supervisor: **Pr. Mohamed ACHACHE**

Thesis defended on July 1st , 2023, in front of the jury composed of:

| | | | | |
|---|---|---|---|---|
| Mrs. | Zakia KEBBICHE | Prof. | Ferhat Abbas University, Sétif 1 | President |
| Mr. | Mohamed ACHACHE | Prof. | Ferhat Abbas University, Sétif 1 | Supervisor |
| Mr. | Mousaab BOUAFIA | Dr. | 8 Mai 1945 University, Guelma | Examiner |
| Mrs. | Hassina GRAR | Dr. | Ferhat Abbas University, Sétif 1 | Examiner |
| Mrs. | Yamina BENCHEIKH | Prof. | Ferhat Abbas University, Sétif 1 | Invited |

**July 2023**

# Acknowledgments

I would like to express my gratitude to my research supervisor, **Prof. Mohamed Achache**, for giving me the opportunity to conduct research and providing invaluable guidance throughout this work. It was a great privilege and honor to work under his guidance.

I would also like to thank, **Prof. Zakia Kebbiche** for her kind acceptance to be the president of jury committee, as well as **Dr. Hassina Grar** and **Dr. Mousaab Bouafia** who have kindly accepted the invitation to be examiners of my thesis. My gratitude to **Prof. Yamina Bencheikh** for their kind acceptance to take part in the jury member invited.

Special thanks are also due to **Prof. Adnan Yassine** for very helpful observations and fruitful discussions while I visited Le Havre University.

Thanks to my **colleagues and all members of Laboratory of Fundamental and Numerical Mathematics,** University Ferhat Abbas, Sétif 1.

I would like to thank **my parents, brothers and sisters** for their continues encouragement during my whole education life.

***Welid Grimes***

# Dedication

*To my dear parents,*

*To my brothers,*

*To my sisters and their children,*

*To my friends.*

# Contents

**Contents** iv

**List of Tables** vi

**List of publications** 1

**Glossary of Notation** 2

**Introduction** 5

**1 Background on complementarity problems and IPMs** 9

1.1 Nonlinear complementarity problems . . . . . . . . . . . . . . . . . . 9

1.2 Some problems formulated as nonlinear complementarity problems . . 10

1.3 Linear complementarity problems . . . . . . . . . . . . . . . . . . . . 11

1.4 Some problems formulated as linear complementarity problems . . . . 13

1.5 Some results of the existence and uniqueness solution of CPs . . . . . . 15

1.6 Solution methods for CPs . . . . . . . . . . . . . . . . . . . . . . . . . 16

1.7 Some ways to determine search directions . . . . . . . . . . . . . . . . 17

**2 Path-following IPA for monotone LCP** 19

2.1 The LCP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

2.2 The interior-point algorithm and its complexity analysis . . . . . . . . . 21

2.3 Numerical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

2.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

**3 A path-following IPA for MLCP based on a modified Newton search direction** 31

3.1 Preliminaries and the problem statement . . . . . . . . . . . . . . . . 32

3.2 The modified search directions for LCP . . . . . . . . . . . . . . . . . 33

3.3 Analysis of the algorithm . . . . . . . . . . . . . . . . . . . . . . . . . 36
3.4 Numerical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
3.5 Conclusion and remarks . . . . . . . . . . . . . . . . . . . . . . . . . 47

**4 A Newton descent logarithmic barrier IPA for MLCP 49**

4.1 Preliminaries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.2 The barrier problem $P_\mu$ associated to the monotone LCP . . . . . . . . 50
4.3 Numerical aspects of $P_\mu$ . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.4 Numerical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
4.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64

**5 Complexity analysis of an IPA for $P_\star(\kappa)-$NCP 65**

5.1 Preliminaries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
5.2 Complexity analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
5.3 Implementation and numerical results . . . . . . . . . . . . . . . . . 72
5.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78

**General conclusion and future work 79**

**Appendices 81**

.1 Appendix A . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
.2 Appendix B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83

**Bibliography 88**

# List of Tables

2.1 Numerical results for **Problem 1.** . . . 27

2.2 Numerical results for **Problem 2.** . . . 28

2.3 Numerical results for **Problem 3** by **Algorithm 2.2.1.** . . . 29

2.4 Numerical results for **Problem 3** by **Algorithm 2.3** . . . 29

2.5 Numerical results for **Problem 3** by **Algorithm 2.1** . . . 29

3.1 Number of iterations and CPU time for **Problems 1,2 and 3.** . . . 45

3.2 Number of iterations and CPU time for **Problems 1,2 and 3.** . . . 46

3.3 Number of iterations and CPU time of the ameliorated Algorithm for **NETLIB** set Problems. . . . 47

4.1 Number of iterations and CPU time for **Problem 1**, where $\mu = 0.5$ . . . 62

4.2 Number of iterations and CPU time for **Problem 2**, where $\mu = 0.5$ . . . 63

4.3 Number of iterations and CPU time for **Problem 3**, where $\mu = 0.4$ . . . 63

5.1 Number of iterations and CPU time for **Problem 3.** . . . 74

5.2 Number of iterations and CPU time for **Problem 4.** . . . 75

5.3 Numerical results for **Problem 1**. by **Ameliorated Algorithm** . . . 76

5.4 Numerical results for **Problem 2**. by **Ameliorated Algorithm** . . . 77

5.5 Numerical results for **Problem 3**. by **Ameliorated Algorithm** . . . 77

5.6 Numerical results for **Problem 4**. by **Ameliorated Algorithm** . . . 77

5.7 Numerical results for **Csizmadia matrix**. . . . 78

# List of publications

- W. Grimes, M. Achache, 'An Infeasible Interior-point Algorithm for Monotone Linear Complementarity Problems', ***International Journal of Informatics and Applied Mathematics,*** **04 (02), 53 – 59 (2021).**
  https://doi.org/10.53508/ijiam.1036022

- W. Grimes, 'Path-following interior-point algorithm for monotone linear complementarity problems', ***Asian-European Journal of Mathematics,*** **15(9) (2022)**
  .
  https://doi.org/S1793557122501704

- W. Grimes, M. Achache, 'A path-following interior-point algorithm for monotone LCP based on a modified Newton search direction', ***RAIRO-Operations Research,*** **(2023)**
  https://doi.org/10.1051/ro/2023054

# Glossary of Notations

## Problem Classes and Terminology

| | | |
|---|---|---|
| CP | : | Complementarity problem; |
| NCP | : | Non-Linear complementarity problem; |
| LCP | : | Linear complementarity problem; |
| MLCP | : | Monotone linear complementarity problem; |
| LP | : | Linear programming or linear optimization; |
| QP | : | Quadratic programming; |
| CQO | : | Convex quadratic optimization; |
| P | : | Primal of a mathematical programming; |
| D | : | Dual of (P); |
| SNE | : | System of nonlinear equations; |
| VIP | : | Variational inequalities problem; |
| AET | : | Algebraic equivalent transformation; |
| IP | : | Interior-Point; |
| IPMs | : | Interior-Point Methods; |
| IPAs | : | Interior-Point Algorithms; |
| IPC | : | Interior-Point-condition; |
| K.K.T | : | Karush-Kuhn-Tucker; |
| PSD | : | Positive semi-definite; |
| PD | : | Positive definite; |
| MIN1 | : | The strategy based on the first minorant function; |
| MIN2 | : | The strategy based on the second minorant function; |
| MAJ1 | : | The strategy based on the first majorant function; |
| MAJ2 | : | The strategy based on the second majorant function; |
| ITER | : | The number of iterations to obtain a solution of LCP; |
| CPU | : | The required time (in seconds) to obtain a solution of LCP; |

## Spaces and Orthants

| | | |
|---|---|---|
| $\mathbb{R}$ | : | the set of real numbers ; |
| $\mathbb{R}^n$ | : | the real $n$-dimensional space; |
| $\mathbb{R}^n_+$ | : | the nonnegative orthant in $\mathbb{R}^n$; |
| $\mathbb{R}^n_{++}$ | : | the positive orthant in $\mathbb{R}^n$; |
| $\mathbb{R}^{n\times n}$ | : | the space of all $n \times n$ real squared matrices; |

## Matrices

| | | |
|---|---|---|
| $M$ | : | $= (m_{ij})$; a matrix with entries $m_{ij}$; |
| $M^T$ | : | the transpose of a matrix $M$; |
| $M^{-1}$ | : | the inverse of a matrix $M$; |
| $I_n$ | : | identity matrix of order $n$; |
| $X = diag(x)$ | : | the diagonal matrix with diagonal elements equal to the components of the vector $x$ with $X_{ii} = x_i$; |

## Vectors

| | | |
|---|---|---|
| $e$ | : | $= (1, \dots, 1)^T$; vector of ones; |
| $x^T$ | : | $= (x_1, \dots, x_n)$; the transpose of avector $x$ with components $x_i$; |
| $xy$ | : | $= (x_1y_1, \dots, x_ny_n)^T$; Hadamard product; |
| $x^Ty = \langle x, y\rangle$ | : | $= \sum_{i=1}^{n} x_iy_i$; the standard inner product of $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$; |
| $\sqrt{x}$ | : | $= (\sqrt{x_1}, \dots, \sqrt{x_n})^T$, $(x \geq 0)$; |
| $x^{-1}$ | : | $= \left(\frac{1}{x_1}, \dots, \frac{1}{x_n}\right)^T$ $(x \neq 0)$; |
| $\frac{x}{y}$ | : | $= \left(\frac{x_1}{y_1}, \dots, \frac{x_n}{y_n}\right)^T$ $(y \neq 0)$; |
| $\lvert x\rvert$ | : | the absolute value of $x \in \mathbb{R}^n$; |
| $\Vert x\Vert$ | : | the Euclidian norm of $x \in \mathbb{R}^n$; |
| $\Vert x\Vert_\infty$ | : | $= \max_{i=1,\dots,n} \lvert x_i\rvert$; the maximum norm of $x \in \mathbb{R}^n$; |
| $\max(x)$ | : | the maximal component of the vector $x$; |
| $\min(x)$ | : | the minimal component of the vector $x$; |
| $(x^0, y^0)$ | : | the initial point; |
| $(x^\star, y^\star)$ | : | the solution of complementarity problem; |

## Functions

| | | |
|---|---|---|
| $\nabla F$ | : | the gradient of a function $F : \mathbb{R}^n \to \mathbb{R}^n$; |
| $\nabla^2 F$ | : | the hessian matrix of $F$; |
| $\dfrac{\partial f_i}{\partial x_j}(x)$ | : | the partial derivative of $f_i$ at $x_j$; |

# Introduction

This thesis deals with the complexity analysis and numerical implementation of some interior-point methods for solving some classes of complementarity problems (abbreviated CP). The CPs in finite dimension consists to find a vector pair $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ satisfying

$$y = F(x),\ x^T y = 0,\ x \geq 0,\ y \geq 0, \tag{CP}$$

where $F(x) : \mathbb{R}^n \to \mathbb{R}^n$ is a given continuously differentiable vector function, i.e., $F(x) = (F_1(x), \cdots, F_n(x))^T$. Here the inequality $(\geq)$ and $x^T y = 0$ indicate the non negativity and the orthogonality of the vectors $x$ and $y$, respectively.

The systematic study of CP began in the mid-1960s and during the last decades it became an important topic of research in mathematical programming and numerical optimization. This problem reflects many applications areas such as operations research, bi-matrix games, variational inequalities, economics, engineering and equilibrium modeling [35, 36] and they have been the subject of much research interest. Also, it generalizes the so-called standard linear complementarity problems (LCP) [2, 20, 38], i.e., when $F(x) = Mx + q$ (affine function) where $M \in \mathbb{R}^{n\times n}$ is a given matrix and $q \in \mathbb{R}^n$. The LCPs includes linear and convex quadratic optimization (for more details we refer the reader to the monograph of Cottle, Pang and Stone [20] ). Also the CPs is a particular case of the semi-definite complementarity problems (SDCP) which acts on the cone of symmetric semi-definite matrices ([4]).

For the solution of CPs, a number of direct as well as iterative methods have been proposed and analyzed. Besides, the direct famous pivoting like simplicial Lemke's algorithm for LCPs, smoothing and non-smoothing methods [4, 40] to solving CPs, the interior-point methods gained much more attention than others due to their polynomial complexity and their highly practical efficiency. These methods showed first their powerful for solving the linear optimization (LO) since Karmarkar's work [44]. For more details concerning IPMs we direct the reader to monographs in references

[67, 73]. These methods have been extended successfully to many types of optimization problems such as convex quadratic optimization (CQO) [1, 73], linear semidefinite problems (SDO) [64], second-order cone optimization (SOCO) [13].

In the literature of interior-point methods, we distinguish four important types of them, notably,

1. Projective methods,
2. Affine-scaling methods,
3. Potential-reduction methods,
4. Path-following methods or central-path methods.

Interior point methods can be classified based on the length of the step and the neighborhood used. In this way, the methods that generate new iterations within smaller neighborhood of the central path are referred to as short update methods, while those that use a wider neighborhood are called large update methods.

In this thesis, our fundamental contributions import on the complexity analysis and the numerical implementation of some path-following interior point methods for solving monotone LCPs and $P_{\star}(\kappa)-$NCPs based on classical and new search directions and on efficient step-sizes (displacement step). The determination of a new search direction and computing efficient step-size on it, have a great effect in improving the complexity analysis of these algorithms and their practical performances. For this purpose, we cite the paradigm of barrier kernel functions which has offered new Newton directions which leads to improving the polynomial complexity for large-step algorithm from $\mathcal{O}\left(n\log\frac{n}{\epsilon}\right)$ which based on the classical logarithmic barrier methods to $\mathcal{O}\left(\sqrt{n}\log n\log\frac{n}{\epsilon}\right)$. However, for the short-step these algorithms deserve the best well-known iterations bound, namely $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$ (see, e.g. [1, 5, 14, 15, 64, 63]). Subsequently, in 2002, an other important proposal presented by Darvay [29] which offered new search directions based on the so called algebraic equivalent transformations (AETs) induced by univariate kernel functions applied to the centrality equations. The application of Newton's methods to the new modified system leads to offer new Newton descent search directions. These methods based AETs enjoy also the best known polynomial complexity for short-step algorithms. Besides, in some interior-point methods based logarithmic barrier approach, the computation of the displacement step (step-size) is usually based on adopting classical line search rule such as Armijo, Wolfe ect... It is shown that these procedures (rules) cost very expensive in computing this step-size [19]. Recently some new procedure are suggested to remedy this drawback. Among them the novel strategy of the so-called majorant and minorant (lower and upper bound) functions for computing the step-size which is exploited by many authors for solving the following optimization problems such as LO, CQO and SDO (see [13, 18, 23, 55, 58]). In this thesis, we have exploited this procedure only for monotone LCP to improve its numerical efficiency for getting a solution of monotone LCPs.

## Short outline of the thesis

The thesis contains five chapters, followed by a bibliography. This thesis is organized as follows

**Chapter 1.** In this chapter, a mathematical background is stated and which will be utile throughout the thesis.

**Chapter 2.** This chapter deals with the study of the short-step full-Newton step primal-dual interior-point algorithm for solving monotone LCP. Under a new suggestion of the defaults of the barrier parameter $\theta$ and the threshold $\tau$, we show that this algorithm has the best-known polynomial complexity, namely, $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$ where $n$ is the dimension of the variables. Its efficiency is also confirmed by presenting some numerical experiments on some monotone LCPs.

**Chapter 3.** In this chapter, we propose a short-step feasible full-Newton step path-following interior-point algorithm (IPA) for monotone linear complementarity problems (MLCPs). The proposed IPA uses the technique of algebraic equivalent transformation (AET) induced by an univariate function to transform the centering equations which defines the central-path. By applying Newton's method to the modified system of the central-path of LCP, a new Newton search direction is obtained. Under new appropriate defaults of the threshold $\tau$ which defines the size of the neighborhood of the central-path and of $\theta$ which determines the decrease in the barrier parameter, we prove that the IPA is well-defined and converges locally quadratically to a solution of the monotone LCPs. Moreover, we derive its iteration bound, namely, $\mathcal{O}(\sqrt{n}\log\frac{n}{\epsilon})$ which coincides with the best-known iteration bound for such algorithms. Finally, some numerical results are presented show its efficiency.

**Chapter 4.** In this chapter, based on optimization techniques, we propose a descent logarithmic barrier interior-point method for solving monotone linear complementarity problems (MLCP). The idea is to transform the MLCP as a convex quadratic optimization (CQO). Then an associated barrier problem is stated. The existence and the uniqueness of the optimal solution of the latter is showed and its convergence to a solution of MLCP is proved. For its numerical aspects, the descent direction is computed by using classical Newton's method. However, the displacement step along this direction for maintaining points interior during the algorithm process, a novel strategy of the so-called minorant and majorant approximating functions is applied. The obtained numerical results are very promising.

**Chapter 5.** In this chapter, we establish the polynomial complexity of a feasible short-step path-following interior-point algorithm (IPA) for solving the class of nonlinear complementarity problems with $P_\star(\kappa)$-mapping (abbreviated $P_\star(\kappa)$-NCP). The proposed algorithm uses only full Newton steps with the advantage that no line search is required. We prove that the algorithm is well defined and converges locally quadratically to a solution of NCP. Moreover, we show that the algorithm has $\mathcal{O}\left(\sqrt{n}(1+4\kappa)\log\left(\frac{n}{\epsilon}\right)\right)$ as a complexity result for short-update methods. Some numerical results are provided to show the efficiency of our algorithm.

Finally, we end the thesis by a conclusion and suggestions for future work. Moreover, some well-known results used in this thesis are summarized in Appendices.

Chapter 1

# *Background on Complementarity problems and Interior-point methods*

In this chapter, we give some relevant elements of the theory of complementarity problems (CPs) and a brief survey on interior-point methods (IPMs).

## 1.1 Nonlinear complementarity problems

In mathematics context and in optimization, a nonlinear complementarity problem NCP consists to find vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ such that

$$y = F(x), \ x^T y = 0, \ x \geq 0, \ y \geq 0, \tag{NCP}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function, the expression $x^T F(x)$ denotes the inner product of $x$ and $F(x)$ in $\mathbb{R}^n$, and the inequalities are understood to be component-wise. From the orthogonality of $x$ and $F(x)$, it turns to require that the component-wise product $x_i^T F_i(x) = 0$ for all $i = 1, \ldots, n$.

In the sequel, the feasible set and the strictly feasible set of NCP are denoted, respectively, by

$$\mathcal{F} = \{(x, y) \in \mathbb{R}^{2n}_+ \, : y = F(x)\},$$

$$\mathcal{F}^s = \{(x, y) \in \mathcal{F} \, : x > 0, \, y > 0\}.$$

The solution set of NCP is given by

$$\mathcal{F}^\star = \text{sol}(LCP) = \{(x^\star, y^\star) \in \mathcal{F} \, : (x^\star)^T y^\star = 0\}.$$

**Definition 1.1.** If $x \in \mathcal{F}$ or $(x \in \mathcal{F}^s)$ then $x$ is called a feasible (or strictly feasible) solution of NCP.

### 1.1.1 Some classes of nonlinear complementarity problems

In this subsection, we cite some important classes of NCPs [41].

- The class of monotone functions:

$$\forall x, y \in \mathbb{R}^n : \ \langle x - y, F(x) - F(y) \rangle \geq 0.$$

- The class of strictly monotone functions:

$$\forall x, y \in \mathbb{R}^n \text{ and } x \neq y : \ \langle x - y, F(x) - F(y) \rangle > 0.$$

- The class of $P_0$-functions:

$$\forall x, y \in \mathbb{R}^n \text{ and } x \neq y : \ \max_{1 \leq i \leq n : x_i \neq y_i} (x_i - y_i)(F_i(x) - F_i(y)) \geq 0.$$

- The class of $P$-functions:

$$\forall x, y \in \mathbb{R}^n \text{ and } x \neq y : \ \max_{1 \leq i \leq n} (x_i - y_i)(F_i(x) - F_i(y)) > 0.$$

- The class of $P_\star(\kappa)$-functions: for any $x \neq y$ in $\mathbb{R}^n$ and $\kappa \geq 0$

$$(1 + 4\kappa) \sum_{i \in \mathcal{I}_+(x,y)} (x_i - y_i)(F_i(x) - F_i(y)) + \sum_{i \in \mathcal{I}_-(x,y)} (x_i - y_i)(F_i(x) - F_i(y)) \geq 0,$$

where

$\mathcal{I}_+(x,y) = \{i : (x_i - y_i)(F_i(x) - F_i(y)) \geq 0\}$ and $\mathcal{I}_-(x,y) = \{i : (x_i - y_i)(F_i(x) - F_i(y)) < 0\}$.

## 1.2 Some problems formulated as nonlinear complementarity problems

### 1.2.1 Variational inequalities problem

The variational inequalities problem (VIP) is defined as: Find $\bar{x} \in C$ such that

$$(x - \bar{x})^T F(x) \geq 0, \text{ for all } x \in C, \tag{VIP}$$

where $C$ is nonempty subset of $\mathbb{R}^n$ and $F$ a mapping from $\mathbb{R}^n$ into itself. If $C = \mathbb{R}^n_+$, then the (VIP) reduces to an NCP (See [42]).

### 1.2.2 System of nonlinear equations

The constrained system of nonlinear equations:

$$h(x, y) = 0,\ x \geq 0,\ y \geq 0, \tag{SNE}$$

where $h : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is defined by

$$h(x, y) = \begin{pmatrix} F(x) - y \\ Yx \end{pmatrix},$$

and $Y$ is the diagonal matrix. The (SNE) is equivalent to the NCP because if the pair vectors $(x^\star, y^\star)$ solves (SNE), then the vector $x^\star$ solves NCP where $y^\star = F(x^\star)$, and conversely (See [4]).

## 1.3 Linear complementarity problems

If $F$ is a linear function i.e. $F(x) = Mx + q$ where $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$, then the problem CP reduces to LCP.

$$x \geq 0,\ y \geq 0,\ y = Mx + q,\ x^T y = 0 \tag{LCP}$$

### 1.3.1 Classes of linear complementarity problems

The LCP has been studied for many different classes of matrices $M \in \mathbb{R}^{n\times n}$. We list some of them [20]:

- The class of skew-symmetric matrices (SS):

  $$\forall x \in \mathbb{R}^n\ ,\ x \neq 0 :\ x^T M x = 0.$$

- The class of positive semi-definite matrices (PSD):

  $$\forall x \in \mathbb{R}^n\ ,\ x \neq 0\ :\ x^T M x \geq 0.$$

  Additionally, if $x^T M x > 0, M$ is called positive definite (PD). The LCP with a PSD matrix $M$ is called monotone LCP abbreviated (MLCP).

- The class of $P$-matrix: Matrix with all principal positive minors or equivalently [31]

  $$\forall x \in \mathbb{R}^n \ , \ x \neq 0 : \ \exists i \in \{1, \dots, n\} \text{ s.t. } x_i(Mx)_i > 0.$$

- The class of $P_0$-matrix: Matrix with all principal non negative minors or equivalently [32]

  $$\forall x \in \mathbb{R}^n \ , \ x \neq 0 : \ \exists i \in \{1, \dots, n\} \text{ s.t. } x_i \neq 0 \text{ and } x_i(Mx)_i \geq 0.$$

- The class of $P_\star(\kappa)$ Matrix:

  $$(1 + 4\kappa) \sum_{i \in \mathcal{I}_+(x)} x_i(Mx)_i + \sum_{i \in \mathcal{I}_-(x)} x_i(Mx)_i \geq 0, \ \forall x \in \mathbb{R}^n,$$

  where $\kappa \geq 0$ and

  $$\mathcal{I}_+(x) = \{i : x_i(Mx)_i > 0\} \text{ and } \mathcal{I}_-(x) = \{i : x_i(Mx)_i < 0\}.$$

  The index sets $\mathcal{I}_+(x)$ and $\mathcal{I}_-(x)$ depend on the matrix $M$ though they do not explicitly represent the dependence. It should be noticed that the last inequality can be rewritten as

  $$x^T Mx + 4\kappa \sum_{i \in \mathcal{I}_+(x)} x_i(Mx)_i \geq 0, \ \forall x \in \mathbb{R}^n.$$

  Observe that $P_\star(0)$ is the set of positive semi-definite (PSD) matrices and $P_\star(\kappa_1) \subset P_\star(\kappa_2)$ if $0 \leq \kappa_1 \leq \kappa_2$.

  We call $M$ a $P_\star$-matrix if it belongs to the class $P_\star = \cup_{\kappa \geq 0} P_\star(\kappa)$.

The relationship between some of the above classes is as follows

$$SS \subset PSD, \ P \cap SS = \varnothing, \ (PSD \cup P) \subset P_\star \subset P_0.$$

Some of these relations are obvious, like $PSD = P_\star(0) \subset P_\star$ or $P \subset P_\star$, while others require a proof which can be found in [21, 52]

## 1.4 Some problems formulated as linear complementarity problems

### 1.4.1 Convex quadratic programming

Let us consider the following convex quadratic program:

$$\min_x \{c^T x + \frac{1}{2} x^T Q x \ : \ Ax \leq b, \ x \geq 0\}, \tag{QP}$$

where $Q$ is a symmetric semidefinite matrix in $\mathbb{R}^{n\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m\times n}$ with rank $(A) = m$.

By the **K.K.T.** optimality condition: $x$ is an optimal solution of QP, if and only if, there exists $y \in \mathbb{R}^m_+$, and $\lambda \in \mathbb{R}^n_+$ such that

$$\begin{cases} Qx + c + A^T y - \lambda = 0 \\ y^T(b - Ax) = 0 \\ \lambda^T x = 0 \end{cases} \Leftrightarrow \begin{cases} \lambda = Qx + c + A^T y \\ \mu = -Ax + 0y + b \\ \lambda^T x + \mu^T y = 0, x, y, \lambda, \mu \geq 0. \end{cases}$$

The above conditions define LCP, where

$$M = \begin{pmatrix} Q & A^T \\ -A & 0 \end{pmatrix}, \quad q = \begin{pmatrix} c \\ b \end{pmatrix}.$$

### 1.4.2 Linear programming

Let us consider the following linear program in its canonical form

$$\min_x \{c^T x \ : \ Ax \leq b, \ x \geq 0\}, \tag{P}$$

and its dual

$$\max_y \{b^T y \ : \ A^T y + z = c, \ z \geq 0\}, \tag{D}$$

inspired from the QP case, the associated LCP, is given by

$$M = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}, \quad q = \begin{pmatrix} c \\ b \end{pmatrix}.$$

**Result 1.2.** *In the corresponding LCP for convex quadratic programs (QP) and linear programming (P), the matrix $M$ is positive semi definite (PSD).*

### 1.4.3 Obstacle problems

The obstacle problem [59] can be formulated using the following equation and inequalities: find $u$ such that

$$(\ddot{u}(z) - f(z))^T(u(z) - g(z)) = 0, \quad \ddot{u}(z) - f(z) \geq 0, \quad u(z) - g(z) \geq 0,$$

where $f(z) : \mathbb{R}^n \to \mathbb{R}^n$, $g(z) : \mathbb{R}^n \to \mathbb{R}^n$, we approximate the second-order derivative $\ddot{u}$ with a second order centred finite difference to get a discrete version on an equispaced grid $z_i = ih, i = 1, ..., n$.

$$(Mu - f)^T(u - g) = 0, \quad Mu - f \geq 0, \quad u - g \geq 0,$$

where $g_i = g(z_i)$, $f_i = f(z_i)$, $M = \begin{pmatrix} \frac{2}{h^2} & \frac{-1}{h^2} & 0 & \dots & 0 \\ \frac{-1}{h^2} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{-1}{h^2} \\ 0 & \dots & 0 & \frac{-1}{h^2} & \frac{2}{h^2} \end{pmatrix}$.

This can be written as a linear complementarity problem by setting $x = u - g$ and $q = Mg - f$, that is

$$(Mx + q)^T x = 0, \; Mx + q \geq 0, \; x \geq 0.$$

### 1.4.4 Absolute value equation

Let us consider the following Absolute value equation (AVE)

$$Az - |z| = b, \tag{AVE}$$

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $z \in \mathbb{R}^n$, and $|z|$ denote the absolute value of $z$. Mangasarian [56] has proved that the Absolute value equation (AVE) is equivalent to the general LCP. Indeed, for $z \in \mathbb{R}^n$, we define $z^+ = \max(z, 0)$ and $z^- = \max(0, -z)$. Then it is easy to conclude that

$$z^+ \geq 0, \; z^- \geq 0, \; z = z^+ - z^-, \; |z| = z^+ + z^- \text{ and } (z^+)^T z^- = 0.$$

Therefore, the AVE is equivalent to the general LCP with coefficient matrix and affine vector:

$$M = (A+I)(A-I)^{-1}, \quad q = ((A+I)(A-I)^{-1} - I)b.$$

## 1.5 Some results of the existence and uniqueness solution of CPs

In the theory of the CPs, mapping classes of the input $F$ (including matrix $M$) play a major role, since the CPs shows different properties depending on a class of input mappings $F$ (including matrices $M$).

**Theorem 1.3** ([43]). *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous and strongly monotone function. Then NCP has a unique solution.*

**Theorem 1.4** ([35]). *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous and monotone function. If the set of strictly feasible points is not empty, then NCP has a non empty compact solution set.*

**Theorem 1.5** ([20]). *Let $M \in \mathbb{R}^{n\times n}$ be a positive semi-definite matrix. If the LCP is feasible, then it is solvable.*

This can be proved by showing that there exists a vector $u$ satisfying the Karush-Kuhn-Tucker (K.K.T.) condition of the quadratic programming problem QP:

$$\begin{cases} (M+M^T)x - M^T u + q \geq 0, \\ x^T((M+M^T)x - M^T u + q) = 0, \\ u \geq 0, \\ u^T(Mx+q) = 0. \end{cases}$$

Another known property of monotone LCPs is that the solutions set is convex.

**Theorem 1.6** ([52]). *If $M \in \mathbb{R}^{n\times n}$ is a positive definite matrix (PD), then the LCP has a unique solution for any $q \in \mathbb{R}^n$.*

**Theorem 1.7** ([52]). *A matrix $M \in \mathbb{R}^{n \times n}$ is a $P$-matrix if and only if the LCP has a unique solution for all vectors $q \in \mathbb{R}^n$.*

## 1.6 Solution methods for CPs

There are a variety of solutions approaches for CPs which have been studied intensively. Among them:

1. Smoothing and non-smoothing methods: By employing an appropriate merit or potential function, the NCP can be solved employing unconstrained minimization techniques [4, 34, 36, 41]. However, this approach generally involves non-smooth functions. The so called NCP functions, $\psi : \mathbb{R}^2 \to \mathbb{R}$, are such that $\psi(a, b) = 0$ implies $a \geq 0, b \geq 0$ and $ab = 0$. Then, a solution of the NCP can be obtained by solving the following nonlinear system of equations in

$$\begin{pmatrix} \psi(x_1, F_1(x)) \\ \vdots \\ \psi(x_n, F_n(x)) \end{pmatrix} = 0$$

2. Lemke's algorithm: is a pivoting algorithm introduced by Lemke [54] for solving LCPs and it is a generalization of Dantzig's Simplex Method developed earlier for LO based on the notion of complementarity pivots. Lemke's algorithm is known for its good numerical study, whereas in theory it is considered inefficient because of its exponential arithmetic complexity expressed by the total number of possible pivots. Nevertheless Lemke's algorithm was extremely useful.

3. Interior-point methods: Originally developed for linear optimization (LO), have been successfully extended to CPs [53, 73]. Researchers have proposed algorithms over the years that improve the effeciency of these methods [1, 10, 40, 41, 45, 57, 71], which have been a significant area of research in the field of op-

timization. The class of path-following primal-dual IPMs deserved much more attention due to their polynomial complexity and their practical efficiency

## 1.7 Some ways to determine search directions

Determining a search direction plays an important role in IPMs. In the last decade, several types of search directions have been proposed. Some of them are based on the strategy of so-called self-regular and kernel barrier functions. Meanwhile, others are based on the strategy of algebraic equivalent transformation (AET) technique.

### 1.7.1 Kernel functions

Kernel functions play an important role in the design of new primal-dual interior-point algorithms for solving CPs. Peng, Roos and Terlaky [64] introduced the notion of self-regular barriers and they determined new large-update IPMs with better iteration bounds than the long step interior-point algorithms studied before. Various IPMs have been successfully generalized to CPs [7, 11, 14, 16, 49, 70, 63].

**Definition 1.8.** [15] A function $\psi : \mathbb{R}_{++} \to \mathbb{R}_{+}$ is called kernel function if it is twice continuously differentiable and if the following conditions hold:

1. $\psi(1) = \psi'(1) = 0$
2. $\psi''(t) > 0$, for all $t > 0$;
3. $\lim\limits_{t\to 0} \psi(t) = \lim\limits_{t\to +\infty} \psi(t) = +\infty$.

It should be mentioned that in some cases in the literature condition 3. used to define the notion of coercive kernel function. This kernel function may be extended to a positive $v \in \mathbb{R}^n_+$ by

$$\Psi(v) = \sum_{i=1}^{n} \psi(v_i).$$

This function is nonnegative, and zero if and only if $v = e$ (i.e., $v$ is the vector of all-one).

### 1.7.2 Algebraic equivalent transformation

In 2002, Darvay [25], uses the AET technique based on the univariate function $\psi(t) = \sqrt{t}$ for LO. By means of this function, a new type of Newton direction is obtained and the best iteration bound for feasible short-step IPAs is derived. This method was extended successfully to convex quadratic optimization (CQO) and monotone LCP by Achache (see, e.g., [1, 2, 5]), semidefinite optimization (SDO) and second-order cone optimization (SCOP) by Wang and Bai (see, e.g.,[68, 69]). Besides, Kheirfam and Haghighi [46] investigated the AETs based on the function $\psi(t) = \dfrac{\sqrt{t}}{2(1+\sqrt{t})}$ to solve $P_*(\kappa)$-LCPs, where the best iteration bound is achieved for short-step methods. Subsequently, Haddou et al. [40] presented a generalized direction in interior-point methods for monotone LCP. Their approach is based on AET induced by the class of smooth concave univariate functions. By utilizing the AET technique based on the logarithmic function $\psi(t) = \log t$, Pan et al. [62] presented an infeasible IPA to solve LO. Furthermore, Darvay and Takács [28] developed a new IPA for LO based on a new modified search direction induced by an asymptotic barrier kernel function. The best polynomial complexity is provided. Recently, Darvay et al. [27], proposed an IPA for LO where their search direction is based on AET introduced by the new univariate function $\psi(t) = t - \sqrt{t}$. Later, Darvay et al. [30] generalized this algorithm for sufficient LCP.

# Chapter 2

# *Path-following interior-point algorithm for monotone linear complementarity problems*

In this chapter, a path-following full-Newton step interior-point algorithm (IPA) for solving monotone LCP is proposed. Under a new suggestion of the defaults of the barrier parameter $\theta$ and the threshold $\tau$, we show that this algorithm has the best-known polynomial complexity, namely, $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$ where $n$ is the dimension of the variables. Its efficiency is also confirmed by presenting some numerical experiments on some monotone LCPs.

## 2.1 The LCP

Recall again from Subsection 1.3, that the LCP consists in finding vectors $x, y$ in $\mathbb{R}^n$ such that

$$x \geq 0,\ y \geq 0,\ y = Mx + q,\ x^T y = 0 \tag{LCP}$$

where $M$ is a given $n \times n$ matrix and $q \in \mathbb{R}^n$. The notation $x \geq 0$ and $y \geq 0$ means that $x_i \geq 0$ and $y_i \geq 0$ for all $i = 1, ..., n$.

For the problem LCP, we assume that the following conditions hold [17].

- The matrix $M$ is positive semidefinite i.e., $\forall x \in \mathbb{R}^n : x^T M x \geq 0$ (LCP is called monotone).
- $\mathcal{F}^s = \{(x, y) \in \mathbb{R}^{2n} : y = Mx + q,\ x > 0,\ y > 0\ \} \neq \varnothing$, the set of all strictly feasible solutions of LCP, i.e. there exists an $x^0 > 0$ such that $y^0 = Mx^0 + q > 0$.

These conditions imply that there exist a solution for the monotone LCP. Also it is known [53] that finding a solution of LCP is equivalent to solving the following system:

$$\begin{cases} y = Mx + q, \\ xy = 0,\ x \geq 0,\ y \geq 0. \end{cases} \tag{2.1}$$

### 2.1.1 The central-path of LCP

The basic idea behind path-following interior-point methods is to replace $xy = 0$ in (2.1) by the perturbed equation $xy = \mu e$ where $\mu > 0$. Hence, we obtain the following system of equations:

$$\begin{cases} y = Mx + q \\ xy = \mu e,\ x > 0,\ y > 0. \end{cases} \tag{2.2}$$

Then for a fixed $\mu > 0$ system (2.2) has a unique solution denoted by $(x(\mu), y(\mu))$, which is called the $\mu-$center of LCP. The set of $\mu-$centers is called the central-path. If $\mu$ goes to zero then the limit of the central-path exists and converges to a solution of LCP [53, 78, 73].

### 2.1.2 The Newton search directions and proximity

Next, we want to define search directions $(\Delta x, \Delta y)$ that move in direction of the $\mu-$centers $(x(\mu), y(\mu))$. Applying Newton's method to (2.2) for a given strictly feasible point $(x, y)$ then the Newton direction $(\Delta x, \Delta y)$ at this point is the unique solution of the following linear system [67, 78, 73]:

$$\begin{cases} \Delta y = M\Delta x, \\ X\Delta y + Y\Delta x = \mu e - xy. \end{cases}$$

This last can be written as:

$$\begin{pmatrix} -M & I \\ Y & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} 0 \\ \mu e - xy \end{pmatrix}, \tag{2.3}$$

where $X$ =diag$(x)$ and $Y$ =diag$(y)$. For simplicity, we use the following notations

$$v := \sqrt{\frac{xy}{\mu}},\ d_x := \frac{v\Delta x}{x},\ d_y := \frac{v\Delta y}{y},\ d := \sqrt{\frac{x}{y}}. \tag{2.4}$$

Using (2.4), it yields the following scaled Newton system:

$$\begin{cases} d_y = \bar{M} d_x, \\ \\ d_x + d_y = p_v, \end{cases} \tag{2.5}$$

where $\bar{M} = DMD$ with $D = \text{diag}(d)$ and $p_v = v^{-1} - v$.

For the analysis of the algorithm, we use a norm-based proximity measure $\delta(v)$ defined by [67]:

$$\delta := \delta(xy; \mu) = \frac{\|p_v\|}{2}. \tag{2.6}$$

Clearly,

$$\delta(v) = 0 \Leftrightarrow v = e \Leftrightarrow xy = \mu e.$$

Hence, the value of $\delta(v)$ can be considered as a measure for the distance between the given pair $(x, y)$ and the corresponding $\mu-$center $(x(\mu), y(\mu))$.

Furthermore, we define the $\tau$-neighborhood of the central path as follows:

$$\mathcal{N}(\tau, \mu) = \{(x, y) : \ x > 0, \ y = Mx + q > 0 : \ \delta(xy; \mu) \leq \tau\},$$

where $\tau$ is a threshold parameter and $\mu > 0$ is fixed.

## 2.2 The interior-point algorithm and its complexity analysis

In this section, we describe the path-following full-Newton step IPA for MLCP. The algorithm starts with a strictly feasible initial point $(x^0, y^0)$ such that $\delta(x^0y^0; \mu_0) \leq \tau$ where $0 < \tau < 1$ and for a certain $\mu_0 > 0$ is known. The full-Newton step between successive iterates for the system (2.2) is defined as $(x_+, y_+) = (x + \Delta x, y + \Delta y)$ where the Newton's direction $\Delta x$ and $\Delta y$ are solutions for the linear system (2.3). Then it updates the parameter $\mu$ by the factor $1 - \theta$ with $0 < \theta < 1$, and target a new $\mu-$center and so on. This procedure is repeated until the stopping criterion $x^T y \leq \epsilon$ is satisfied for a given accuracy parameter $\epsilon$.

Therefore, the generic path-following full-Newton step interior-point algorithm for LCP is described in Algorithm 1 as follows.

**Algorithm 2.2.1** Full-Newton step IPA for monotone LCP

```
Initialize:
$k = 0$, $\mu^{(0)} > 0$, $(x^{(0)}, y^{(0)}) > 0$;
an accuracy parameter $\epsilon > 0$;
a barrier update parameter $0 < \theta < 1$; default $\theta = \frac{1}{\sqrt{2(n+1)}}$;
a threshold parameter $0 < \tau < 1$; default value $\tau = \frac{1}{\sqrt{2}}$;
Ensure: $(x^{(0)}, y^{(0)}) \in \mathcal{N}(\tau, \mu^{(0)})$
while $n\mu \geq \varepsilon$ do
    Compute $(\Delta x, \Delta y)$ from (2.3);
    Set $x^{(k+1)} = x^{(k)} + (\Delta x)^{(k)}$ and $y^{(k+1)} = y^{(k)} + (\Delta y)^{(k)}$;
    Set $\mu^{(k+1)} := (1-\theta)\mu^{(k)}$;
    Set $k = k + 1$;
End
```

### 2.2.1 Analysis of the short-step IPA

In this subsection, we give a condition which guarantees the feasibility of the full-Newton step. We start to give the following lemma which is a useful technical result that will be used later in the analysis

**Lemma 2.1** ([12]). *Let $\delta = \delta(xy;\mu) > 0$ where $\mu > 0$ and $(d_x, d_y)$ be the unique solution of system (2.5). Then one has*

$$0 \leq d_x^T d_y \leq 2\delta^2, \tag{2.7}$$

*and*

$$\|d_x d_y\|_\infty \leq \delta^2, \; \|d_x d_y\| \leq \sqrt{2}\delta^2. \tag{2.8}$$

**Lemma 2.2.** *Let $(x, y) \in \mathcal{F}^s$ be a strictly feasible primal-dual point, then $x_+ = x + \Delta x$ and $y_+ = y + \Delta y$ are positive if and only if $e + d_x d_y > 0$.*

*Proof.* We have,

$$x_+ y_+ = (x + \Delta x)(y + \Delta y) = xy + x\Delta y + y\Delta x + \Delta x \Delta y.$$

Due to (2.4), we have $\mu d_x d_y = \Delta_x \Delta_y$, and $x\Delta_y + y\Delta_x = \mu v(d_x + d_y)$, then we get

$$x_+ y_+ = \mu(e + d_x d_y).$$

If $x_+ > 0$ and $y_+ > 0$ then $x_+ y_+ > 0$ and so $e + d_x d_y > 0$. Conversely, for each $0 \leq \alpha \leq 1$ denote $x(\alpha) := x + \alpha \Delta x$, $y(\alpha) := y + \alpha \Delta y$. Therefore,

$$x(\alpha)y(\alpha) = xy + \alpha(\mu e - xy) + \alpha^2 \Delta x \Delta y.$$

If $e + d_x d_y > 0$ then $\mu e + \Delta x \Delta y > 0$ and $\Delta x \Delta y > -\mu e$. Hence

$$x(\alpha)y(\alpha) > (1-\alpha)(xy + \alpha\mu e) \geq 0.$$

Thus, we obtain that for each $0 \leq \alpha \leq 1$ the $x(\alpha)y(\alpha) \geq 0$ inequality holds, which means that the linear functions of $\alpha$, $x(\alpha)$ and $y(\alpha)$ do not change sign on the interval $[0, 1]$. Consequently, $x(0) = x > 0$ and $y(0) = y > 0$ yields $x(1) = x_+ > 0$ and $y(1) = y_+ > 0$. This completes the proof. □

**Lemma 2.3.** *Let $\delta = \delta(xy; \mu) < 1$. Then $x_+ > 0$ and $y_+ > 0$, which means that the full-Newton step is strictly feasible.*

*Proof.* Due to Lemma 2.2, $x_+ > 0$, $y_+ > 0$ if and only if $e + d_x d_y > 0$. Since

$$1 + (d_x d_y)_i \geq 1 - |(d_x d_y)_i| \geq 1 - \|d_x d_y\|_\infty \quad \forall i,$$

then by (2.8), it follows that

$$1 + (d_x d_y)_i \geq 1 - \delta^2.$$

Thus $e + d_x d_y > 0$ if $\delta < 1$ and then $x_+$ and $y_+$ are strictly feasible. This completes the proof. □

The next lemma shows the influence of the full-Newton step on the proximity measure. For convenience, we may write

$$v_+ = \sqrt{\frac{x_+ y_+}{\mu}}.$$

It is easy to see that

$$v_+^2 = e + d_x d_y \Leftrightarrow x_+ y_+ = \mu(e + d_x d_y). \tag{2.9}$$

**Lemma 2.4.** *If* $\delta < 1$. *Then* $\delta_+ := \delta(v_+;\mu) \leq \frac{\delta^2}{\sqrt{2(1-\delta^2)}}$.

*Proof.* By definition $2\delta_+ = \|v_+ - v_+^{-1}\|$. Due to (2.9), we get $v_+ = \sqrt{e + d_x d_y}$ and $v_+^{-1} = \frac{e}{\sqrt{e + d_x d_y}}$, then

$$2\delta_+ = \|\frac{d_x d_y}{\sqrt{e + d_x d_y}}\| \leq \frac{\|d_x d_y\|}{\sqrt{1 - \|d_x d_y\|_\infty}}.$$

Hence from (2.8), $\delta_+ \leq \frac{\delta^2}{\sqrt{2(1-\delta^2)}}$. This completes the proof. □

**Corollary 2.5.** *Let* $\delta \leq \frac{1}{\sqrt{2}}$, *thus* $\delta_+ \leq \delta^2$ *which means that the full-Newton step ensures the local quadratic convergence of the proximity measure during the algorithm.*

### 2.2.2 Complexity analysis of the short-step IPA

The next lemma examines what is the effect of the full-Newton step on the duality gap during the algorithm process.

**Lemma 2.6.** *Let* $\delta = \delta(xy;\mu)$ *and suppose that the vectors* $x_+$ *and* $y_+$ *are obtained using a full-Newton step, thus* $x_+ = x + \Delta x$ *and* $y_+ = y + \Delta y$. *We have*

$$(x_+)^T y_+ \leq \mu(n + 2\delta^2),$$

*and if* $\delta \leq \frac{1}{\sqrt{2}}$, *then*

$$(x_+)^T y_+ \leq 2\mu n. \tag{2.10}$$

*Proof.* Using (2.9), we obtain $(x_+)^T y_+ = \mu e^T(e + d_x^T d_y) = \mu(n + d_x^T d_y)$. Next, due to (2.7), it follows that $(x_+)^T y_+ \leq \mu(n+2\delta^2)$. Now, let $\delta \leq \frac{1}{\sqrt{2}}$ then $\delta^2 \leq \frac{1}{2}$ from which it implies that $(x_+)^T y_+ \leq \mu(n+1)$. But since $n+1 \leq 2n$, $\forall n \geq 1$, this gives the required result. □

In the next lemma we analyze the effect which a Newton step followed by an update of the parameter $\mu$ has on the proximity measure. Suppose that $\mu$ is reduced by the factor $(1-\theta)$ at each iteration.

**Lemma 2.7.** *Let* $\delta \leq \dfrac{1}{\sqrt{2}}$ *and* $\mu_+ = (1-\theta)\mu$, *where* $0 \leq \theta < 1$. *Then*

$$\delta^2(x_+y_+;\mu_+) \leq \frac{2}{15}(1-\theta) + \frac{\theta}{2} + \frac{1}{8(1-\theta)}.$$

*Moreover, if* $\theta = \frac{1}{\sqrt{2(n+1)}}$ *and* $n \geq 2$, *then* $\delta(x_+y_+;\mu_+) \leq \dfrac{1}{\sqrt{2}}$.

*Proof.* We have,

$$\begin{aligned}4\delta^2(x_+y_+;\mu_+) &= \|\sqrt{1-\theta}v_+^{-1} - \frac{1}{\sqrt{1-\theta}}v_+\|^2 \\ &= \|\sqrt{1-\theta}(v_+^{-1} - v_+) - \frac{\theta}{\sqrt{1-\theta}}v_+\|^2 \\ &= (1-\theta)\|v_+^{-1} - v_+\|^2 + \frac{\theta^2}{1-\theta}\|v_+\|^2 - 2\theta(v_+^{-1} - v_+)^T v_+ \\ &= (1-\theta)\|v_+^{-1} - v_+\|^2 + \frac{\theta^2}{1-\theta}\|v_+\|^2 - 2\theta(v_+^{-1})^T v_+ + 2\theta(v_+)^T v_+ \\ &= 4\delta_+^2(1-\theta) + \frac{\theta^2}{1-\theta}\|v_+\|^2 - 2\theta n + 2\theta\|v_+\|^2,\end{aligned}$$

because $(v_+^{-1})^T v_+ = n$ and $v_+^T v_+ = \|v_+\|^2$. Next, Lemma 2.6 (2.10), implies that

$$\|v_+\|^2 = \frac{1}{\mu}x_+^T y_+ \leq (n+1),$$

which it follows that

$$\delta^2(x_+y_+;\mu_+) \leq (1-\theta)\delta_+^2 + \frac{\theta^2(n+1)}{4(1-\theta)} + \frac{\theta}{2}.$$

Let $\delta \leq \dfrac{1}{\sqrt{2}}$ so $\delta_+^2 \leq \dfrac{2}{15}$, and $\theta = \frac{1}{\sqrt{2(n+1)}}$, then $\theta^2 = \frac{1}{2(n+1)}$, these imply that

$$\delta^2(x_+y_+;\mu_+) \leq \frac{2}{15} + \frac{11}{30}\theta + \frac{1}{8(1-\theta)}.$$

For $n \geq 2$, $\theta \in \left[0, \frac{1}{\sqrt{6}}\right]$ and consider the following function

$$f(\theta) = \frac{2}{15} + \frac{11}{30}\theta + \frac{1}{8(1-\theta)}.$$

This function is continuous and monotone increasing on $\left[0, \frac{1}{\sqrt{6}}\right]$ since

$$f'(\theta) = \frac{1}{8\,(\theta-1)^2} + \frac{11}{30} > 0.$$

Consequently,

$$f(\theta) \leq f(\tfrac{1}{\sqrt{6}}) = 0.4943 < \frac{1}{2}, \text{ for all } \theta \in \left[0, \tfrac{1}{\sqrt{6}}\right].$$

Then, after the barrier parameter is update to $\mu_+ = (1-\theta)\mu$ with $\theta = \frac{1}{\sqrt{2(n+1)}}$ and if $\delta \leq \frac{1}{\sqrt{2}}$, we obtain $\delta(x_+y_+;\mu_+) \leq \frac{1}{\sqrt{2}}$. This completes the proof. □

Lemma 2.7 indicates that **Algorithm 2.2.1** is well defined, more exactly that the conditions $x > 0$, $y > 0$ and $\delta(x_+y_+;\mu_+) \leq \frac{1}{\sqrt{2}}$ hold during the algorithm process.

In the next lemma we compute a bound for the number of iterations produced by the algorithm.

**Lemma 2.8.** *Assume that the pair $(x^0, y^0) \in \mathcal{F}^s$ with $\delta(x^0y^0;\mu_0) \leq \frac{1}{\sqrt{2}}$ for each $\mu_0 > 0$. Let $x^k$ and $y^k$ be the vectors obtained after $k$ iterations. Then the inequality $(x^k)^Ty^k \leq \epsilon$ is satisfied if $k \geq \left[\frac{1}{\theta}\log\frac{2n\mu_0}{\epsilon}\right]$.*

*Proof.* It follows from (2.10) in Lemma 2.6 that $(x^k)^Ty^k \leq 2n\mu_k = 2n(1-\theta)^k\mu_0$. In this way $(x^k)^Ty^k \leq \epsilon$ stands if $2n(1-\theta)^k\mu_0 \leq \epsilon$. We take logarithms, so we may write $k\log(1-\theta) \leq \log\epsilon - \log(2n\mu_0)$. We know that $-\log(1-\theta) \geq \theta$ for $0 \leq \theta < 1$, so the inequality holds only if $k\theta \geq \log\epsilon - \log(2n\mu_0) = \log\left(\frac{2n\mu_0}{\epsilon}\right)$. This proves the lemma. □

We end this subsection with a theorem that gives the iteration bound of the algorithm.

**Theorem 2.9.** *Using the defaults $\theta = \frac{1}{\sqrt{2(n+1)}}$ and $\tau = \frac{1}{\sqrt{2}}$ where $\mu_0 = \frac{1}{2}$, we obtain that **Algorithm 2.2.1** given in Fig. 1 requires at most $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$ iterations for getting an $\epsilon$- approximated solution of LCP.*

*Proof.* Let $\theta = \frac{1}{\sqrt{2(n+1)}}$ and $\mu_0 = \frac{1}{2}$, by using Lemma 2.8, the proof is straightforward. This completes the proof. □

## 2.3 Numerical results

For the implementation, our accuracy is set to $\epsilon = 10^{-6}$ and we use different values of the barrier parameter $\mu_0$. In addition, we compare the obtained numerical results by **Algorithm 2.2.1** with those obtained by **Algorithm 2.1** (Achache and Goutali [8]), and **Algorithm 2.3** (Achache and Tabchouche [12]) where their proposed thresholds $\tau$ and update barrier parameters $\theta$ are given, respectively, by $(\tau = \frac{1}{\sqrt{2}}, \theta = \frac{1}{2\sqrt{n}})$

and $\left(\tau = \frac{2}{\sqrt{10}},\ \theta = \sqrt{\frac{6}{23n}}\right)$.

**Problem 1.** Consider the monotone LCP where $M$ and $q$ are given by:

$$M = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{pmatrix},\ q = \begin{pmatrix} 8 \\ 6 \\ -2 \\ 6 \end{pmatrix}.$$

The strictly feasible starting point taken in the algorithm is:

$$x^0 = (0.05, 0.08, 1.79, 0.22)^T,\ y^0 = Mx^0 + q.$$

The details of obtained numerical results with different values of $\mu_0$ are summarized in Table 2.1.

Table 2.1: Numerical results for **Problem 1.**

| | 0.5 | | 0.05 | | 0.005 | | 0.0005 | |
|---|---|---|---|---|---|---|---|---|
| | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| **Algorithm 2.2.1** | 39 | 0.0154 | 33 | 0.0112 | 27 | 0.0092 | 20 | 0.0058 |
| **Algorithm 2.3** | 50 | 0.0181 | 42 | 0.0136 | 34 | 0.0114 | 26 | 0.0095 |
| **Algorithm 2.1** | 51 | 0.0164 | 43 | 0.0138 | 35 | 0.0108 | 27 | 0.0097 |

A solution of LCP is given by

$$x^\star = (0,\ 0,\ 2,\ 0)^T,\ y^\star = (10,\ 6,\ 0,\ 2)^T.$$

**Problem 2.** Consider the following monotone LCP where $M$ and $q$ are given by

$$M = \begin{pmatrix} 1 & 0 & -0.5 & 0 & 1 & 3 & 0 \\ 0 & 0.5 & 0 & 0 & 2 & 1 & -1 \\ -0.5 & 0 & 1 & 0.5 & 1 & 2 & -4 \\ 0 & 0 & 0.5 & 0.5 & 1 & -1 & 0 \\ -1 & -2 & -1 & -1 & 0 & 0 & 0 \\ -3 & -1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 & 0 & 0 \end{pmatrix},\ q = \begin{pmatrix} -1 \\ 3 \\ 1 \\ -1 \\ 5 \\ 6 \\ 1.5 \end{pmatrix}.$$

The strictly feasible starting point for this example is taken as:

$$x^0 = (0.98,\, 0.14,\, 0.31,\, 1.84,\, 0.32,\, 0.12,\, 0.17)^T,\; y^0 = Mx^0 + q.$$

The details of numerical results obtained with different relaxed barrier values of $\mu_0$ are summarized in Table 2.2.

Table 2.2: Numerical results for **Problem 2.**

| | $0.5$ | | $0.05$ | | $0.005$ | | $0.0005$ | |
|---|---|---|---|---|---|---|---|---|
| | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| **Algorithm 2.2.1** | 53 | 0.0259 | 45 | 0.0170 | 37 | 0.0150 | 29 | 0.0114 |
| **Algorithm 2.3** | 71 | 0.0330 | 60 | 0.0281 | 49 | 0.0207 | 39 | 0.0172 |
| **Algorithm 2.1** | 72 | 0.0335 | 61 | 0.0284 | 50 | 0.0209 | 39 | 0.0173 |

The solution for this problem is given by

$$x^\star = (1,\, 0,\, 0,\, 2,\, 0,\, 0,\, 0)^T,\, y^\star = (0,\, 3,\, 1.5,\, 0,\, 2,\, 5,\, 1.5)^T.$$

**Problem 3.** The matrix $M$ and the vector $q$ for the monotone LCP are given by

$$M = \begin{pmatrix} 4 & -2 & 0 & \cdots & 0 \\ -2 & 4 & -2 & \ddots & 0 \\ 0 & -2 & 4 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 4 \end{pmatrix},\; q = \begin{pmatrix} -1 \\ 1 \\ \vdots \\ 1 \\ -1 \end{pmatrix}.$$

The strictly feasible starting point taken in the algorithm is:

$$x^0 = e \in \mathbb{R}^n,\; y^0 = Mx^0 + q.$$

The obtained numerical results by three previous algorithms are summarized in Tables 2.3, 2.4 and 2.5.

Table 2.3: Numerical results for **Problem 3** by **Algorithm 2.2.1.**

| | 0.5 | | 0.05 | | 0.005 | | 0.0005 | |
|---|---|---|---|---|---|---|---|---|
| | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| 5 | 44 | 0.0077 | 37 | 0.0064 | 30 | 0.0053 | 23 | 0.0036 |
| 10 | 65 | 0.0156 | 55 | 0.0136 | 46 | 0.0105 | 36 | 0.0084 |
| 50 | 164 | 0.9144 | 142 | 0.8543 | 120 | 0.6329 | 98 | 0.5619 |
| 100 | 243 | 2.6210 | 212 | 2.3213 | 180 | 2.0481 | 149 | 1.7305 |
| 500 | 603 | 172.0450 | 531 | 154.0775 | 459 | 132.4235 | 388 | 112.8244 |
| 1000 | 887 | 1669.3372 | 785 | 1510.5483 | 683 | 1311.0767 | 581 | 1097.1375 |

Table 2.4: Numerical results for **Problem 3** by **Algorithm 2.3**

| | 0.5 | | 0.05 | | 0.005 | | 0.0005 | |
|---|---|---|---|---|---|---|---|---|
| | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| 5 | 57 | 0.0103 | 48 | 0.0083 | 40 | 0.0071 | 31 | 0.0059 |
| 10 | 88 | 0.0211 | 75 | 0.0175 | 62 | 0.0149 | 49 | 0.0109 |
| 50 | 228 | 1.2854 | 197 | 1.1388 | 166 | 0.9484 | 136 | 0.6826 |
| 100 | 339 | 3.1661 | 295 | 3.0346 | 251 | 2.7138 | 207 | 2.3356 |
| 500 | 837 | 239.1470 | 738 | 214.7906 | 638 | 196.9205 | 538 | 155.4740 |
| 1000 | 1231 | 2286.9273 | 1089 | 2025.3267 | 948 | 1761.2855 | 806 | 1515.7447 |

Table 2.5: Numerical results for **Problem 3** by **Algorithm 2.1**

| | 0.5 | | 0.05 | | 0.005 | | 0.0005 | |
|---|---|---|---|---|---|---|---|---|
| | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| 5 | 59 | 0.0111 | 50 | 0.0097 | 41 | 0.0078 | 31 | 0.0067 |
| 10 | 90 | 0.0215 | 77 | 0.0186 | 63 | 0.0159 | 50 | 0.0118 |
| 50 | 233 | 1.3874 | 201 | 1.2191 | 170 | 1.0140 | 139 | 0.8314 |
| 100 | 346 | 3.4691 | 301 | 3.1584 | 256 | 2.7953 | 211 | 2.3650 |
| 500 | 856 | 244.7315 | 754 | 221.2114 | 652 | 188.4625 | 550 | 160.8751 |
| 1000 | 1257 | 2335.6518 | 1113 | 2058.6456 | 969 | 1795.3135 | 824 | 1537.322 |

A solution of LCP is given by

$$x^\star = (0.25,\, 0, ...,\, 0,\, 0.25)^T,\; y^\star = (0,\, 0.5,\, 1,\, ...,\, 1,\, 0.5,\, 0)^T.$$

**Comments.** Through the obtained numerical results stated in tables, we see that the algorithm offers a solution for monotone LCPs under new choices of the defaults of the updating barrier parameter $\theta$ and the threshold $\tau$ and obtains better numerical results with the relaxed parameter $\mu_0 = 0.00005$ and the update barrier $\theta = \frac{1}{\sqrt{2(n+1)}}$, since the number of iterations is significantly reduced compared to the numerical results obtained by using the two other defaults. Moreover, across the numerical results obtained for the three examples of monotone LCP, confirm that **Algorithm 2.2.1** performs well in practice in comparison with **Algorithm 2.1** and **Algorithm 2.3**, since the number of iterations and the elapsed times (CPU) produced by our algorithm, are always less than those obtained by the other algorithms.

## 2.4 Conclusion

In this chapter, we presented an efficient short-step full-Newton step primal-dual interior-point algorithm for solving monotone LCP. We showed under the threshold $\tau = \frac{1}{\sqrt{2}}$ and the updating barrier parameter $\theta = \frac{1}{\sqrt{2(n+1)}}$ that this algorithm achieves the best-known polynomial complexity, namely, $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$. The numerical results obtained by our algorithm, are very encouraging.

Chapter 3

# A path-following interior-point algorithm for monotone LCP based on a modified Newton search direction

In [47], Kheirfam and Nasrollahi utilized the AETs technique based on the power function $\psi_q(t) = t^{\frac{q}{2}}, q \geq 1$, to develop a full-Newton short-step IPA for LO. They offered a family of new search directions with respect to the parameter $q$. Further, under defaults $\tau = \dfrac{2}{(q-1)^2+2}$ and $\theta = \dfrac{1}{((2q-1)^3-5q)\sqrt{n}}$ with $q \geq 3$, they obtained the following iteration bound, namely,

$$\mathcal{O}\left(\sqrt{n}\log\frac{\mu^0\left(n+\dfrac{4q(q-2)}{((q-1)^2+2)^2}\right)}{\epsilon}\right).$$

Their study includes some earlier works. For example,

- for $q = 1$, then $\psi(t) = \sqrt{t}$ and which is introduced by Darvay in [26, 29] for LO and $P_*(\kappa)$-LCP;
- for $q = 2$, then $\psi(t) = t$, this gives the classical Newton directions [67];
- for $q = 3$, then $\psi(t) = t^{\frac{3}{2}}$ which is introduced recently by Moussaoui and Achache in [61] for CQO;
- for $q = 4$, then $\psi(t) = t^2$ which is introduced by Kheirfam in [48] for LO.

For more details about the AETs technique we direct the reader to the references [26, 29, 45, 71].

Based on the iteration bound obtained by [47], we notice that if $q$ becomes very large then $\theta$ gets very small. Consequently, the rate $(1-\theta)$ which determines the decrease in the barrier parameter converges to one. This leads to a slow convergence and even to a divergence of their algorithm. So it clear that taking a member of $\psi_q(t) = t^{\frac{q}{2}}$ with a large value of $q$ leads to bad numerical results.

In this chapter, in order to improve the numerical results of these algorithms, we reconsider the analysis of their IPAs designed for LO to the monotone LCP where a non parametric univariate function, namely, $\psi_q(t) = t^{\frac{5}{2}}$ is suggested. Therefore, similar to LO, we use the AET technique introduced by this function to nonlinear equations of the system which defines the central-path of monotone LCPs, a modified nonlinear equations is obtained. The application of Newton method to the latter, a modified search direction is offered. The proposed IPA uses full-Newton steps for tracing approximately the central-path. Unlike LO case, the presence of non orthogonality of scaled directions in LCPs, a different analysis is stated. Further, under new appropriate choices of defaults $\tau$ and $\theta$, we prove that the algorithm is well-defined and converges locally quadratically to a solution of monotone LCP. Moreover, the currently best known iteration bound for the algorithm with short-update method, namely, $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$ is obtained. Some numerical results are presented to evaluate our proposed algorithm. Further, in order to improve its efficiency, some changes are imported on it. Finally, we compare the performances of our algorithm with a previously Fischer type IPAs on a set of monotone LCPs.

## 3.1 Preliminaries and the problem statement

A monotone LCP consists in finding a couple of vectors $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$y - Mx = q, xy = 0, \; x \geq 0, \; y \geq 0, \tag{LCP}$$

where $M \in \mathbb{R}^{n\times n}$ is positive semi-definite (PSD) and $q \in \mathbb{R}^n$. Throughout this chapter, we assume that the interior-point condition (IPC) holds for the monotone LCP, i.e., there exists a pair of vectors $(x^0, y^0)$ such that

$$y^0 - Mx^0 = q,\ x^0 > 0,\ y^0 > 0.$$

In this case the monotone LCP has a solution.

The main idea of path-following IPAs is to replace the equation $xy = 0$ in LCP by the parameterized equation $xy = \mu e$ where $\mu > 0$. Hence, we obtain the system of equations:

$$y - Mx = q,\ xy = \mu e,\ x, y > 0. \tag{3.1}$$

Under the IPC condition, Kojima et al. [52] shows that system (3.1) has a unique solution denoted by $(x(\mu), y(\mu))$ for each $\mu > 0$, which is called the $\mu-$center of monotone LCP. The set of $\mu-$centers is called the central-path of monotone LCP. If $\mu$ goes to zero, then the limit of central-path exists and since the limit point satisfies the complementarity condition $xy = 0$, the limit yields a solution of LCP.

## 3.2 The modified search directions for LCP

Following [2, 25], the AET technique for computing a new search direction for IPAs is based on the transformation of the centrality equation $xy = \mu e$ in (3.1) to the new equation $\psi(\dfrac{xy}{\mu}) = \psi(e)$ where $\psi : (0, +\infty) \to \mathbb{R}$ is a continuously differentiable function and invertible i.e., $\psi^{-1}$ exists. Then, (3.1) is transformed to the following system:

$$y - Mx = q,\ \psi(\tfrac{xy}{\mu}) = \psi(e),\ x, y \geq 0, \tag{3.2}$$

where $\psi$ is applied coordinate-wisely. Applying Newton's method to system (3.2) for a given strictly feasible point $(x, y)$ yields the new Newton system:

$$\begin{cases} \Delta y - M\Delta x = 0, \\ \dfrac{1}{\mu} y\psi'(\dfrac{xy}{\mu})\Delta x + \dfrac{1}{\mu} x\psi'(\dfrac{xy}{\mu})\Delta y = \psi(e) - \psi(\dfrac{xy}{\mu}), \end{cases} \tag{3.3}$$

where $\psi'$ denotes the derivative of $\psi$.

Next, to facilitate the analysis of the algorithm, we introduce the following notations:

$$v := \sqrt{\frac{xy}{\mu}},\ d_x := \frac{v\Delta x}{x},\ d_y := \frac{v\Delta y}{y},\ d := \sqrt{\frac{x}{y}}. \tag{3.4}$$

From (3.2) and (3.4), system (3.3) can be written as follows:

$$\begin{cases} d_y - \bar{M}d_x = 0, \\ d_x + d_y = p_v, \end{cases} \tag{3.5}$$

where

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

and $\bar{M} := DMD$, $D := diag(d)$. The system (3.5) determines a family of new scaled Newton search directions related to the function $\psi$.

Next, some values of the vector $p_v$ related to different choices of the function $\psi$ are stated.

| Functions $\psi(t)$ | The vector $p_v$ |
|---|---|
| $\psi(t) = t$ | $(v^{-1} - v)$, (Roos and al. [67]. Standard), |
| $\psi(t) = \sqrt{t}$ | $2(e - v)$, (Darvay [26]), |
| $\psi(t) = t - \sqrt{t}$ | $\frac{2(v - v^2)}{2v - e}$, $v > \frac{e}{2}$ (Darvay and Takács [28]), |
| $\psi(t) = \log t$ | $-2v \log v$, (Pan [62]), |
| $\psi(t) = \frac{\sqrt{t}}{2(1+\sqrt{t})}$ | $e - v^2$, (Kheirfam and Haghani [46]), |
| $\psi(t) = t^{\frac{q}{2}}, q \geq 1$ | $\frac{2}{q}(v^{1-q} - v)$ (Kheirfam and Nasrollahi [47]), |
| $\psi(t) = t^{\frac{3}{2}}$ | $\frac{2}{3}(v^{-2} - v)$ (Moussaoui and Achache [61]). |

**The value of $p_v$ for different functions $\psi$.**

Now, we consider the AET introduced by the function $\psi(t) = t^{\frac{q}{2}}$ with $q = 5$. This yields

$$p_v = \frac{2}{5}(v^{-4} - v). \tag{3.6}$$

Moreover, system (3.3) becomes

$$\begin{cases} \Delta y - M\Delta x = 0, \\ y\Delta x + x\Delta y = \dfrac{2\mu}{5}\left[\left(\dfrac{xy}{\mu}\right)^{-\frac{3}{2}} - \left(\dfrac{xy}{\mu}\right)\right]. \end{cases} \tag{3.7}$$

Next, according to (3.6), we define a norm-based proximity measure as follows:

$$\delta(v) := \delta(xy;\mu) := \frac{5}{2}\|p_v\| = \|v^{-4} - v\|. \tag{3.8}$$

Clearly, $\delta(v) = 0 \Leftrightarrow v = e \Leftrightarrow xy = \mu e$. Therefore, the value of $\delta(v)$ can be considered as a measure of the distance between the given pair $(x, y)$ and the central-path. Furthermore, we define the $\tau$-neighborhood of the central-path as follows:

$$\mathcal{N}(\tau,\mu) = \{(x,y):\ x > 0,\ y = Mx + q > 0:\ \delta(xy;\mu) \leq \tau\},$$

where $\tau$ is a threshold (default) and $\mu > 0$ is fixed.

### 3.2.1 The Algorithm

Now we are ready to describe the generic full-Newton step IPA for monotone LCP as follows. First, we use a suitable threshold value, with $0 < \tau < 5$ and we suppose that an initial point $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$ exists for certain $\mu_0 > 0$ is known. The full-Newton step between successive iterates is defined as $(x_+, y_+) = (x + \Delta x, y + \Delta y)$ where the Newton directions $\Delta x$ and $\Delta y$ are solutions for the linear system (3.7). Then it updates the parameter $\mu$ by the factor $(1 - \theta)$ with $0 < \theta < 1$, and target a new $\mu$-center and so on. This procedure is repeated until the stopping criterion $x^T y \leq \epsilon$ is satisfied for a given accuracy parameter $\epsilon$.

Therefore, the generic algorithm the full-Newton step IPA for monotone LCP is stated as follows.

**Algorithm 3.2.1** Generic full-Newton step IPA for monotone LCP

```
Initialize:
$k = 0$, $\mu^{(0)} > 0$, $(x^{(0)}, y^{(0)}) > 0$;
an accuracy parameter $\epsilon > 0$;
a barrier update parameter $0 < \theta < 1$; default $\theta = \frac{1}{35\sqrt{2n}}$;
a threshold parameter $0 < \tau < 1$; default $\tau = \frac{1}{\sqrt{2}}$;
Ensure: $(x^{(0)}, y^{(0)}) \in \mathcal{N}(\tau, \mu^{(0)})$
while $n\mu \geq \varepsilon$ do
    Compute $(\Delta x, \Delta y)$ from (3.7);
    Set $x^{(k+1)} = x^{(k)} + (\Delta x)^{(k)}$ and $y^{(k+1)} = y^{(k)} + (\Delta y)^{(k)}$;
    Set $\mu^{(k+1)} := (1-\theta)\mu^{(k)}$;
    Set $k = k + 1$;
End
```

## 3.3 Analysis of the algorithm

In this section, we will show across the new defaults of $\theta$ and $\tau$, that Algorithm 3.2.1 is well-defined and converges locally quadratically to a solution of monotone LCP. Moreover, we show its polynomial complexity. We mention that due to non orthogonality of the scaled direction, our analysis is quite different from those used in LO case.

Next, the following technical results are essentially provided in our analysis. The next lemma is similar to the one given in Lemma 2.1. Here, for the benefits of readers, we reconsider the proof according to the new proximity.

**Lemma 3.1.** *Let $\mu > 0$ and $(d_x, d_y)$ be a solution of system (3.5) with $\delta := \delta(xy; \mu)$. Then one has*

$$0 \leq d_x^T d_y \leq \frac{2}{25}\delta^2 \tag{3.9}$$

*and*

$$\|d_x d_y\|_\infty \leq \frac{\delta^2}{25}, \ \|d_x d_y\| \leq \frac{\sqrt{2}}{25}\delta^2. \tag{3.10}$$

*Proof.* For the first part of (3.9) we have, $d_x^T d_y = \frac{1}{\mu}(\Delta x)^T \Delta y = \frac{1}{\mu}(\Delta x)^T M \Delta x \geq 0$, since $M$ is PSD. The second part of it, follows trivially from the following equality

$$\frac{4}{25}\delta^2 = \|p_v\|^2 = \|d_x + d_y\|^2 = \|d_x\|^2 + \|d_y\|^2 + 2d_x^T dy \geq 2d_x^T d_y.$$

For the first claim in (3.10), since

$$d_x d_y = \frac{1}{4}((d_x + d_y)^2 - (d_x - d_y)^2),$$

and

$$\|d_x + d_y\|^2 = \|d_x - d_y\|^2 + 4d_x^T d_y,$$

but since $d_x^T d_y \geq 0$, it follows that $\|d_x - d_y\| \leq \|d_x + d_y\|$. On the other hand,

$$\begin{aligned}
\|d_x d_y\|_\infty &= \frac{1}{4}\|(d_x + d_y)^2 - (d_x - d_y)^2\|_\infty \\
&\leq \frac{1}{4}\max(\| d_x + d_y\|_\infty^2, \|d_x - d_y\|_\infty^2) \\
&\leq \frac{1}{4}\max(\|d_x + d_y\|^2, \|d_x - d_y\|^2) \leq \frac{1}{4}\|d_x + d_y\|^2 = \frac{1}{4}\|p_v\|^2 = \frac{\delta^2}{25}.
\end{aligned}$$

Therefore, $\| d_x d_y \|_\infty \leq \frac{\delta^2}{25}$. Next, to prove the last claim of (3.10), we have

$$\begin{aligned}
\|d_x d_y\|^2 = e^T(d_x d_y)^2 &= \frac{1}{16}e^T\left((d_x + d_y)^2 - (d_x - d_y)^2\right)^2 \\
&= \frac{1}{16}\left\|(d_x + d_y)^2 - (d_x - d_y)^2\right\|^2 \\
&\leq \frac{1}{16}\left(\|(d_x + d_y)^2\|^2 + \|(d_x - d_y)^2\|^2\right) \leq \frac{1}{16}\left(\|d_x + d_y\|^4 + \|d_x - d_y\|^4\right) \\
&\leq \frac{1}{8}\|d_x + d_y\|^4 = \frac{1}{8}\|p_v\|^4 = \frac{2}{5^4}\delta^4.
\end{aligned}$$

Hence, $\|d_x d_y\| \leq \frac{\sqrt{2}}{25}\delta^2$. This completes the proof. □

In the next lemma, we show the feasibility of a full-Newton step under the condition $\delta < 5$ throughout the algorithm.

**Lemma 3.2.** *Let $\delta < 5$, then $x_+ = x + \Delta x > 0$ and $y_+ = y + \Delta y > 0$, i.e., the full-Newton step is strictly feasible.*

*Proof.* For any $\alpha \in [0, 1]$ and $(x, y)$ a strictly feasible point for LCP LCP, we have

$$x(\alpha)y(\alpha) = (x + \alpha\Delta x)(y + \alpha\Delta y) = xy + \alpha(x\Delta y + y\Delta x) + \alpha^2\Delta x\Delta y.$$

Using (3.4), we get,

$$x(\alpha)y(\alpha) = \mu\left((1 - \alpha)v^2 + \alpha(v^2 + vp_v + \alpha d_x d_y)\right). \tag{3.11}$$

So $x(\alpha)y(\alpha) > 0$ if $v^2 + vp_v + \alpha d_x d_y > 0$. Due to (3.10) in Lemma 3.1 and (3.6) with $\delta < 5$, it follows that

$$\begin{aligned} v^2 + vp_v + \alpha d_x d_y &\geq v^2 + vp_v - \alpha \|d_x d_y\|_\infty e \\ &\geq v^2 + vp_v - \alpha \frac{\delta^2}{25} e \\ &= \frac{3}{5}v^2 + \frac{2}{5}v^{-3} - \alpha \frac{\delta^2}{25} e \\ &> \frac{3}{5}v^2 + \frac{2}{5}v^{-3} - e. \end{aligned}$$

Hence it is clear that $x(\alpha)y(\alpha) > 0$ if $\frac{3}{5}v^2 + \frac{2}{5}v^{-3} - e \geq 0$. Let us consider the function $g(t) = \frac{3}{5}t^2 + \frac{2}{5}t^{-3} - 1$ for $t > 0$. From $g''(t) > 0$, it follows that $g$ is strictly convex and therefore, it has a minimum at $t = 1$, since $g'(1) = 0$, so $g(t) \geq g(1) = 0, \forall t > 0$. Hence,

$$\frac{3}{5}v^2 + \frac{2}{5}v^{-3} - e \geq 0.$$

Thus $x(\alpha)y(\alpha) > 0$ for $\alpha \in [0, 1]$. Since $x$ and $y$ are positive we obtain that $x(\alpha) > 0$ and $y(\alpha) > 0$ for all $\alpha \in [0, 1]$. Then, by continuity the vectors $x(1) = x_+$ and $y(1) = y_+$ are positive, i.e., $x_+ > 0$ and $y_+ > 0$ are strictly feasible. This completes the proof. □

For convenience, we may write

$$v_+ = \sqrt{\frac{x_+ y_+}{\mu}}.$$

**Lemma 3.3.** *If* $\delta < 5$, *then* $\min v_+ \geq \frac{1}{5}\sqrt{25 - \delta^2}$.

*Proof.* From (3.11) in the proof of Lemma 3.2, setting $\alpha = 1$ and as $x(1) = x_+$, $y(1) = y_+$ and $p_v = \frac{2}{5}(v^{-4} - v)$, we get

$$v_+^2 = v^2 + vp_v + d_x d_y = \frac{3}{5}v^2 + \frac{2}{5}v^{-3} + d_x d_y.$$

By Lemma 3.2, we have seen that $\frac{3}{5}v^2 + \frac{2}{5}v^{-3} - e \geq 0$ if $\delta < 5$, hence $\frac{3}{5}v^2 + \frac{2}{5}v^{-3} \geq e$. Consequently, $v_+^2 \geq e + d_x d_y$. Next, using (3.10), we deduce that

$$v_+^2 \geq e + d_x d_y \geq (e - \|d_x d_y\|_\infty e) \geq \left(1 - \frac{\delta^2}{25}\right) e = \frac{1}{25}(25 - \delta^2)e,$$

hence $\min v_+ \geq \frac{1}{5}\sqrt{25 - \delta^2}$. This completes the proof. □

Next, we prove that the iterate across the proximity measure is locally quadratically convergent during the algorithm.

**Lemma 3.4.** *Let $(x, y)$ a strictly feasible point for LCP LCP and $\delta < 5$. Then*

$$\delta^+ := \delta(v_+) := \delta(x_+y_+;\mu) \leq \left(\frac{25}{25-\delta^2} + \frac{5^4}{(25-\delta^2)^2} + \frac{5}{5+\sqrt{25-\delta^2}}\right)\left(\frac{3}{5}+\frac{\sqrt{2}}{25}\right)\delta^2.$$

*In addition, assume $\delta \leq \frac{1}{4}$, then $\delta^+ \leq \delta^2$, which means the locally quadratically convergence of the proximity measure.*

*Proof.* We have

$$\delta^+ = \left\|v_+^{-4} - v_+\right\| = \left\|\frac{e - v_+^5}{v_+^4}\right\| = \left\|(e - v_+^2)\left(\frac{e + v_+ + v_+^2 + v_+^3 + v_+^4}{(e+v_+)v_+^4}\right)\right\|.$$

Consider the function

$$f(t) = \frac{1+t+t^2+t^3+t^4}{(1+t)t^4} = \frac{1}{t^2} + \frac{1}{t^4} + \frac{1}{1+t}.$$

Using $f$ we can write

$$\delta^+ = \|f(v_+)(e - v_+^2)\| \leq \|f(v_+)\|_\infty \|e - v_+^2\|.$$

The function $f$ is continuous and positive on $(0, +\infty)$. Moreover, from $f'(t) < 0$ for $t > 0$ it follows that $f$ is decreasing, therefore,

$$0 < |f((v_+)_i)| = f((v_+)_i) \leq f(\min v_+) \leq f\left(\frac{1}{5}\sqrt{25-\delta^2}\right).$$

Consequently

$$\|f(v_+)\|_\infty = \frac{25}{25-\delta^2} + \frac{5^4}{(25-\delta^2)^2} + \frac{5}{5+\sqrt{25-\delta^2}}.$$

This implies that

$$\delta^+ \leq \left(\frac{25}{25-\delta^2} + \frac{5^4}{(25-\delta^2)^2} + \frac{5}{5+\sqrt{25-\delta^2}}\right)\|e - v_+^2\|.$$

Next, setting $\alpha = 1$ in (3.11), and due to (3.6), we have,

$$\|e - v_+^2\| = \|e - (v^2 + vp_v + d_xd_y)\| = \|e - \frac{3}{5}v^2 - \frac{2}{5}v^{-3} - d_xd_y\|.$$

Hence

$$\|e - v_+^2\| \leq \left\| e - \frac{3}{5}v^2 - \frac{2}{5}v^{-3} \right\| + \|d_x d_y\|.$$

Next, we may write

$$\left\| e - \frac{3}{5}v^2 - \frac{2}{5}v^{-3} \right\| = \left\| \varphi(v).\frac{25}{4}p_v^2 \right\|,$$

where

$$\varphi(v) = \frac{e - \frac{3}{5}v^2 - \frac{2}{5}v^{-3}}{(v^{-4} - v)^2} = -\frac{v^5(3v^3 + 6v^2 + 4v + 2e)}{5(v^4 + v^3 + v^2 + v + e)^2}.$$

Let us consider the function

$$\varphi(t) = -\frac{t^5(3t^3 + 6t^2 + 4t + 2)}{5(t^4 + t^3 + t^2 + t + 1)^2}. \tag{3.12}$$

This function is continuous and monotonically decreasing and negative on $(0, +\infty)$, and consequently, we have

$$-\frac{3}{5} = \lim_{t \mapsto +\infty} \varphi(t) < \varphi(t) \leq 0, \, \forall t > 0.$$

This implies that

$$0 \leq |\varphi(v_i)| = -\varphi(v_i) < \frac{3}{5}, \forall i = 1, ..., n.$$

Then as $\|p_v\|^2 = \frac{4}{25}\delta^2$, $\|\varphi(v)\|_\infty = \max_i |\varphi(v_i)| \leq \frac{3}{5}$, and $\|p_v^2\| \leq \|p_v\|^2$, it yields

$$\left\| e - \frac{3}{5}v^2 - \frac{2}{5}v^{-3} \right\| \leq \|\varphi(v)\|_\infty \frac{25}{4}\|p_v\|^2 = \frac{3}{5}\delta^2.$$

Due to (3.10), $\|d_x d_y\| \leq \frac{\sqrt{2}}{25}\delta^2$, we get $\|e - v_+^2\| \leq \left(\frac{3}{5} + \frac{\sqrt{2}}{25}\right)\delta^2$. This implies that

$$\delta^+ \leq \left( \frac{25}{25 - \delta^2} + \frac{5^4}{(25 - \delta^2)^2} + \frac{5}{5 + \sqrt{25 - \delta^2}} \right) \left( \frac{3}{5} + \frac{\sqrt{2}}{25} \right) \delta^2.$$

Next, assume $\delta \leq \frac{1}{4}$, then we get $\delta^+ \leq 1.6466\delta^2$. This completes the proof. □

The next lemma shows the influence of a full-Newton step on the duality gap and gives an upper bound for it.

**Lemma 3.5.** *Let $\delta = \delta(xy; \mu)$ and suppose that the vectors $x_+$ and $y_+$ are obtained using a full-Newton step, thus $x_+ = x + \Delta x$ and $y_+ = y + \Delta y$. We have $(x_+)^T y_+ \leq \mu(n + 2\delta^2)$. In addition, if $\delta \leq \frac{1}{4}$, then*

$$(x_+)^T y_+ \leq 2\mu n. \tag{3.13}$$

*Proof.* Due to (3.4) and (3.6) we have

$$v^2 + vp_v + d_xd_y = \frac{3}{5}v^2 + \frac{2}{5}v^{-3} + d_xd_y,$$

then

$$x_+y_+ = \mu(v^2 + vp_v + d_xd_y) = \mu\left(e + \frac{25}{4}p_v^2.\frac{\frac{3}{5}v^2 + \frac{2}{5}v^{-3} - e}{(v^{-4} - v)^2} + d_xd_y\right)$$
$$\leq \mu\left(e + \frac{25}{4}p_v^2 + d_xd_y\right),$$

since after some reductions,

$$\frac{\frac{3}{5}v_i^2 + \frac{2}{5}v_i^{-3} - 1}{(v_i^{-4} - v_i)^2} = -\varphi(v_i) < \frac{3}{5} < 1, \text{for all } i$$

where $\varphi$ is defined in (15). Then from (3.8) and (3.9), we get

$$(x_+)^Ty_+ = e^T(x_+y_+) \leq \mu\left(n + \frac{25}{4}\|p_v\|^2 + \frac{2}{25}\delta^2\right) \leq \mu(n + 2\delta^2).$$

Next, let $\delta \leq \frac{1}{4}$ so $\delta^2 \leq 1$. Using this fact we deduce that $(x_+)^Ty_+ \leq \mu(n+2)$. But since $n + 2 \leq 2n, \forall n \geq 2$. This completes the proof. □

In the next theorem we analyze the effect of a full-Newton step on the proximity by updating the parameter $\mu$ by a factor $(1 - \theta)$.

**Theorem 3.6.** *Let $\delta \leq \frac{1}{4}$ and $\mu_+ = (1-\theta)\mu$, where $0 < \theta < 1$. Then*

$$\delta(x_+y_+; \mu_+) \leq \frac{5\sqrt{2n}\,\theta}{\sqrt{1-\theta}} + \delta^+.$$

*In addition, if $\delta \leq \frac{1}{4}$ and $\theta = \frac{1}{35\sqrt{2n}}$, $n \geq 2$, then $\delta(x_+y_+; \mu_+) \leq \frac{1}{4}$.*

*Proof.* As $\sqrt{\frac{x_+y_+}{\mu_+}} = \frac{1}{\sqrt{1-\theta}}v_+$ and $\delta_+ = \|v_+^{-4} - v_+\|$, we have

$$\delta(x_+y_+; \mu_+) = \left\|\left(\sqrt{\tfrac{x_+y_+}{(1-\theta)\mu}}\right)^{-4} - \sqrt{\frac{x_+y_+}{(1-\theta)\mu}}\right\| = \left\|(1-\theta)^2v_+^{-4} - \frac{1}{\sqrt{1-\theta}}v_+\right\|$$
$$= \left\|(1-\theta)^2v_+^{-4} - (1-\theta)^2v_+ + (1-\theta)^2v_+ - \frac{1}{\sqrt{1-\theta}}v_+\right\|.$$

Next, from the triangular inequality it follows that

$$\begin{aligned}
\delta(x_+y_+;\mu_+) &\leq \left\|(1-\theta)^2 v_+^{-4} - (1-\theta)^2 v_+\right\| + \left\|(1-\theta)^2 v_+ - \frac{1}{\sqrt{1-\theta}} v_+\right\| \\
&= (1-\theta)^2\delta^+ + \left|(1-\theta)^2 - \frac{1}{\sqrt{1-\theta}}\right| \|v_+\| \\
&\leq \delta^+ + \left|(1-\theta)^2 - \frac{1}{\sqrt{1-\theta}}\right| \|v_+\| \\
&= \delta^+ + \left|(1-\theta)^2 - \frac{\sqrt{1-\theta}}{1-\theta}\right| \|v_+\| \\
&= \delta^+ + \frac{1}{|1-\theta|}\left|(1-\theta)^3 - \sqrt{1-\theta}\right| \|v_+\| \\
&= \delta^+ + \left|\frac{(1-\theta)^5 - 1}{(1-\theta)^3 + \sqrt{1-\theta}}\right| \|v_+\| \\
&= \delta^+ + \left|\frac{-\theta(\theta^4 - 5\theta^3 + 10\theta^2 - 10\theta + 5)}{(1-\theta)^3 + \sqrt{1-\theta}}\right| \|v_+\|.
\end{aligned}$$

Next, as $\dfrac{1}{(1-\theta)^3 + \sqrt{1-\theta}} \leq \dfrac{1}{\sqrt{1-\theta}}$, and $0 < \theta^4 - 5\theta^3 + 10\theta^2 - 10\theta + 5 \leq 5, \forall\, \theta \in (0,1)$, and also due to (3.13) in Lemma 3.5, we have $\|v_+\| \leq \sqrt{2n}$ it follows that,

$$\delta(x_+y_+;\mu_+) \leq \delta^+ + \frac{5\sqrt{2n}\theta}{\sqrt{1-\theta}}, \forall\, \theta \in (0,1).$$

As $\delta^+ \leq 1.6466\delta^2$ and let $\delta \leq \dfrac{1}{4}$, then we get

$$\delta(x_+y_+;\mu_+) \leq 0.1029 + \frac{5\sqrt{2n}\,\theta}{\sqrt{1-\theta}}.$$

Let $\theta = \dfrac{1}{35\sqrt{2n}}$, $n \geq 2$, hence $\theta \in \left[0, \dfrac{1}{70}\right]$. Finally, we conclude that

$$\delta(x_+y_+;\mu_+) \leq h(\theta)$$

where

$$h(\theta) = \frac{1}{7\sqrt{1-\theta}} + 0.1029.$$

As $h'(\theta) > 0, \forall\theta \in \left[0, \dfrac{1}{70}\right]$, hence $h(\theta)$ is an increasing function so $h(\theta) \leq h(\dfrac{1}{70}) = 0.2468$. Finally, we deduce $\delta(x_+y_+;\mu_+) \leq \dfrac{1}{4}$. This proves the theorem. □

Theorem 3.6 shows that Algorithm 3.1 is well defined, since $(x, y) \in \mathcal{N}$ is maintained throughout the algorithm.

### 3.3.1 Iteration bound

**Lemma 3.7.** *Assume that the pair* $(x^0, y^0)$ *is strictly feasible point and* $\delta(x^0y^0; \mu_0) \leq \frac{1}{4}$ *for a fixed* $\mu_0 > 0$. *Let* $x^k$ *and* $y^k$ *be the vectors obtained after* $k$ *iterations with* $\mu := \mu^k$. *Then the inequality* $(x^k)^T y^k \leq \epsilon$ *is satisfied if* $k \geq \left\lceil \frac{1}{\theta} \log \frac{n\mu_0}{\epsilon} \right\rceil$.

*Proof.* We have $\mu^k = (1-\theta)\mu^{k-1} = (1-\theta)^k\mu_0$, then from Lemma 3.5 (3.13), $(x^k)^T y^k \leq (1-\theta)^k 2n\mu_0$. So $(x^k)^T y^k \leq \epsilon$ holds if $(1-\theta)^k 2n\mu_0 \leq \epsilon$. By taking the logarithm, we deduce that $k\log(1-\theta) \leq \log\epsilon - \log 2n\mu_0$. Next, by the inequality $-\log(1-\theta) \geq \theta$ for all $0 < \theta < 1$, we obtain $k\theta \geq \log 2n\mu_0 - \log\epsilon = \log \frac{2n\mu_0}{\epsilon}$. This proves the lemma. □

**Theorem 3.8.** *Using the defaults* $\theta = \frac{1}{35\sqrt{2n}}$, $n \geq 2$ *and* $\delta = \frac{1}{4}$ *where* $\mu_0 = \frac{1}{2}$. *Algorithm 3.2.1 requires at most* $\mathcal{O}(\sqrt{n}\log\frac{n}{\epsilon})$ *iterations for getting an* $\epsilon$*-approximate solution of LCP.*

*Proof.* Let $\theta = \frac{1}{35\sqrt{2n}}$ and $\mu_0 = \frac{1}{2}$, by using Lemma 3.7, the proof is straightforward. □

## 3.4 Numerical results

To evaluate the performance of **Algorithm 3.2.1**, we consider some examples of monotone LCP problems of different sizes, each example is followed by two tables one contains initial starting points and the other summarizes numerical results obtained by the algorithm. In our implementation, we use $\epsilon = 10^{-4}$ and $\mu_0 = \frac{(x^0)^T y^0}{n}$. In addition, we compare the obtained numerical results by **Algorithm 3.2.1** with those obtained by **Algorithm in Fig. 1** (Kheirfam and Nasrollahi [47]), where their proposed threshold $\tau$ and update barrier parameter $\theta$ if $q = 5$ are given by ($\tau = \frac{1}{9}$, $\theta = \frac{1}{704\sqrt{n}}$). A solution of the monotone LCP is denoted by $x^\star$. Finally, the "Iter" and "CPU" denote the number of iterations and the elapsed time, respectively.

<u>**Problem 1.**</u> Consider the following MLCP where $M \in \mathbb{R}^{5\times5}$ and $q \in \mathbb{R}^5$ are given by:

$$M = \begin{pmatrix} 6 & 6 & 4 & 3 & 2 \\ 8 & 21 & 14 & 10 & 12 \\ 4 & 14 & 13 & 5 & 9 \\ 4 & 10 & 5 & 6 & 5 \\ 3 & 12 & 8 & 4 & 10 \end{pmatrix}, \; q = \begin{pmatrix} -20.5 \\ -64.5 \\ -44.5 \\ -29.5 \\ -36.5 \end{pmatrix}.$$

For this example, the initial point for the algorithm is taken as

$$x^0 = (1,1,1,1,1)^T, \;\; y^0 = Mx^0 + q = (0.5, 0.5, 0.5, 0.5, 0.5)^T,$$

with the proximity $\delta(x^0 y^0, \mu_0) = 0$, i.e., $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$.

A solution of Problem 1 is:

$$x^\star = (0.6364,\ 2.3222,\ 0.5847,\ 0,\ 0.2046)^T,\ y^\star = (0,\ 0,\ 0,\ 0.2149,\ 0)^T.$$

**Problem 2.** In this example of MLCP, $M \in \mathbb{R}^{8\times 8}$ and $q \in \mathbb{R}^8$ are given by:

$$M = \begin{pmatrix} 8 & 9 & 13 & 13 & 5 & 11 & 9 & 10 \\ 8 & 10 & 15 & 15 & 7 & 12 & 10 & 12 \\ 13 & 15 & 26 & 26 & 10 & 20 & 13 & 21 \\ 13 & 15 & 26 & 26 & 10 & 20 & 12 & 20 \\ 5 & 7 & 10 & 10 & 5 & 9 & 5 & 8 \\ 11 & 12 & 20 & 20 & 9 & 19 & 13 & 15 \\ 9 & 10 & 13 & 12 & 5 & 13 & 16 & 13 \\ 10 & 12 & 21 & 20 & 8 & 15 & 13 & 22 \end{pmatrix}, \; q = \begin{pmatrix} -8.265 \\ -9.3033 \\ -14.835 \\ -14.4633 \\ -5.995 \\ -12.4133 \\ -10.015 \\ -12.3033 \end{pmatrix}.$$

The initial starting points considered for Problem 2 is

$$x^0 = (0.2233, 0.1893, 0.1207, 0.1202, 0.2758, 0.1431, 0.1961, 0.1403)^T,$$

$$y^0 = Mx^0 + q = (4.4778, 5.2820, 8.2864, 8.3217, 3.6254, 6.9875, 5.1000, 7.1288)^T,$$

with the proximity $\delta(x^0 y^0, \mu_0) = 0.0011$, i.e., $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$.

A solution of Problem 2 is:

$$x^\star = (0.1948,\ 0,\ 0.2655,\ 0,\ 0.2506,\ 0,\ 0.2221,\ 0)^T,$$

$$y^\star = (0,\ 0.2165,\ 0,\ 0.1495,\ 0,\ 0.1872,\ 0,\ 0.1178)^T.$$

**Problem 3.** Let us consider the MLCP where $M$ and $q$ are given by:

$$M = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ 2 & 5 & 6 & \cdots & 6 \\ 2 & 6 & 9 & \cdots & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 6 & 10 & \cdots & 4n-3 \end{pmatrix},\ q = -Me + e.$$

For this example, we consider the following initial starting point $x^0 = y^0 = e$, where the proximity $\delta(x^0 y^0, \mu_0) = 0$, i.e., $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$.

In Table 3.1, the obtained numerical results for Problems 1,2 and 3 are summarized.

Table 3.1: Number of iterations and CPU time for **Problems 1,2 and 3.**

| | | **Algorithm 3.2.1** | | **Kheirfam Algorithm** | |
|---|---|---|---|---|---|
| | $n$ | ITER | CPU | ITER | CPU |
| Problem 1 | 5 | 1116 | 0.5311 | 15937 | 4.9717 |
| Problem 2 | 8 | 1479 | 0.8817 | 21095 | 7.7729 |
| Problem 3 | 5 | 1193 | 0.5481 | 17027 | 5.1758 |
| | 10 | 1797 | 1.1259 | 25625 | 12.7354 |
| | 20 | 2696 | 2.4844 | 38424 | 31.6842 |
| | 30 | 3413 | 4.4300 | 48624 | 70.1209 |
| | 50 | 4587 | 13.6326 | - | - |
| | 100 | 6832 | 120.4609 | - | - |

**Comment 1.** Across the Table 3.1, we see that the number of iterations and the elapsed time produced by our algorithm by using our theoretical default $\theta = \dfrac{1}{35\sqrt{2n}}$ are too much higher. This means that the algorithm converges to a solution of LCP but very slowly. Meanwhile, for the default $\theta = \dfrac{1}{704\sqrt{n}}$ proposed by [47], the algorithm diverges when $n$ reaches the size $n \geq 50$. It is clear that our numerical results are better than those given by the algorithm in [47].

In general, the obtained numerical results by using both $\theta$ are not good. The cause is due to the fact that the defaults of $\theta$ become very small for large size problems.

This leads during the execution of algorithms that the rate of decrease $(1-\theta)$ in the sequence of barrier parameters $\{\mu_k\}$ becomes close enough to one.

### 3.4.1 A numerical amelioration for the algorithm

In this subsection, in order to ameliorate the performances of **Algorithm 3.2.1**, we make some changes, where instead of using the theoretical default of $\theta$, we select them as a constant which belongs to $(0,1)$ and we use $\epsilon = 10^{-7}$. Further, to keep iterations positive, we introduce a step-size $\alpha^k > 0$ such that $x^k + \rho\alpha^k(\Delta x)^k > 0$ and $y^k + \rho\alpha^k(\Delta y)^k > 0$ with $\alpha^k = \min\{1, \bar{\alpha}\}$ and $\rho \in (0,1)$ where $\bar{\alpha} = \min\{\alpha_x, \alpha_y\}$ and $\alpha_x$, $\alpha_y$ are given by

$$\alpha_x = \{-(x_i)^k/(\Delta x_i)^k;\ (\Delta x_i)^k < 0\};\ \ \alpha_y = \{-(y_i)^k/(\Delta y_i)^k;\ (\Delta y_i)^k < 0\}.$$

Our obtained numerical results based on those modifications are compared with Fischer Algorithm [34] and summarized in Table 3.2.

**Comment 2.** Across the Table 3.2, we see that the number of iterations and the elapsed time produced by the ameliorated algorithm when using the default $\theta$ as a constant are too much lower. Meanwhile, for the algorithm proposed by [34], the algorithm diverges when $n$ reaches the size $n > 100$. It is clear that our numerical results are better than those given by the algorithm in [34].

Table 3.2: Number of iterations and CPU time for **Problems 1,2 and 3.**

| | | **Ameliorated Algorithm** | | | | **Fischer Algorithm** | |
|---|---|---|---|---|---|---|---|
| | $\theta$ | $0.7$ | | $0.9$ | | | |
| | $n$ | ITER | CPU | ITER | CPU | ITER | CPU |
| **Problem 1** | 5 | 11 | 0.0070 | 6 | 0.0043 | 21 | 0.9959 |
| **Problem 2** | 8 | 11 | 0.0084 | 6 | 0.0055 | 24 | 1.86589 |
| **Problem 3** | 10 | 11 | 0.0092 | 6 | 0.0061 | 24 | 2.1962 |
| | 20 | 12 | 0.0150 | 6 | 0.0076 | 26 | 3.1721 |
| | 50 | 13 | 0.0626 | 7 | 0.0364 | 29 | 5.1747 |
| | 100 | 13 | 0.3405 | 7 | 0.2154 | 31 | 7.1820 |
| | 500 | 15 | 6.0800 | 8 | 3.9539 | - | - |
| | 1000 | 16 | 44.6532 | 8 | 22.9708 | - | - |

The numerical results presented in Table 3.1, while confirming the theory, are done on some very simple and artificial examples, we decided to present in the second part of this section the results obtained by running our algorithm on some selection of **LCPs** from the **NETLIB** repository [37], the obtained numerical results for this set of Problems are summarized in Table 3.3. In this case for the choice of feasible initial points we used the homogeneous and self-dual LCPs model in [75].

Table 3.3: Number of iterations and CPU time of the ameliorated Algorithm for **NETLIB** set Problems.

| | $\theta$ | 0.55 | | 0.65 | |
|---|---|---|---|---|---|
| Problem | $n$ | ITER | CPU | ITER | CPU |
| afiro | 78 | 26 | 0.5634 | 20 | 0.4593 |
| kb2 | 111 | 27 | 1.2623 | 20 | 0.9382 |
| sc50b | 128 | 27 | 1.5837 | 20 | 1.1549 |
| blend | 188 | 27 | 3.7828 | 21 | 3.1589 |
| adlittle | 194 | 27 | 3.5741 | 21 | 2.7790 |
| share2b | 258 | 28 | 5.9648 | 21 | 5.3182 |
| stocfor1 | 282 | 28 | 11.9502 | 21 | 8.2487 |
| recipe | 295 | 28 | 6.6336 | 21 | 5.0846 |
| scagr7 | 314 | 28 | 11.7109 | 21 | 8.8844 |
| share1b | 370 | 28 | 19.5827 | 21 | 23.6124 |
| grow7 | 441 | 28 | 39.6760 | 22 | 26.6245 |
| beaconfd | 468 | 28 | 31.4545 | 22 | 27.4596 |
| e226 | 695 | 29 | 114.0905 | 22 | 68.2650 |
| capri | 753 | 29 | 115.2951 | 22 | 96.8791 |
| bandm | 777 | 29 | 174.6404 | 22 | 138.2905 |
| agg | 1103 | 30 | 329.4844 | 24 | 212.2758 |

## 3.5 Conclusion and remarks

In this chapter, we presented a feasible full-Newton step IPA for monotone LCP. The proposed algorithm is based on a new modified search direction obtained by using the technique of AET induced by the function $\psi(t) = t^{\frac{5}{2}}$. The polynomial complexity of this algorithm is proved via appropriate choices of parameters (defaults), namely

$\tau = \frac{1}{4}$ and $\theta = \frac{1}{35\sqrt{2n}}$, $n \geq 2$. The evaluation of Algorithm 3.2.1, across the obtained numerical results, shows its slow convergence. Even for the version proposed by [47].

Based on our numerical amelioration the new obtained results are significantly ameliorated. It should be noted here that getting initial starting points for these algorithms which verify the theoretical conditions such that the strict feasibility and the closeness to central-path still a serious problem.

Chapter 4

# A Newton descent logarithmic barrier interior-point algorithm for monotone LCP

Among many solution methods for LCP, the barrier logarithmic interior-point approach deserves a great attention due to its numerical efficiency (see [2, 5, 19, 27, 40, 73]).

In this chapter, based on the optimization techniques, we have transformed the LCP problem into a convex quadratic optimization (CQO) problem, then an unconstrained barrier logarithmic problem is stated. The existence and the uniqueness of an optimal solution of the latter is showed and its convergence to an optimal solution of LCP is proved. For its numerical aspects, the descent direction is computed via Newton's method. Meanwhile, our contribution in this chapter is to exploit the alternative of so-called minorant and majorant functions for determining the displacement step along this direction. The obtained numerical results are promising and show the effectiveness of our new approach used.

## 4.1 Preliminaries

Throughout the chapter, we assume that the following conditions hold for the LCP.

- **(PSD)**. The matrix $M$ is positive semi-definite. In this case the LCP is called monotone.

- **(SF)**. $\mathcal{F}^s = \{x \in \mathbb{R}^n : \; x > 0, \; Mx + q > 0\} \neq \varnothing$. The set of strictly feasible solutions of LCP.

### 4.1.1 Statistical inequalities

**Lemma 4.1** (Theorem 5 in [23]). *Let $w_1, \dots, w_n$ s.t. $w_i > 0$, $\forall i = 1, \dots, n$ be a sample of size $n$, then*

$$D_1 \leq \sum_{i=1}^{n} \ln(w_i) \leq D_2$$

*where*

$$D_1 = (n-1)\ln\left(\bar{w} + \frac{\sigma_w}{\sqrt{n-1}}\right) + \ln\left(\bar{w} - \sigma_w\sqrt{n-1}\right),$$
$$D_2 = (n-1)\ln\left(\bar{w} - \frac{\sigma_w}{\sqrt{n-1}}\right) + \ln\left(\bar{w} + \sigma_w\sqrt{n-1}\right),$$

*and*

$$\bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i \quad et \quad \sigma_w = \sqrt{\frac{1}{n}\|w\|^2 - \bar{w}^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(w_i - \bar{w})^2},$$

*denote the mean value of $w$ and its deviation, respectively.*

## 4.2 The barrier problem $P_\mu$ associated to the monotone LCP

First, we have transformed LCP into a convex quadratic optimization:

$$m = \inf[g(x) = \langle x, Mx + q\rangle : \ x \geq 0, \ Mx + q \geq 0]. \qquad (CQO)$$

We denote by **Sol**$(CQO)$ the set of optimal solutions of this problem and by **Sol**$(LCP)$ the set of solutions of LCP. Clearly, $x \in$ **Sol**$(LCP)$ if and only if $m = 0$ and $x \in$ **Sol**$(CQO)$. It is known that, if **(PSD)** and **(SF)** hold, then $m = 0$, **Sol**$(CQO)$ and **Sol**$(LCP)$ coincide (e.g, see, Wright [73], Theorem 8.4 ).

We apply now the logarithmic barrier approach to $CQO$ to replace the non-negativity conditions $x \geq 0$ and $Mx + q \geq 0$ by additional logarithmic barrier terms to the objective function for $\mu > 0$ (the barrier parameter), we obtain

$$f(x, \mu) = \begin{cases} \langle x, Mx + q\rangle - \mu \sum_{i=1}^{n} \ln x_i \langle Mx + q, e_i\rangle + n\mu \ln \mu, & x \in \mathcal{F}^s, \\ +\infty, & \text{otherwise}, \end{cases}$$

where $\{e_i\}_{i=1}^n$ is the canonical basis of $\mathbb{R}^n$.

Let now introduce the function $m(\mu)$ defined by

$$m(\mu) = \inf_x \left[ f_\mu(x) := f(x, \mu) : x \in \mathcal{F}^s \right]. \tag{$P_\mu$}$$

$P_\mu$ is the barrier problem associated to $CQO$. By construction, $P_0$ is only the problem $CQO$, and we have $m = m(0) = 0$. So solving LCP is equivalent to solving the perturbed problem $P_\mu$ with $\mu$ gradually decreases to zero.

Now we focus on the theoretical study of $P_\mu$ i.e., the existence and the uniqueness of an optimal solution of $P_\mu$ and this optimal solution converges to a solution of LCP as $\mu$ tends to zero.

We have the gradient and the Hessian of the objective $f_\mu(x)$ of $P_\mu$ are given by

$$\nabla f_\mu(x) = \nabla g(x) - \mu X^{-1} e - \mu \sum_{i=1}^n \frac{M^T e_i}{\langle e_i, Mx + q \rangle}, \tag{4.1}$$

and

$$H := \nabla^2 f_\mu(x) = \nabla^2 g(x) + \mu X^{-2} + \mu \sum_{i=1}^n \frac{(M^T e_i)(M^T e_i)^T}{\langle e_i, Mx + q \rangle^2}, \tag{4.2}$$

where

$$\nabla g(x) = (M + M^T)x + q \text{ and } \nabla^2 g(x) = (M + M^T), \tag{4.3}$$

with $X^{-1} := diag\left(\frac{1}{x}\right)$ and $X^{-2} := diag\left(\frac{1}{x^2}\right)$.

**Lemma 4.2.** *Under our hypothesis the objective function $f_\mu(x)$ is strictly convex.*

*Proof.* We shall prove that the Hessian $H$ is a positive definite matrix for all $x \in \mathcal{F}^s$ and $\mu > 0$. We prove first that the matrix $(M^T e_i)(M^T e_i)^T$ is symmetric positive semidefinite for all $i$. We have for all nonzero $y \in \mathbb{R}^n$,

$$\langle (M^T e_i)(M^T e_i)^T y, y \rangle = \langle e_i e_i^T My, My \rangle = \langle e_i^T My, e_i^T My \rangle = |e_i^T My|^2 \geq 0, \, \forall i = 1, \ldots, n.$$

This implies that $(M^T e_i)(M^T e_i)^T \in \mathbb{R}^{n \times n}$ are symmetric positive semidefinite for all $i = 1, \ldots, n$. So their sum is also symmetric positive semidefinite matrix. Next, as $M$ is positive semi-definite and the diagonal matrix $\mu X^{-2}$ is positive definite with $\mu > 0$, then the matrix $H$ is positive definite too. This implies that $f_\mu(x)$ is strictly convex for each $\mu > 0$. Therefore, if any minimizer of $P_\mu$ exists, is unique. This completes the proof. □

**Theorem 4.3** (Theorem A.3. in [52]). *Let the LCP satisfy the conditions **(PSD)** and **(SF)**. Then the problem $P_\mu$ has a unique minimal solution $x(\mu)$ for every $\mu > 0$.*

**Lemma 4.4.** *Let $\mu > 0$ and $x(\mu)$ be an optimal solution of $P_\mu$, then $x^\star = \lim_{\mu \to 0} x(\mu)$ is a solution of LCP.*

*Proof.* Let $x \in \mathcal{F}^s$ and $\mu > 0$ be given. Since the function $f(x, \mu)$ is differentiable at the point $(x(\mu), \mu)$, we have,

$$\begin{aligned} g(x) &= f(x, 0) \\ &\geq f(x(\mu), \mu) + \langle x - x(\mu), \nabla_x f(x(\mu), \mu)\rangle + (0 - \mu)\nabla_\mu f(x(\mu), \mu) \\ &= f(x(\mu), \mu) - \mu \nabla_\mu f(x(\mu), \mu), \end{aligned}$$

since the point $(x(\mu), \mu)$ is an optimal solution of $P_\mu$, we have, $\nabla_x f(x(\mu), \mu) = 0$, this implies that

$$\begin{aligned} g(x) \geq g(x(\mu)) + n\mu \ln \mu - \mu \sum_{i=1}^{n} \ln(x_i)_\mu - \mu \sum_{i=1}^{n} \ln\langle e_i, Mx(\mu) + q\rangle \\ - \mu \left( n + n\ln(\mu) - \mu \sum_{i=1}^{n} \ln(x_i)_\mu - \mu \sum_{i=1}^{n} \ln\langle e_i, Mx(\mu) + q\rangle \right) \\ = g(x(\mu)) - n\mu. \end{aligned}$$

Since $x \in \mathcal{F}^s$ was arbitrary, we then have

$$g(x(\mu)) - n\mu \leq \min_{x \in \mathcal{F}^s} g(x) \leq g(x(\mu)).$$

Letting $\mu$ goes to zero and $x^\star := \lim_{\mu \to 0}(x(\mu))$, we get

$$g(x^\star) = g\left(\lim_{\mu \to 0}(x(\mu))\right) = \lim_{\mu \to 0} g(x(\mu)) = \min_{x \in \mathcal{F}^s} g(x).$$

This shows that $x^\star$ is a solution of LCP. This completes the proof. □

## 4.3 Numerical aspects of $P_\mu$

In this section we focus on the numerical solution of $P_\mu$, based on the optimality conditions which are necessary and sufficient, $x$ is an optimal of $P_\mu$ if:

$$\nabla f_\mu(x) = 0. \tag{4.4}$$

We solve (4.4) by using the Newton's method i.e., finding a vector $d^{(k)}$ at each iteration satisfying the following system:

$$H^{(k)} d^{(k)} = -\nabla f_\mu(x^{(k)}). \tag{4.5}$$

For the sake of simplicity, we drop the index $\mu$ from $x(\mu)$ and $x(\mu)^{(k)}$, and write $x$ instead of $x(\mu)$ and $x^{(k)}$ instead of $x(\mu)^{(k)}$ and $H^{(k)} \equiv H(x^{(k)})$.

The principal of our algorithm is to generate a sequence $\{x^{(k)}\}$ of interior points, i.e. $x^{(k)}$ remains in $\mathcal{F}^s$ and which converges to an approximate solution of LCP. Therefore at each iteration $k$, a displacement step $t^{(k)} > 0$ is taken such that

$$x^{(k)} + t^{(k)} d^{(k)} > 0 \ , \ M(x^{(k)} + t^{(k)} d^{(k)}) + q > 0,$$

and

$$f_\mu(x^{(k+1)}) < f_\mu(x^{(k)}).$$

### 4.3.1 The prototype algorithm

A prototype algorithm for solving $P_\mu$ is formally stated as follows.

**Algorithm 4.3.1** Prototype algorithm for solving $P_\mu$

```
Initialize $k = 0$, $\mu^{(0)} > 0$, $x^{(0)}$, $d^{(0)}$, define $f(x\mu)$,$\nabla f(x,\mu)$, $\varepsilon > 0$ and $0 < \rho < 1$;
Define $g(x) = f(x, 0)$ and $\nabla g(x)$;
Ensure: $x^{(0)} \in \mathcal{F}^s$
while $|\nabla g(x^{(k)}) d^{(k)}| > \varepsilon$ do
    Solve the system $H^{(k)} d^{(k)} = -\nabla f_\mu(x^{(k)})$;
    Compute the displacement step $t^{(k)}$;
    Set $x^{(k+1)} = x^{(k)} + t^{(k)} d^{(k)}$;
    Set $\mu^{(k+1)} := \rho\mu^{(k)}$
    Set $k = k + 1$;
End
```

Next task, is the determination of displacement step $t^{(k)}$ along the direction $d^{(k)}$ by using the strategy of minorant and majorant functions.

### 4.3.2 The determination of the displacement step

Following Crouzeix and Merikhi [23], instead of using the classical line search methods for computing the displacement step $t^{(k)}$ at each iteration $k$ which require minimizing the unidimensional function

$$\min_{t>0} f_\mu(x+td),$$

we approach the function defined by

$$\gamma(t) := \frac{1}{\mu}\left(f_\mu(x+td) - f_\mu(x)\right),$$

by simple majorant and minorant approximating functions which give at each iteration $k$ an efficient $t^{(k)}$ which occurs a significant decrease in $\gamma(t)$.

Let us notice that the function $\gamma(t)$ is definite only if $x+td > 0$ and $M(x+td)+q > 0$. Then it is sufficient to find a $t_{\max} > 0$ such that $x+td > 0$ and $M(x+td)+q > 0$ for all $t \in [0, t_{\max})$. So $t_{\max}$ which remains points interior is taken as:

$$t_{\max} = \min_i \left\{ -\frac{x_i}{d_i} \,:\, d_i < 0,\; -\frac{\langle e_i, Mx+q\rangle}{(Md)_i} \,:\, (Md)_i < 0 \right\}.$$

**Theorem 4.5.** *The function $\gamma(t)$ is well-defined on $t \in [0, t_{\max})$, and can be written as follows:*

$$\gamma(t) = \left(\frac{t^2-2t}{2}\right) a + tb - \sum_{i=1}^{n} \ln(1+tz_i) - \sum_{i=1}^{n} \ln(1+ts_i).$$

*where $z_i = \dfrac{d_i}{x_i}$ and $s_i = \dfrac{\langle e_i, Md\rangle}{\langle e_i, Mx+q, \rangle}, \forall i.$*

*Proof.* Let $x \in \mathcal{F}^s$, then we have

$$\begin{aligned}\gamma(t) &= \frac{1}{\mu}\left(f_\mu(x+td) - f_\mu(x)\right)\\ &= \frac{t}{\mu}\left[d^T(M+M^T)x + d^Tq + td^TMd - \mu\sum_{i=1}^{n}\ln\left(1+t\frac{d_i}{x_i}\right) - \mu\sum_{i=1}^{n}\ln\left(1+t\frac{\langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle}\right)\right].\end{aligned}$$

Next, according to (4.3), we obtain

$$\gamma(t) = \frac{t}{\mu}\left[d^T\nabla g(x) + t\frac{\langle\nabla^2 g(x)d, d\rangle}{2}\right] - \sum_{i=1}^{n}\ln\left(1 + t\frac{d_i}{x_i}\right) - \sum_{i=1}^{n}\ln\left(1 + t\frac{\langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle}\right).$$

Now, by (4.1), we have

$$\begin{aligned}\langle\nabla f_\mu(x), d\rangle = d^T\nabla f_\mu(x) &= d^T\nabla g(x) - \mu d^T X^{-1}e - \mu d^T\sum_{i=1}^{n}\frac{M^T e_i}{\langle e_i, Mx+q\rangle}\\ &= d^T\nabla g(x) - \mu d^T X^{-1}e - \mu\sum_{i=1}^{n}\frac{\langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle},\end{aligned}$$

from which we deduce that

$$d^T\nabla g(x) = d^T\nabla f_\mu(x) + \mu d^T X^{-1}e + \mu\sum_{i=1}^{n}\frac{\langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle}. \tag{4.6}$$

Due to the fact that the direction $d$ satisfies $\nabla^2 f_\mu(x)d = -\nabla f_\mu(x)$, then we have

$$d^T\nabla^2 f_\mu(x)d = -d^T\nabla f_\mu(x),$$

and

$$d^T\nabla g(x) = -d^T\nabla^2 f_\mu(x)d + \mu d^T X^{-1}e + \mu\sum_{i=1}^{n}\frac{\langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle}.$$

From (4.2), we have

$$\begin{aligned}d^T\nabla^2 f_\mu(x)d &= \langle\nabla^2 g(x)d, d\rangle + \mu d^T X^{-2}d + \mu\sum_{i=1}^{n}\frac{d^T(M^T e_i)(M^T e_i)^T d}{\langle e_i, Mx+q\rangle^2}\\ &= \langle\nabla^2 g(x)d, d\rangle + \mu d^T X^{-2}d + \mu\sum_{i=1}^{n}\frac{\langle e_i, Md\rangle \times \langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle^2}\\ &= \langle\nabla^2 g(x)d, d\rangle + \mu d^T X^{-2}d + \mu\sum_{i=1}^{n}\frac{\langle e_i, Md\rangle^2}{\langle e_i, Mx+q\rangle^2}.\end{aligned}$$

Letting $z_i = \dfrac{d_i}{x_i}$ and $s_i = \dfrac{\langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle}$ then

$$d^T\nabla^2 f_\mu(x)d = \langle\nabla^2 g(x)d, d\rangle + \mu\|z\|^2 + \mu\|s\|^2. \tag{4.7}$$

Substituting (4.7) into (4.6), we get

$$\begin{aligned}d^T\nabla g(x) &= -\left(\langle\nabla^2 g(x)d, d\rangle + \mu\|z\|^2 + \mu\|s\|^2\right) + \mu d^T X^{-1}e + \mu\sum_{i=1}^{n}\frac{\langle e_i, Md\rangle}{\langle e_i, Mx+q\rangle}.\\ &= -\langle\nabla^2 g(x)d, d\rangle - \mu\|z\|^2 - \mu\|s\|^2 + \mu\sum_{i=1}^{n} z_i + \mu\sum_{i=1}^{n} s_i.\end{aligned}$$

Consequently,

$$\begin{aligned}\gamma(t) &= \frac{t}{\mu}\left[d^T\nabla g(x) + t\frac{\langle \nabla^2 g(x)d, d\rangle}{2}\right] - \sum_{i=1}^{n}\ln(1+tz_i) - \sum_{i=1}^{n}\ln\left(1+ts_i\right) \\ &= \left(\frac{t^2-2t}{2}\right)a + tb - \sum_{i=1}^{n}\ln(1+tz_i) - \sum_{i=1}^{n}\ln(1+ts_i),\ \forall t\in[0,t_{\max}),\end{aligned}$$

where

$$a = \frac{1}{\mu}\langle \nabla^2 g(x)d, d\rangle,\ b = \sum_{i=1}^{n} z_i + \sum_{i=1}^{n} s_i - \|z\|^2 - \|s\|^2.$$

This completes the proof. □

The first and the second derivatives of $\gamma$ are:

$$\gamma'(t) = (t-1)a + b - \sum_{i=1}^{n}\frac{z_i}{1+tz_i} - \sum_{i=1}^{n}\frac{s_i}{1+ts_i},$$

$$\gamma''(t) = a + \sum_{i=1}^{n}\frac{z_i^2}{(1+tz_i)^2} + \sum_{i=1}^{n}\frac{s_i^2}{(1+ts_i)^2} > 0,\ \forall t\in[0,t_{\max}).$$

We remark that the function $\gamma$ satisfies the following properties:

$$\gamma(0) = 0,\ -\gamma'(0) = \gamma''(0) = (a + \|z\|^2 + \|s\|^2) > 0.$$

It is clear that $\gamma$ is strictly convex and differentiable on its domain of definition and so it reaches a unique minimum $\hat{t} > 0$ when $\gamma'(t) = 0$. It is reported that finding a solution of it, is a hard task. So it is worth to find upper and lower approximating functions easy to handle than $\gamma$.

In this chapter, we construct such approximating functions of $\gamma$ by using the strategy of minorant and majorant functions of $\gamma$. Such minorant and majorant functions $\hat{\gamma}(t)$ are constructed to be close enough to $\gamma$ and satisfying the following properties

$$\hat{\gamma}(0) = 0,\ -\hat{\gamma}'(0) = \hat{\gamma}''(0) = (a + \|z\|^2 + \|s\|^2) > 0. \tag{4.8}$$

### 4.3.3 Minorant and majorant functions

**The first minorant and majorant function**

We may define the first minorant function $\gamma_1$ by

$$\gamma_1(t) = \left(\frac{t^2-2t}{2}\right)a + tb - (n-1)\ln\left(1+t\delta\right) - \ln\left(1+t\lambda\right) - (n-1)\ln\left(1+t\kappa\right) - \ln\left(1+t\nu\right),$$

on $[0, \tilde{t}_1)$ where

$$\tilde{t}_1 = \min\left\{t_{\max},\ -\frac{1}{\delta} : \delta < 0,\ -\frac{1}{\lambda} : \lambda < 0,\ -\frac{1}{\kappa} : \kappa < 0,\ -\frac{1}{\nu} : \nu < 0,\right\},$$

where

$$\delta = \bar{z} - \frac{\sigma_z}{\sqrt{n-1}},\ \lambda = \bar{z} + \sigma_z\sqrt{n-1},\ \kappa = \bar{s} - \frac{\sigma_s}{\sqrt{n-1}},\ \nu = \bar{s} + \sigma_s\sqrt{n-1}, \tag{4.9}$$

and

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i,\ \sigma_z = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2},\ \bar{s} = \frac{1}{n}\sum_{i=1}^{n} s_i,\ \sigma_s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - \bar{s})^2}.$$

Now, we define the first majorant function $\gamma_2$ by

$$\gamma_2(t) = \left(\frac{t^2 - 2t}{2}\right) a + tb - (n-1)\ln(1 + t\beta) - \ln(1 + t\eta) - (n-1)\ln(1 + t\tau) - \ln(1 + t\omega),$$

on $t \in [0, \tilde{t}_2)$ where

$$\tilde{t}_2 = \min\left\{t_{\max},\ -\frac{1}{\beta} : \beta < 0,\ -\frac{1}{\eta} : \eta < 0,\ -\frac{1}{\tau} : \tau < 0,\ -\frac{1}{\omega} : \omega < 0\right\},$$

where

$$\beta = \bar{z} + \frac{\sigma_z}{\sqrt{n-1}},\ \ \eta = \bar{z} - \sigma_z\sqrt{n-1},\ \ \tau = \bar{s} + \frac{\sigma_s}{\sqrt{n-1}},\ \ \omega = \bar{s} - \sigma_s\sqrt{n-1}, \tag{4.10}$$

and

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i,\ \ \sigma_z = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2},\ \ \bar{s} = \frac{1}{n}\sum_{i=1}^{n} s_i,\ \ \sigma_s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - \bar{s})^2}.$$

Next lemma shows that $\gamma_1$ is a minorant function of $\gamma$ and $\gamma_2$ is a majorant function for $\gamma$, respectively.

**Lemma 4.6.** *Let $t \in [0, \hat{t}_1)$, then $\gamma_1(t) \leq \gamma(t) \leq \gamma_2(t)$ for all $t \in [0, \hat{t}_1)$, where $\hat{t}_1 = \min\{\tilde{t}_1, \tilde{t}_2\}$.*

*Proof.* From the right hand of the inequality in Lemma 4.1, we have

$$-D_2 \leq -\sum_{i=1}^{n} \ln w_i.$$

Letting first, $w_i = 1 + tz_i$, so $\bar{w} = 1 + t\bar{z}$, $\sigma_w = t\sigma_z$, these imply on one hand that

$$-(n-1)\ln(1+t\delta) - \ln(1+t\lambda) \leq -\sum_{i=1}^{n} \ln(1+tz_i).$$

On the other hand if we set $w_i = 1 + ts_i$, so $\bar{w} = 1 + t\bar{s}$, $\sigma_w = t\sigma_s$, these imply that

$$-(n-1)\ln(1+t\kappa) - \ln(1+t\nu) \leq -\sum_{i=1}^{n} \ln(1+ts_i),$$

where $\delta, \lambda, \kappa$ and $\nu$ are defined in (4.9). By summation of these two inequalities, we get,

$$-(n-1)\ln(1+t\delta)-\ln(1+t\lambda)-(n-1)\ln(1+t\kappa)-\ln(1+t\nu) \leq -\sum_{i=1}^{n} \ln(1+tz_i)-\sum_{i=1}^{n} \ln(1+ts_i).$$

Now, adding the term $\left[\left(\frac{t^2-2t}{2}\right)a + tb\right]$ for both sides to the previous inequality, we obtain

$$\gamma_1(t) = \left(\frac{t^2-2t}{2}\right)a+tb-(n-1)\ln(1+t\delta)-\ln(1+t\lambda)-(n-1)\ln(1+t\kappa)-\ln(1+t\nu) \leq \gamma(t).$$

This shows that $\gamma_1(t) \leq \gamma(t)$ for all $t \in [0, \tilde{t}_1)$.

For the proof of the first majorant function we have omitted it since it is similar to the one used for the first minorant function, where we use the inequality

$$-\sum_{i=1}^{n} \ln w_i \leq -D_1,$$

in Lemma 4.1. Then, it is easy to deduce that $\gamma(t) \leq \gamma_2(t)$ for all $t \in [0, \tilde{t}_2)$.

In addition, we have

$$\gamma_1'(t) = (t-1)a + b - \frac{(n-1)\delta}{1+t\delta} - \frac{\lambda}{1+t\lambda} - \frac{(n-1)\kappa}{1+t\kappa} - \frac{\nu}{1+t\nu},$$

$$\gamma_1''(t) = a + b + \frac{(n-1)\delta^2}{(1+t\delta)^2} + \frac{\lambda^2}{(1+t\lambda)^2} + \frac{(n-1)\kappa^2}{(1+t\kappa)^2} + \frac{\nu^2}{(1+t\nu)^2} > 0,$$

$$\gamma_2'(t) = (t-1)a + b - \frac{(n-1)\beta}{1+t\beta} - \frac{\eta}{1+t\eta} - \frac{(n-1)\tau}{1+t\tau} - \frac{\omega}{1+t\omega},$$

and

$$\gamma_2''(t) = a + \frac{(n-1)\beta^2}{(1+t\beta)^2} + \frac{\eta^2}{(1+t\eta)^2} + \frac{(n-1)\tau^2}{(1+t\tau)^2} + \frac{\omega^2}{(1+t\omega)^2} > 0,$$

Further, the properties in (4.8) are satisfied by $\gamma_1$ and $\gamma_2$. Also it is clear that

$$\min_t \gamma_1(t) \leq \min_t \gamma(t) \leq \min_t \gamma_2(t) \text{ for } t \in [0, \hat{t}_1).$$

This completes the proof. □

### The second minorant and majorant function

We may define the second minorant function $\gamma_3$ by

$$\gamma_3(t) = \left(\frac{t^2 - 2t}{2}\right) a + t\left(\|z\| + \|s\| - \|z\|^2 - \|s\|^2\right) + \ln(1 - t\|z\|) + \ln(1 - t\|s\|),$$

on $[0, \tilde{t_3})$, and the second majorant function $\gamma_4$ by

$$\gamma_4(t) = \left(\frac{t^2 - 2t}{2}\right) a - t\left(\|z\| + \|s\| + \|z\|^2 + \|s\|^2\right) - \ln(1 - t\|z\|) - \ln(1 - t\|s\|),$$

on $[0, \tilde{t_3})$ where

$$\tilde{t_3} = \max\{t > 0 : 1 - t\|z\| > 0,\ 1 - t\|s\| > 0\}.$$

Next, to show that $\gamma_3$ is a minorant function of $\gamma$ and $\gamma_4$ is a majorant function of $\gamma$, respectively, for all $t \in [0, \tilde{t_3})$, we need to prove the following technical lemma.

**Lemma 4.7.** *For all $t \in \mathbb{R}$ and $z, s \in \mathbb{R}^n$ we have*

$$t\left(\sum_{i=1}^{n} z_i - \|z\| + \sum_{i=1}^{n} s_i - \|s\|\right) - \sum_{i=1}^{n} \ln(1+tz_i) - \ln(1-t\|z\|) - \sum_{i=1}^{n} \ln(1+ts_i) - \ln(1-t\|s\|) \geq 0,$$

*and*

$$t\left(\sum_{i=1}^{n} z_i + \|z\| + \sum_{i=1}^{n} s_i + \|s\|\right) - \sum_{i=1}^{n} \ln(1+tz_i) + \ln(1-t\|z\|) - \sum_{i=1}^{n} \ln(1+ts_i) + \ln(1-t\|s\|) \leq 0.$$

*Proof.* Let

$$h_1(t) = t\left(\sum_{i=1}^{n} z_i - \|z\| + \sum_{i=1}^{n} s_i - \|s\|\right) - \sum_{i=1}^{n} \ln(1+tz_i) - \ln(1-t\|z\|) - \sum_{i=1}^{n} \ln(1+ts_i) - \ln(1-t\|s\|),$$

and

$$h_2(t) = t\left(\sum_{i=1}^{n} z_i + \|z\| + \sum_{i=1}^{n} s_i + \|s\|\right) - \sum_{i=1}^{n} \ln(1+tz_i) + \ln(1-t\|z\|) - \sum_{i=1}^{n} \ln(1+ts_i) + \ln(1-t\|s\|),$$

then

$$\begin{aligned}
h_1'(t) &= \sum_{i=1}^{n} z_i - \|z\| + \sum_{i=1}^{n} s_i - \|s\| - \sum_{i=1}^{n} \frac{z_i}{1+tz_i} + \frac{\|z\|}{1-t\|z\|} - \sum_{i=1}^{n} \frac{s_i}{1+ts_i} + \frac{\|s\|}{1-t\|s\|} \\
&= \sum_{i=1}^{n} z_i \left(1 - \frac{1}{1+tz_i}\right) + \|z\| \left(\frac{1}{1-t\|z\|} - 1\right) + \sum_{i=1}^{n} s_i \left(1 - \frac{1}{1+ts_i}\right) + \|s\| \left(\frac{1}{1-t\|s\|} - 1\right) \\
&= \sum_{i=1}^{n} z_i \left(\frac{tz_i}{1+tz_i}\right) + \|z\| \left(\frac{t\|z\|}{1-t\|z\|}\right) + \sum_{i=1}^{n} s_i \left(\frac{ts_i}{1+ts_i}\right) + \|s\| \left(\frac{t\|s\|}{1-t\|s\|}\right) \\
&= \sum_{i=1}^{n} \frac{tz_i^2}{1+tz_i} + \frac{t\|z\|^2}{1-t\|z\|} + \sum_{i=1}^{n} \frac{ts_i^2}{1+ts_i} + \frac{t\|s\|^2}{1-t\|s\|} \\
&= t\sum_{i=1}^{n} z_i^2 \left(\frac{1}{1+tz_i} + \frac{1}{1-t\|z\|}\right) + t\sum_{i=1}^{n} s_i^2 \left(\frac{1}{1+ts_i} + \frac{1}{1-t\|s\|}\right),
\end{aligned}$$

we have $|z_i| \leq \|z\|$ and $|s_i| \leq \|s\|$ for all $i$, which gives $-\|z\| \leq z_i \leq \|z\|$ and $-\|s\| \leq s_i \leq \|s\|, \forall i$. Then

$$\frac{1}{1+t\|z\|} \leq \frac{1}{1+tz_i} \leq \frac{1}{1-t\|z\|} \text{ and } \frac{1}{1+t\|s\|} \leq \frac{1}{1+ts_i} \leq \frac{1}{1-t\|s\|}.$$

This implies that

$$h_1'(t) \geq t\sum_{i=1}^{n} z_i^2 \left(\frac{1}{1-t^2\|z\|^2}\right) + t\sum_{i=1}^{n} s_i^2 \left(\frac{1}{1-t^2\|s\|^2}\right) \geq 0,$$

which implies that the function $h_1$ is increasing and as $h_1(0) = 0$, then $h_1(t) \geq 0$, $\forall t$ in its domain of definition. By the same argument we prove that $h_2(t)$ is decreasing and $h_2(0) = 0$ so $h_2(t) \leq 0$, $\forall t$. This completes the proof. □

By the first inequality in Lemma 4.7, we have

$$t(\|z\|+\|s\|)+\ln(1-t\|z\|)+\ln(1-t\|s\|) \leq -\sum_{i=1}^{n} \ln(1+tz_i) - \sum_{i=1}^{n} \ln(1+ts_i) + t(\sum_{i=1}^{n} z_i + \sum_{i=1}^{n} s_i).$$

Now, adding the term $\left[\left(\frac{t^2-2t}{2}\right) a - t\left(\|z\|^2 + \|s\|^2\right)\right]$ for both sides to the previous inequality, we obtain $\gamma_3(t) \leq \gamma(t)$ for all $t \in [0, \tilde{t}_3)$.

The proof of the second majorant function is similar to the one used for $\gamma_3(t)$, where we use the second inequality in Lemma 4.7. Then it is easy to prove that $\gamma(t) \leq \gamma_4(t)$ for all $t \in [0, \tilde{t}_3)$. We mention that these two functions have the same domain of definition.

In addition, we have

$$\gamma_3'(t) = (t-1)a + \left(\|z\| + \|s\| - \|z\|^2 - \|s\|^2\right) - \frac{\|z\|}{1-t\|z\|} - \frac{\|s\|}{1-t\|s\|},$$

$$\gamma_3''(t) = a + \frac{\|z\|^2}{(1-t\|z\|)^2} + \frac{\|s\|^2}{(1-t\|s\|)^2} > 0,$$

$$\gamma_4'(t) = (t-1)a - \left(\|z\| + \|s\| + \|z\|^2 + \|s\|^2\right) + \frac{\|z\|}{1-t\|z\|} + \frac{\|s\|}{1-t\|s\|},$$

and

$$\gamma_4''(t) = a + \frac{\|z\|^2}{(1-t\|z\|)^2} + \frac{\|s\|^2}{(1-t\|s\|)^2} > 0.$$

Further, the properties in (4.8) are satisfied by $\gamma_3$ and $\gamma_4$. Also it is clear that

$$\min_t \gamma_3(t) \leq \min_t \gamma(t) \leq \min_t \gamma_4(t) \text{ for } t \in [0, \tilde{t}_3).$$

Finally, we conclude that the computation of a displacement step $t^{(k)}$ at each iteration $k$, is based on the optimal minimum of the functions $\gamma_i(t), i = 1, \ldots, 4$. It is clear that these functions are definite and strictly convex on their domains $[0, \tilde{t}_i)$, $i = 1, \ldots, 4$ and therefore they reach their minimum when

$$\gamma_i'(t) = 0, \; i = 1, \ldots, 4. \tag{4.11}$$

Hence, in the numerical implementation of our algorithm, we take only the root which belongs to the domain of each function.

We end this section by the convergence result of the algorithm.

**Lemma 4.8.** *Let $(x^{(k+1)})$ and $(x^{(k)})$ be two successive strictly feasible solutions of $P_\mu$ obtained at the iteration $k+1$ and $k$, respectively, then $f_\mu(x^{(k+1)}) < f_\mu(x^{(k)})$.*

*Proof.* We have

$$f_\mu(x^{(k+1)}) \simeq f_\mu(x^{(k)}) + \langle \nabla f_\mu(x^{(k)}), x^{(k+1)} - x^{(k)} \rangle = f_\mu(x^{(k)}) + t^{(k)}(d^{(k)})^T \nabla f_\mu(x^{(k)}).$$

Due to (4.5), we deduce that

$$\langle \nabla f_\mu(x^{(k)}), d^{(k)} \rangle = \langle -\nabla^2 f_\mu(x^{(k)}) d^{(k)}, d^{(k)} \rangle \text{ and } x^{(k+1)} = x^{(k)} + t^{(k)} d^{(k)},$$

it follows that

$$f_\mu(x^{(k+1)}) - f_\mu(x^{(k)}) \simeq t^{(k)}\langle \nabla f_\mu(x^{(k)}), d^{(k)}\rangle = -t^{(k)}\langle \nabla^2 f_\mu(x^{(k)})d^{(k)}, d^{(k)}\rangle < 0.$$

Therefore $f_\mu(x^{(k+1)}) < f_\mu(x^{(k)})$. This completes the proof. □

## 4.4 Numerical results

In this section, we present some numerical results to show the efficiency of our algorithm applied on some examples taken from the literature. In addition, we compare the obtained numerical results with the classical Wolfe and Powell line search. In the implementation, our accuracy is set to $\epsilon = 10^{-5}$. Also in view of the influence of the barrier parameter $\mu$ on the algorithm, different values of them are used in order to improve its performances.

**Problem 1.** The matrix $M$ and the vector $q$ for the MLCP are given by

$$M = \begin{pmatrix} 3 & -2 & -1 \\ -2 & 2 & 1 \\ -1 & 1 & 1 \end{pmatrix},\ q = (14, -11, -7)^T.$$

The initial point taken in the algorithm is $x^1 = (0.1646, 4.2045, 3.1691)^T$. A solution of LCP is given by

$$x^\star = (0, 4, 3)^T.$$

The details of obtained numerical results for Problem 1 are summarized in Table 4.1.

Table 4.1: Number of iterations and CPU time for **Problem 1**, where $\mu = 0.5$

| MIN1 | | MAJ1 | | MIN2 | | MAJ2 | | Wolfe | |
|---|---|---|---|---|---|---|---|---|---|
| ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| 2 | 0.0015 | 4 | 0.0049 | 5 | 0.0063 | 4 | 0.0046 | 7 | 0.0372 |

**Problem 2.** Consider the following MLCP where $M$ and $q$ are given by

$$M = \begin{pmatrix} 2 & 1 & 0 & 3 & -3 \\ 1 & 4 & 0 & 4 & 2 \\ 0 & 0 & 6 & -2 & 1 \\ -3 & -4 & 2 & 0 & 0 \\ 3 & -2 & -1 & 0 & 0 \end{pmatrix}, \; q = (1,\, -2,\, 4,\, 10,\, 2)^T.$$

The initial point for this Problem is taken as $x^0 = e \in \mathbb{R}^5$. A solution of the LCP is given by

$$x^\star = (0,\, 0.5,\, 0,\, 0,\, 0)^T.$$

The details of obtained numerical results for Problem 2 are summarized in Table 4.2.

Table 4.2: Number of iterations and CPU time for **Problem 2**, where $\mu = 0.5$

| MIN1 | | MAJ1 | | MIN2 | | MAJ2 | | Wolfe | |
|---|---|---|---|---|---|---|---|---|---|
| ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| 8 | 0.0087 | 8 | 0.0083 | 9 | 0.0118 | 7 | 0.0076 | 12 | 0.0924 |

**Example 3.** Consider the MLCP where $M$ and $q$ are given by:

$$M = (m_{ij}) = \begin{cases} 4 & \text{if } \; i = j, \\ -1 & \text{if } \; |i-j| = 1, \\ 0 & \text{otherwise,} \end{cases} \qquad q = (-1,\, 1,\, \ldots,\, 1,\, -1)^T \in \mathbb{R}^n.$$

The initial point taken for this Problem is $x^0 = e \in \mathbb{R}^n$. The details of obtained numerical results for different size of $n$ are showed in Table 4.3.

Table 4.3: Number of iterations and CPU time for **Problem 3**, where $\mu = 0.4$

| | MIN1 | | MAJ1 | | MIN2 | | MAJ2 | | Wolfe | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| 100 | 36 | 0.3934 | 32 | 0.1273 | 98 | 0.6274 | 92 | 0.5104 | 105 | 0.6408 |
| 200 | 39 | 0.9060 | 34 | 0.6109 | 141 | 1.1924 | 124 | 1.1259 | 135 | 1.8736 |
| 500 | 45 | 9.6555 | 42 | 5.2072 | 229 | 34.9834 | 203 | 21.2705 | 242 | 37.1834 |
| 1000 | 72 | 26.2486 | 67 | 23.4821 | 253 | 64.3724 | 244 | 56.2613 | 256 | 59.0025 |

## 4.5 Conclusion

In this chapter, we presented a logarithmic barrier method for solving monotone LCP problems. We have transformed the monotone LCP into an equivalent optimization problem. We have shown the existence and the uniqueness of the optimal solution of the latter problem and its convergence to a optimal solution of the LCP problem. We apply Newton's method to determine the descent direction. The novel strategy of minorant and majorant approximating functions is used for computing the displacement step in this algorithm. The numerical results are better more those given by Wolfe's and Powell line search.

Chapter 5

# *Complexity analysis of an interior-point algorithm for $P_\star(\kappa)$ nonlinear complementarity problems*

In this chapter, we establish the polynomial complexity of a feasible short-step path-following interior-point algorithm (IPA) for solving the class of nonlinear complementarity problems with $P_\star(\kappa)$-mapping (abbreviated $P_\star(\kappa)$-NCP). The proposed algorithm uses only full Newton steps with the advantage that no line search is required. We prove that the algorithm is well defined and converges locally quadratically to a solution of NCP. Moreover, we show that the algorithm has $\mathcal{O}\left(\sqrt{n}(1+4\kappa)\log(\frac{n}{\epsilon})\right)$ as a complexity result for short-update methods. Some numerical results are provided to show the efficiency of our algorithm.

## 5.1 Preliminaries

Recall again from **Subsection 1.1**, that the NCP consists in finding vectors $x, y$ in $\mathbb{R}^n$ such that

$$y = F(x),\ x^T y = 0,\ x \geq 0,\ y \geq 0, \tag{5.1}$$

where $F(x) : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function, the expression $x^T F(x)$ denotes the usual inner product of the two vectors $x$ and $F(x)$ in $\mathbb{R}^n$, and the inequalities are understood to be component-wise. The complementarity condition $x^T F(x)$ means that $x_i^T F_i(x) = 0$ for all $i = 1, \ldots, n$. Recall that $F$ is said to be a $P_\star(\kappa)-$mapping if there exist a scalar $\kappa \geq 0$ such that

$$(1+4\kappa)\sum_{i\in\mathcal{I}_+(x,y)}(x_i-y_i)(F_i(x)-F_i(y))+\sum_{i\in\mathcal{I}_-(x,y)}(x_i-y_i)(F_i(x)-F_i(y))\geq 0,$$

for any $x\neq y$ in $\mathbb{R}^n$, where

$\mathcal{I}_+(x,y)=\{i:(x_i-y_i)(F_i(x)-F_i(y))\geq 0\}$ and $\mathcal{I}_-(x,y)=\{i:(x_i-y_i)(F_i(x)-F_i(y))<0\}$.

**Lemma 5.1** (Proposition 4.2.10 in [64]). *If F is is a continuously differentiable $P_\star(\kappa)$ mapping on $\mathbb{R}^n$, then for each $x\in\mathbb{R}^n$, the Jacobian $\nabla F(x)$ is a $P_\star(\kappa)$ matrix.*

**Lemma 5.2** (Lemma 4.1 in [53]). *If $M\in\mathbb{R}^{n\times n}$ is a $P_\star(\kappa)-$ matrix, then*

$$\begin{pmatrix}-M & I\\ Y & X\end{pmatrix},$$

*is a nonsingular matrix for any positive definite diagonal matrices $X,Y\in\mathbb{R}^{n\times n}$.*

### 5.1.1 The central path of NCP

Throughout this chapter we assume that $F$ is a continuously differentiable $P_\star(\kappa)-$mapping, $\kappa\geq 0$, problem (5.1) is called a $P_\star(\kappa)-$NCP, and satisfies the interior-point condition (IPC), i.e., there exists $(x^0,y^0)\in\mathcal{F}^s$ which implies the existence of a solution to $P_\star(\kappa)-$NCP. The basic idea of path-following IPA is to relax the complementarity condition in (5.1) to the following parameterized system: for $\mu>0$,

$$\begin{cases} y=F(x)\\ xy=\mu e,\ (x,y)\geq 0.\end{cases}\tag{5.2}$$

By Theorem 4.4 in [53], the system (5.2) has a unique solution denoted by $(x(\mu),y(\mu))$ for each $\mu>0$, which is called the $\mu$-center of NCP. The set of $\mu$-centers is called the central-path. If $\mu$ goes to zero then the limit of the central-path exists and converges to a solution of NCP.

### 5.1.2 The Newton search direction

Next, we want to define search directions $(\Delta x,\Delta y)$ that move in direction of the $\mu-$centers $(x(\mu),y(\mu))$. Applying Newton's method to (5.2) for a given $(x,y)\in\mathcal{F}^s$, we get the following linear system:

$$\begin{cases} \Delta y = \nabla F(x)\Delta x, \\ x\Delta y + y\Delta x = \mu e - xy. \end{cases} \tag{5.3}$$

By our assumption and Lemma 5.1, $\nabla F(x)$ is a $P_\star(\kappa)$-matrix. By Lemma 5.2, the system (5.3) has a unique search direction vector $(\Delta x, \Delta y)$ for each $\mu > 0$. The update iterate with a full-Newton step is taken as: $x_+ = x + \Delta x$ and $y_+ = x + \Delta y$.

Now, for simplicity, let us define the vector $v := \sqrt{\frac{xy}{\mu}}$ and the scaled search directions as follows:

$$d_x := \frac{v\Delta x}{x},\ d_y := \frac{v\Delta y}{y}. \tag{5.4}$$

Then we get the following modified Newton system:

$$\begin{cases} d_y = \overline{\nabla F(x)} d_x, \\ d_x + d_y = p_v, \end{cases} \tag{5.5}$$

where $\overline{\nabla F(x)} = \mu VY^{-1}\nabla F(x)VY^{-1}$ and $p_v = v^{-1} - v$.

### 5.1.3 The proximity measure

For the analysis of the algorithm, we use a norm-based proximity measure $\delta(v)$ defined by:

$$\delta := \delta(xy; \mu) = \frac{\|p_v\|}{2}. \tag{5.6}$$

Clearly, $\delta(v) = 0 \Leftrightarrow v = e \Leftrightarrow xy = \mu e$. Hence, the value of $\delta(v)$ can be considered as a measure for the distance between the given pair $(x, y)$ and the corresponding $\mu-$center $(x(\mu), y(\mu))$.

Furthermore, we define the $\tau$-neighborhood of the central path as follows:

$$\mathcal{N}(\tau, \mu) = \{(x, y) :\ x > 0,\ y = F(x) > 0 :\ \delta(xy; \mu) \leq \tau\},$$

where $\tau$ is a threshold parameter and $\mu > 0$ is fixed.

### 5.1.4 The algorithm

The generic path-following full-Newton step IPA for NCP is described in as follows.

**Algorithm 5.1.1** Full-Newton step IPA for NCP

```
1: Initialize:
2: $k = 0$, $\mu > 0$, $(x^{(0)}, y^{(0)}) > 0$
3: an accuracy parameter $\epsilon > 0$
4: a barrier update parameter $0 < \theta < 1$; default $\theta = \frac{1}{\sqrt{2(n+1)(1+4\kappa)}}$; $\kappa \geq 0$;
5: a threshold parameter $0 < \tau < 1$; default $\tau = \frac{1}{\sqrt{2}(1+4\kappa)}$;
Ensure: $(x^{(0)}, y^{(0)}) \in \mathcal{N}(\tau, \mu)$;
6: while $n\mu \geq \varepsilon$ do
7:     Compute $(\Delta x, \Delta y)$ from (5.3);
8:     Set $x^{(k+1)} = x^{(k)} + (\Delta x)^{(k)}$ and $y^{(k+1)} = y^{(k)} + (\Delta y)^{(k)}$;
9:     Set $\mu := (1 - \theta)\mu$;
10:    Set $k = k + 1$;
11: End
```

## 5.2 Complexity analysis

In this section, we establish under our new defaults $\theta = \frac{1}{\sqrt{2(n+1)(1+4\kappa)}}$ and $\tau = \frac{1}{\sqrt{2}(1+4\kappa)}$) the polynomial complexity of Algorithm 5.1.1.

Before doing so, the following lemma is crucial for the analysis of the algorithm. For a detailed proof, see ([10], Lemma 3.1).

**Lemma 5.3.** *Let $\delta = \delta(xy; \mu) > 0$ where $\mu > 0$ and $(d_x, d_y)$ be the unique solution of system (5.5). Then one has*

$$-4\kappa\delta^2 \leq d_x^T d_y \leq \delta^2 \tag{5.7}$$

*and*

$$\|d_x d_y\|_\infty \leq (1 + 4\kappa)\delta^2, \qquad \|d_x d_y\| \leq (\sqrt{2} + 4\kappa)\delta^2. \tag{5.8}$$

**Lemma 5.4.** *If $\delta < \dfrac{1}{\sqrt{1+4\kappa}}$, then $x_+ > 0$, $y_+ > 0$, i.e., $x_+$ and $y_+$ are strictly feasible.*

*Proof.* Let $(x, y) \in \mathcal{F}^s$ i.e., $y = F(x)$ with $x > 0$ and $y > 0$. Let $\alpha \in [0, 1]$, $x(\alpha) = x + \alpha\Delta x$, $y(\alpha) = F(x(\alpha))$. Now,for each $\alpha > 0$, we define the function $\Delta y(\alpha) = \dfrac{1}{\alpha}(F(x + \alpha\Delta x) - F(x)$, this implies that $y(\alpha) = F(x + \alpha\Delta x) = y + \alpha\Delta y(\alpha)$. Note that

$\Delta y(\alpha)$ is not defined at $\alpha = 0$. For $\alpha = 0$ we define this function as follows

$$\Delta y(0) = \lim_{\alpha \mapsto 0} \Delta y(\alpha) = \nabla F(x)\Delta x = \Delta y.$$

Then, we have $y(\alpha) = y + \alpha\Delta y$ and consequently,

$$x(\alpha)y(\alpha) = (1-\alpha)xy + \alpha(\mu e + \alpha\Delta x\Delta y),$$

hence, $x(\alpha)y(\alpha) > 0$, if $\mu e + \alpha\Delta x\Delta y > 0$, or

$$\mu\left(e + \alpha\frac{\Delta x\Delta y}{\mu}\right) = \mu\left(e + \alpha d_x d_y\right),$$

$$\mu e + \alpha\Delta x\Delta y > 0 \iff e + \alpha d_x d_y > 0,$$

we have

$$e + \alpha d_x d_y \geq e - \alpha\|d_x d_y\|_\infty e > (1-(1+4\kappa)\delta^2)e.$$

As $\delta < \frac{1}{\sqrt{1+4\kappa}}$ it follows that $e + \alpha d_x d_y > 0$. Thus, we obtain that for each $0 \leq \alpha \leq 1$ the $x(\alpha)y(\alpha) > 0$ inequality holds, which means that the linear functions of $\alpha$, $x(\alpha)$ and $y(\alpha)$ do not change sign on the interval $[0,1]$. Consequently, $x(0) = x > 0$ and $y(0) = y > 0$ yields $x(1) = x_+ > 0$ and $y(1) = F(x_+) = y_+ > 0$. This completes the proof. □

The next lemma shows the influence of full-Newton step on the proximity measure. For convenience, we may write $v_+ = \sqrt{\frac{x_+y_+}{\mu}}$. It is easy to see that

$$v_+^2 = e + d_x d_y \Leftrightarrow x_+y_+ = \mu(e + d_x d_y). \tag{5.9}$$

**Lemma 5.5.** *If $\delta < \dfrac{1}{\sqrt{1+4\kappa}}$, then $\delta_+ := \delta(x_+y_+;\mu) \leq \dfrac{(\sqrt{2}+4\kappa)\delta^2}{2\sqrt{1-(1+4\kappa)\delta^2}}$.*

*Proof.* By definition $2\delta_+ = \|v_+ - v_+^{-1}\|$. Due to (5.9), we get $v_+ = \sqrt{e + d_x d_y}$ and $v_+^{-1} = \dfrac{e}{\sqrt{e + d_x d_y}}$, then

$$2\delta_+ = \|\frac{d_x d_y}{\sqrt{e + d_x d_y}}\| \leq \frac{\|d_x d_y\|}{\sqrt{1 - \|d_x d_y\|_\infty}}.$$

Hence from (5.8), $\delta_+ \leq \dfrac{(\sqrt{2}+4\kappa)\delta^2}{2\sqrt{1-(1+4\kappa)\delta^2}}$. This completes the proof. □

**Corollary 5.6.** *If $\delta \leq \dfrac{1}{\sqrt{2}(1+4\kappa)}$, then $\delta_+ \leq (1+4\kappa)\delta^2$ which means that the full-Newton step ensures the local quadratic convergence of the proximity measure through the algorithm.*

*Proof.* From Lemma 5.5, it follows that

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

This completes the proof. □

### 5.2.1 Complexity analysis of Algorithm

The next lemma examines what is the effect of the full-Newton step on the duality gap during the algorithm.

**Lemma 5.7.** *Let $\delta = \delta(xy;\mu)$ and suppose that the vectors $x_+$ and $y_+$ are obtained using a full-Newton step, thus $x_+ = x + \Delta x$ and $y_+ = y + \Delta y$. We have*

$$(x_+)^T y_+ \leq \mu(n + 2\delta^2),$$

*and if $\delta \leq \frac{1}{\sqrt{2}(1+4\kappa)}$, then*

$$(x_+)^T y_+ \leq 2\mu n. \tag{5.10}$$

*Proof.* Using (5.9), we obtain $(x_+)^T y_+ = \mu e^T(e + d_x^T d_y) = \mu(n + d_x^T d_y)$. Next, due to (5.7), it follows that $(x_+)^T y_+ \leq \mu(n+\delta^2)$. Now, let $\delta \leq \frac{1}{\sqrt{2}(1+4\kappa)}$ then $\delta^2 \leq \frac{1}{2(1+4\kappa)^2} < 1$ from which it implies that $(x_+)^T y_+ \leq \mu(n+1)$. But since $n+1 \leq 2n$, $\forall n \geq 1$, this gives the required result. □

In the next theorem, we discuss the influence on the proximity measure of the barrier parameter update $\mu_+ = (1-\theta)\mu$ on the Newton process along the path.

**Theorem 5.8.** *Let $\delta \leq \frac{1}{\sqrt{2}(1+4\kappa)}$ and $\mu_+ = (1-\theta)\mu$, where $0 \leq \theta < 1$. Then*

$$\delta^2(x_+y_+;\mu_+) \leq \frac{1}{(1+4\kappa)^2}\left(\frac{1}{4} + \frac{1}{8(1-\theta)}\right).$$

*Moreover, if $\theta = \frac{1}{\sqrt{2(n+1)(1+4\kappa)}}$ and $n \geq 2$, then $\delta(x_+y_+;\mu_+) \leq \frac{1}{\sqrt{2}(1+4\kappa)}$.*

*Proof.* We have,

$$\begin{aligned}4\delta^2(x_+y_+;\mu_+) &= \|\sqrt{1-\theta}v_+^{-1}-\frac{1}{\sqrt{1-\theta}}v_+\|^2 = \|\sqrt{1-\theta}(v_+^{-1}-v_+)-\frac{\theta}{\sqrt{1-\theta}}v_+\|^2\\ &= (1-\theta)\|v_+^{-1}-v_+\|^2+\frac{\theta^2}{1-\theta}\|v_+\|^2-2\theta(v_+^{-1}-v_+)^Tv_+\\ &= (1-\theta)\|v_+^{-1}-v_+\|^2+\frac{\theta^2}{1-\theta}\|v_+\|^2-2\theta(v_+^{-1})^Tv_++2\theta(v_+)^Tv_+\\ &= 4\delta_+^2(1-\theta)+\frac{\theta^2}{1-\theta}\|v_+\|^2-2\theta n+2\theta\|v_+\|^2.\end{aligned}$$

Because $(v_+^{-1})^Tv_+=n$ and $v_+^Tv_+=\|v_+\|^2$. Next, by Lemma 5.7, implies that

$$\|v_+\|^2=\frac{1}{\mu}x_+^Ty_+\leq n+1,$$

which it follows that

$$\delta^2(x_+y_+;\mu_+)\leq(1-\theta)\delta_+^2+\frac{\theta^2(n+1)}{4(1-\theta)}+\frac{\theta\delta^2}{2}.$$

Let $\delta\leq\frac{1}{\sqrt{2}(1+4\kappa)}$ so $\delta_+\leq\frac{1}{2(1+4\kappa)}$, and $\theta=\frac{1}{\sqrt{2(n+1)}(1+4\kappa)}$, then $\theta^2=\frac{1}{2(n+1)(1+4\kappa)^2}$, these imply that

$$\begin{aligned}\delta^2(x_+y_+;\mu_+) &\leq \frac{1-\theta}{4(1+4\kappa)^2}+\frac{\frac{1}{2(n+1)(1+4\kappa)^2}(n+1)}{4(1-\theta)}+\frac{\theta}{4(1+4\kappa)^2}\\ &\leq \frac{1}{(1+4\kappa)^2}\left(\frac{1}{4}+\frac{1}{8(1-\theta)}\right).\end{aligned}$$

For $n\geq 2$, $\theta\in\left[0,\frac{1}{\sqrt{6}(1+4\kappa)}\right]$ and consider the following function

$$h(\theta)=\frac{1}{(1+4\kappa)^2}\left(\frac{1}{4}+\frac{1}{8(1-\theta)}\right).$$

This function is continuous and monotone increasing on $\left[0,\frac{1}{\sqrt{6}(1+4\kappa)}\right]$ since $h'(\theta)>0$. Therefore

$$h(\theta)\leq h(\frac{1}{\sqrt{6}(1+4\kappa)})=\frac{g(\kappa)}{(1+4\kappa)^2},$$

where

$$g(\kappa)=\frac{1}{4}+\frac{1}{8(1-\frac{1}{\sqrt{6}(1+4\kappa)})}.$$

Since $g(\kappa)$ is a monotonic decreasing function for $\kappa\geq 0$, then $g(\kappa)\leq g(0)$. Hence

$$h(\theta)\leq\frac{g(0)}{(1+4\kappa)^2}\leq\frac{1}{2(1+4\kappa)^2},$$

since $g(0) = \frac{\sqrt{6}-9}{4\sqrt{6}-24} \in (0,1)$.

Then, after the barrier parameter is update to $\mu_+ = (1-\theta)\mu$ with $\theta = \frac{1}{\sqrt{2(n+1)(1+4\kappa)}}$ and if $\delta \leq \frac{1}{\sqrt{2}(1+4\kappa)}$, we obtain $\delta(x_+y_+;\mu_+) \leq \frac{1}{\sqrt{2}(1+4\kappa)}$. This completes the proof. □

Theorem 5.8 indicates that Algorithm 5.1.1 is well-defined since the conditions $x_+ > 0$, $y_+ > 0$ and $\delta(x_+y_+;\mu_+) \leq \frac{1}{\sqrt{2}(1+4\kappa)}$ are maintained through the algorithm.

In the next lemma we compute a bound for the number of iterations produced by the Algorithm 2.4.

**Lemma 5.9.** *Assume that the pair $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$ with $\tau = \frac{1}{\sqrt{2}(1+4\kappa)}$ for each $\mu_0 > 0$. Let $x^k$ and $y^k$ be the vectors obtained after $k$ iterations. Then the inequality $(x^k)^T y^k \leq \epsilon$ is satisfied if $k \geq \left\lceil \frac{1}{\theta} \log \frac{2n\mu_0}{\epsilon} \right\rceil$.*

*Proof.* It follows from (5.10) in Lemma 5.7 that $(x^k)^T y^k \leq 2n\mu_k = 2n(1-\theta)^k\mu_0$. In this way $(x^k)^T y^k \leq \epsilon$ stands if $2n(1-\theta)^k\mu_0 \leq \epsilon$. We take logarithms, so we may write $k\log(1-\theta) \leq \log\epsilon - \log(2n\mu_0)$. We know that $-\log(1-\theta) \geq \theta$ for $0 \leq \theta < 1$, so the inequality holds only if $k\theta \geq \log\epsilon - \log(2n\mu_0) = \log\left(\frac{2n\mu_0}{\epsilon}\right)$. This proves the lemma. □

**Theorem 5.10.** *Let $\theta = \frac{1}{\sqrt{2(n+1)(1+4\kappa)}}$ and $\tau = \frac{1}{\sqrt{2}(1+4\kappa)}$ and suppose that $\mu_0 = \frac{1}{2}$ with $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$. Then the Algorithm 5.1.1 requires at most $\mathcal{O}\left(\sqrt{n}(1+4\kappa)\log\frac{n}{\epsilon}\right)$ iterations for getting an $\epsilon$- approximated solution of $P_\star(\kappa)$-NCP.*

*Proof.* Let $\theta = \frac{1}{\sqrt{2(n+1)(1+4\kappa)}}$ and $\mu_0 = \frac{1}{2}$, by using Lemma 5.9, the proof is straightforward. This completes the proof. □

## 5.3 Implementation and numerical results

In this section, to evaluate the efficiency of **Algorithm 5.1.1**, we present some numerical results for some examples of NCPs of different sizes. In our implementation, we use $\epsilon = 10^{-7}$ and $\mu_0 = \frac{(x^0)^T y^0}{n}$. The initial point $(x^0, y^0)$ in the algorithm is taken such that $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$. An exact solution of the NCP is denoted by $x^\star$.

**Problem 1.** Let $F : \mathbb{R}^4 \to \mathbb{R}^4$ be a monotone mapping, which is defined by

$$F(x) = \begin{cases} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 3x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 3x_4 - 1 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3. \end{cases}$$

For this example, the initial point for the algorithm are obtained via the solution of this problem, where

$$x^0 = e^T, \;\; y^0 = (5, 7, 10, 6)^T,$$

with the proximity $\delta(x^0y^0, \mu_0) = 0.2582$, i.e., $(x^0, y^0) \in \mathcal{N}(\tau, \mu_0)$. Also, here $\kappa = 0$ and $n = 4$, so $\theta = \frac{1}{\sqrt{2(n+1)}}$.

We have the best result ITER=52 and CPU=1.8791. An exact solution of NCP is given by

$$x^\star = \left(\frac{\sqrt{6}}{2},\, 0,\, 0,\, \frac{1}{2}\right)^T, \quad F(x^\star) = \left(0,\, \frac{\sqrt{6}}{2} + 2,\, 5,\, 0\right)^T.$$

**Problem 2.** Let $F : \mathbb{R}^9 \to \mathbb{R}^9$ be a monotone mapping, which is defined by

$$F(x) = \begin{cases} x_2(x_1 + 1) \\ x_3(\frac{1}{3}x_2 - 1) \\ x_3^2 - x_5 \\ x_4 + x_7^2 + 2x_8 - 1 \\ x_5 - 1 \\ x_5x_6 + x_7 - 1 \\ x_3(x_2 - x_7) + x_1x_7 \\ x_6 - x_7 + 3x_8 + 1 \\ -3x_1 + x_2 + 3x_3 - 2x_4 - 2x_5 + 3x_6 - 2x_7 + 3x_8 + 2x_9. \end{cases}$$

Also across this problem, we get the following initial point:

$$x^0 = (0.85, 4, 3, 2, 5, 1.5, 0.9, 1.5, 1.25)^T, \;\; y^0 = (7.4, 3, 4, 4.81, 4, 7.4, 10.065, 6.1, 6.15)^T,$$

with $\delta(x^0 y^0, \mu_0) = 0.4744$. Now, since $\kappa = 0$ and $n = 9$, so $\theta = \frac{1}{\sqrt{2(n+1)}}$. We have the best result ITER=82 and CPU=4.0995. An exact solution of NCP is given by

$$x^\star = (0,\, 2,\, 1,\, 1,\, 1,\, 1,\, 0,\, 0,\, 0)^T, \quad F(x^\star) = (2,\, 0,\, 0,\, 0,\, 0,\, 0,\, 2,\, 2,\, 4)^T.$$

**Problem 3.** Let $F : \mathbb{R}^n \to \mathbb{R}^n$ which is defined by

$$F_i(x) = -x_{i+1} + 2x_i - x_{i-1} + \frac{1}{3}x_i^3 - b_i,\ b_i = (-1)^i,\ i = 1, \ldots, n, \ \text{and}\ \ x_0 = x_{n+1} = 0.$$

The number of iterations and CPU time for different sizes $n$, are summarized in Table 5.1.

Table 5.1: Number of iterations and CPU time for **Problem 3.**

| $n =$ | 15 | 25 | | 50 | | 100 | |
|---|---|---|---|---|---|---|---|
| ITER | CPU | ITER | CPU | ITER | CPU | ITER | CPU |
| 95 | 11.8380 | 126 | 29.0884 | 186 | 118.2347 | 276 | 618.5556 |

**Problem 4.** An example of $P_\star(\kappa)$-LCP with size $n$, which is defined by $F(x) = Mx + q$, where

$$M = \begin{pmatrix} Q_2 & \mathbf{0} & \ldots & \ldots & \mathbf{0} \\ \mathbf{0} & Q_3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & Q_2 & \mathbf{0} \\ \mathbf{0} & \ldots & \ldots & \mathbf{0} & Q_3 \end{pmatrix}, \text{ where } Q_2 = \begin{pmatrix} 0 & 1+4\kappa_1 \\ 1 & 0 \end{pmatrix}, Q_3 = \begin{pmatrix} 0 & 1+4\kappa_2 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and $q = -Me + e$, thus, $x^0 = y^0 = e$ are the initial points for Algorithm 2.4, where $\delta(x^0 y^0; \mu_0) = 0$. The handicap $\kappa(M) = \kappa_1(M) = \kappa_2(M) \in \{0,\ \frac{1}{2},\ 1,\ 5,\ 10\}$. The number of iterations and CPU time results, for different sizes $n$ are summarized in Table 5.2.

Table 5.2: Number of iterations and CPU time for **Problem 4.**

| $n$ | $\kappa$ | ITER | CPU |
|---|---|---|---|
| 10 | $\frac{1}{2}$ | 250 | 0.1089 |
| | 1 | 423 | 0.1844 |
| | 5 | 1806 | 0.7549 |
| | 10 | 3534 | 1.6333 |
| 25 | $\frac{1}{2}$ | 409 | 0.4451 |
| | 1 | 688 | 0.5313 |
| | 5 | 2919 | 2.07532 |
| | 10 | 5708 | 3.7239 |
| 50 | $\frac{1}{2}$ | 597 | 6.8034 |
| | 1 | 1002 | 9.5784 |
| | 5 | 4239 | 33.8677 |
| | 10 | 8285 | 61.8657 |
| 100 | $\frac{1}{2}$ | 874 | 11.5629 |
| | 1 | 1463 | 20.6772 |
| | 5 | 6175 | 77.4400 |
| | 10 | 12066 | 141.8432 |

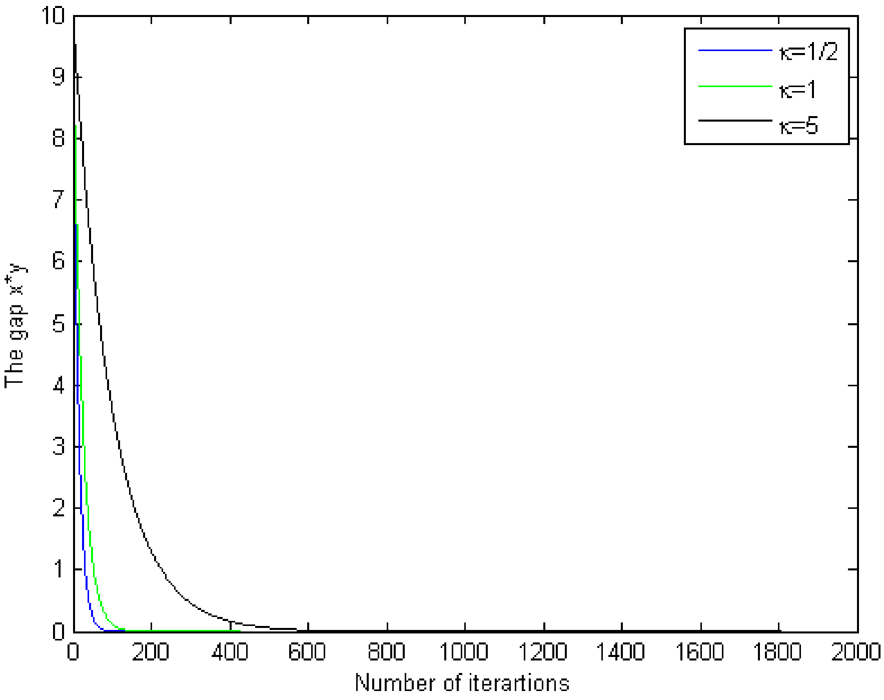

**Fig. 5.1** A comparison of the different value of the handicap $\kappa$ where $n = 10$

**Comment.** Across the numerical results obtained by **Algorithm 5.1.1** stated in the above tables, we see clearly the influence of the handicap $\kappa$ and the size $n$ of the NCPs on our update-barrier $\theta$ provided by our analysis. In fact, as $n$ and $\kappa$ become very large so $\theta$ becomes very small and consequently, the rate of decrease $(1-\theta)$ in the sequence $\mu_k$ approaches to one. This leads to a slow convergence since the number of iterations grows rapidly to infinity (see for example the numerical results for Problem 4 in Table 5.2 and Fig. 5.1). Furthermore, we mention that in some examples of NCPs, the handicap $\kappa$ is not known where we can not calculate exactly the update-barrier $\theta$ during the algorithm. This case will be examined in Subsection 5.3.2.

### 5.3.1 An efficient numerical amelioration for the Algorithm 5.1.1

In this subsection, based on the above comment and in order to ameliorate the performances of **Algorithm 5.1.1**, we import some changes, where instead of using the default of $\theta$ provided by our analysis, we select it as a constant which belongs to the set $\mathcal{B} = \{0.1, \ldots, 0.9\}$. With this modification, we note that the iterations do not maintain interiors through the algorithm, we then introduce a step-size $\alpha_{\max} > 0$ such that $x + \rho\alpha_{\max}\Delta x > 0$ and $y + \rho\alpha_{\max}\Delta y > 0$ with $\alpha_{\max} = \min\{\alpha_x, \alpha_y\}$ and $\rho \in (0,1)$ where $\alpha_x$ and $\alpha_y$ are given by

$$\alpha_x = \begin{cases} \min\limits_i \left(-\dfrac{x_i}{\Delta x_i}\right) & \text{if} \quad \Delta x_i < 0 \\ 1 & \text{if} \quad \Delta x_i \geq 0 \end{cases}, \quad \alpha_y = \begin{cases} \min\limits_i \left(-\dfrac{y_i}{\Delta y_i}\right) & \text{if} \quad \Delta y_i < 0 \\ 1 & \text{if} \quad \Delta y_i \geq 0. \end{cases}$$

Our new obtained numerical results are summarized in Tables 5.3, 5.4, 5.5 and 5.6 where the same initial points are taken in the algorithm.

Table 5.3: Numerical results for **Problem 1**. by **Ameliorated Algorithm**

| | $0.5$ | | $0.7$ | | $0.9$ | |
|---|---|---|---|---|---|---|
| | ITER | CPU | ITER | CPU | ITER | CPU |
| **Problem 1** | 29 | 1.0913 | 17 | 0.6486 | 9 | 0.3509 |

Table 5.4: Numerical results for **Problem 2**. by **Ameliorated Algorithm**

| | $0.5$ | | $0.7$ | | $0.9$ | |
|---|---|---|---|---|---|---|
| | ITER | CPU | ITER | CPU | ITER | CPU |
| **Problem 2** | 30 | 1.6252 | 18 | 0.9115 | 9 | 0.4813 |

Table 5.5: Numerical results for **Problem 3**. by **Ameliorated Algorithm**

| $\theta$ | $0.5$ | | $0.7$ | | $0.9$ | |
|---|---|---|---|---|---|---|
| $n$ | ITER | CPU | ITER | CPU | ITER | CPU |
| 8 | 26 | 1.6461 | 15 | 0.9672 | 8 | 0.5811 |
| 15 | 27 | 2.8298 | 16 | 1.8054 | 8 | 0.9720 |
| 25 | 28 | 5.9430 | 16 | 3.0810 | 9 | 2.0497 |
| 50 | 29 | 13.8829 | 17 | 11.5730 | 9 | 4.7274 |
| 100 | 30 | 65.1629 | 17 | 34.6950 | 9 | 19.4291 |
| 500 | 32 | 255.0226 | 18 | 132.9599 | 10 | 76.1361 |

Table 5.6: Numerical results for **Problem 4**. by **Ameliorated Algorithm**

| | $\theta$ | $0.5$ | | $0.7$ | | $0.9$ | |
|---|---|---|---|---|---|---|---|
| $n$ | $\kappa$ | ITER | CPU | ITER | CPU | ITER | CPU |
| 10 | $\frac{1}{2}$ | 27 | 0.0103 | 16 | 0.0053 | 8 | 0.0027 |
| | 1 | 27 | 0.0114 | 16 | 0.0066 | 8 | 0.0027 |
| | 5 | 27 | 0.0110 | 16 | 0.0069 | 8 | 0.0027 |
| | 10 | 27 | 0.0149 | 16 | 0.0064 | 8 | 0.0027 |
| 25 | $\frac{1}{2}$ | 28 | 0.0289 | 17 | 0.0160 | 9 | 0.0086 |
| | 1 | 28 | 0.0309 | 17 | 0.0156 | 9 | 0.0083 |
| | 5 | 28 | 0.0318 | 17 | 0.0197 | 9 | 0.0081 |
| | 10 | 28 | 0.0348 | 17 | 0.0208 | 9 | 0.0082 |
| 50 | $\frac{1}{2}$ | 29 | 0.3994 | 17 | 0.1923 | 9 | 0.0360 |
| | 1 | 29 | 0.4291 | 17 | 0.2394 | 9 | 0.0753 |
| | 5 | 29 | 0.4489 | 17 | 0.2666 | 9 | 0.1320 |
| | 10 | 29 | 0.5337 | 17 | 0.2836 | 9 | 0.1543 |
| 100 | $\frac{1}{2}$ | 30 | 0.7073 | 18 | 0.3964 | 9 | 0.1871 |
| | 1 | 30 | 0.6704 | 18 | 0.4228 | 9 | 0.2157 |
| | 5 | 30 | 0.6865 | 18 | 0.4675 | 9 | 0.2236 |
| | 10 | 30 | 0.7197 | 18 | 0.5112 | 9 | 0.2406 |

### 5.3.2 Computational results on $P_\star(\kappa)$-LCP with Csizmadia-matrix

This problem is based on the $P$-matrix proposed by Zsolt Csizmadia and presented in [50], where $M \in \mathbb{R}^{n\times n}$ and $q$ are given by

$$M = (m_{ij}) = \begin{cases} 0 & \text{if} \quad i < j, \\ 1 & \text{if} \quad i = j, \\ -1 & \text{if} \quad i > j, \end{cases} \quad q = -Me + e.$$

In this example, we mention that the handicap $\kappa$ is not given explicitly, we have only $\kappa(M) \geq 2^{2n-8} - \dfrac{1}{4}$ as De Klerk and E.-Nagy [50] proved. Therefore our theoretical updating $\theta$ can not be computed exactly due to the absent of the value of $\kappa$. Then we are also obliged to take $\theta$ as a constant.

Table 5.7: Numerical results for **Csizmadia matrix**.

| | $0.1$ | | $0.2$ | |
|---|---|---|---|---|
| $n$ | ITER | CPU | ITER | CPU |
| 8 | 173 | 0.1452 | 82 | 0.0738 |
| 15 | 179 | 0.2452 | 85 | 0.1069 |
| 25 | 184 | 0.2576 | 87 | 0.1489 |
| 50 | 191 | 3.4756 | 90 | 1.9728 |
| 100 | 197 | 4.9863 | 93 | 3.5600 |
| 500 | 212 | 64.1363 | 101 | 31.7542 |

## 5.4 Conclusion

In this chapter, we presented a feasible short-step path-following IPA for solving $P_\star(\kappa)$-NCPs. We adopted the basic analysis used by many authors for $P_\star(\kappa)$-LCPs and develop them to be suited for $P_\star(\kappa)$-NCPs. We proved that short-step algorithm requires $\mathcal{O}\left(\sqrt{n}(1+4\kappa)\log(\dfrac{n}{\epsilon})\right)$ iteration bound for getting an approximated solution of $P_\star(\kappa)$-NCPs. Preliminary obtained numerical results are very encouraging.

# General conclusion and future work

In this thesis, we have presented some interior-point methods for solving some classes of complementarity problems (linear and nonlinear).

In Chapters 2 and 3, a polynomial short-step primal-dual interior-point methods (IPMs) are proposed for solving the monotone linear complementarity problems ( MLCPs).

Firstly, we have used only the classical search directions i.e., $\psi(t) = t$. At each iteration the algorithm uses only full-Newton steps without using line search. For its well definiteness and its local convergence, a suitable defaults update barrier $\theta$ and threshold $\tau$ are selected for this purpose. Some numerical results are presented which show the efficiency of this algorithm for solving MLCPs.

Secondly, instead to using the classical Newton search directions, we have used the technique of algebraically equivalent transformation (AET) induced by the univariate function $\psi(t) = t^{\frac{5}{2}}$ to the central path to offer new search directions. With the same manner as in Chapter 2, we have analyzed a short-step primal-dual interior-point methods for MLCPs. Also, under new defaults update barrier $\theta$ and threshold $\tau$ the algorithm is well-defined and converge locally quadratically with polynomial complexity to a solution of MLCPs. some experimental results are also given.

Regarding the numerical results obtained by the algorithm we have concluded that the convergence is very slow towards the solution of MLCPs. To remedy this drawback, we have ameliorated this algorithm where instead some Netlib testing problems.

In Chapter 4, by exploiting the strategy of minorant and majorant functions for computing step-lengths we have proposed a logarithmic barrier methods for solving also MLCPs. Some numerical results are given followed by a comparison Wolfe's classical line search rule.

Finally, we end the thesis by extending the idea of Chapter 2 i.e. the development and the analysis of short-step path-following IPMs for solving the class of $P_{\star}(\kappa)-$NCP. Also, some new defaults depending on the handicap $\kappa$ are choosing for proving the well definiteness of the algorithm. The numerical results are not good because as the dimension of problem $n$ and the handicap $\kappa$ became very large the $\theta$ tends to zero. In some problems of $P_{\star}(\kappa)-$NCP handicap $\kappa$ is absent. therefore, our $\theta = \frac{1}{\sqrt{2(n+1)(1+4\kappa)}}$ is not valuable and consequently we have obliged to deviate the algorithm where we use some $\theta$ independents of the handicap $\kappa$ and the dimension $n$. For maintain its strict feasibility, we have introduced a step-lengths $\alpha$ to ensure this goal.

In general, we have treated in this thesis the following points such as : the direction, the step-lengths, the initial points to improving the performance of these algorithms to solve MLCPs and $P_{\star}(\kappa)-$NCPs.

## Future work

In this section, an interesting topics of research in the future are:

1. The extension of Algorithm 3.1.1 to the class of $P_{\star}(\kappa)$-LCPs.

2. The development of an infeasible full-Newton step IPA for monotone LCP based on our AET in Chapter 3.

3. It would be interesting to use the new approach of majorant and minorant functions to other classes of nonlinear optimization problems.

4. The extension of these methods for more general semidefinite complementarity problems such as the semidefinte nonlinear complementarity problems (SNLCP).

# Appendices

## .1 Appendix A

### .1.1 Newton-Raphson's method for solving nonlinear systems

Newton's method for solving the nonlinear system of equations

$$G(x) = 0.$$

where $G : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable. generates a sequence $\{x^k\}$ according to the following rule:

$$x^{k+1} = x^k - (J(x^k))^{-1}G(x^k),\ k = 0, 1, \dots$$

where $x^0 \in \mathbb{R}^n$ is a suitable starting point.

We define the $n \times n$ Jacobian matrix $J(x)$ as

$$J(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \frac{\partial g_1}{\partial x_2}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \frac{\partial g_2}{\partial x_1}(x) & \frac{\partial g_2}{\partial x_2}(x) & \dots & \frac{\partial g_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1}(x) & \frac{\partial g_n}{\partial x_2}(x) & \dots & \frac{\partial g_n}{\partial x_n}(x) \end{bmatrix},$$

where $\frac{\partial g_i}{\partial x_j}(x)$ is called the partial derivative of $g_i$ at $x_j$.

The Hessian matrix is a matrix of second order partial derivatives, define as follow:

$$H(x) = \begin{bmatrix} \frac{\partial^2 g_1}{\partial x_1^2}(x) & \frac{\partial^2 g_1}{\partial x_1 \partial x_2}(x) & \dots & \frac{\partial^2 g_1}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 g_2}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 g_2}{\partial x_2^2}(x) & \dots & \frac{\partial^2 g_2}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 g_n}{\partial x_n \partial x_1}(x) & \frac{\partial^2 g_n}{\partial x_n \partial x_2}(x) & \dots & \frac{\partial^2 g_n}{\partial x_n^2}(x) \end{bmatrix}.$$

## .1.2 Solving LCP without having an initial interior-point

Feasible IPMs start with a strictly feasible interior point and the iterates maintain feasible during the computation process. However, to obtain such an initial point for some MLCPs, we have been use the homogeneous and self-dual LCPs model considered by Ye [75].

## .1.3 Line search Wolfe and Powell strategy

The Algorithm .1.1 helps to understand the structure of the Wolfe and Powell line search procedure.

**Algorithm .1.1** The Wolfe and Powell line search algorithm

```
Initialize $\lambda_{\text{min}} = 0$, $\lambda_{\text{max}} = L$ ($L \in \mathbb{R}_+$ enough big), $m_1 \in ]0,1[$ and $m_2 \in ]m_1,1[$
while Arret=Faux do
    calculate $f(x^k + \lambda d^k)$
    if $f(x^k + \lambda d^k) \leq f(x^k) + m_1 \lambda \nabla f(x^k)$ then
        calculate $\nabla f(x^k + \lambda d^k)$
        if $\nabla f(x^k + \lambda d^k) \geq m_2 \nabla f(x^k)$ then
            Arret=Vrai
        End
    End
    if Arret=Faux then
        $\lambda_{\text{min}} = \lambda$
    else
        $\lambda_{\text{max}} = \lambda$
    End
    $\lambda = \dfrac{\lambda_{\text{min}} + \lambda_{\text{max}}}{2}$
End
```

## .1.4 Netlib collection LO tests

To get the collection of LO test problems gathered by David Gay [37] connect to netlib site http://www.netlib.org/lp.

### .1.5 LU decomposition

This decomposition is used in numerical analysis to solve system of linear equations $Ax = b$ where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. The square matrix $A$ can be decomposed into two square matrices $L$ and $U$ such that $A = LU$, where $U$ is an upper triangular matrix formed as a result of applying the Gauss elimination method on $A$, and $L$ is a lower triangular matrix with diagonal elements being equal to $1$. We substitute $A = LU$. Thus, we have $LUx = b$. We put $Z = Ux$, where $Z$ is a vector of artificial variables and solve for $LZ = b$ first and then solve for $Ux = Z$ to find $x$.

### .1.6 Implementation and numerical results

We implemented the above algorithms on the software **MATLAB 7.9** and run on a **PC** with **CPU 2.13 GHz** and **2G RAM** memory and double precision format.

## .2 Appendix B

The following MATLAB codes for the **Algorithms 2.2.1** and **3.2.1** respectively.

```
%%%%% Problem 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=4;               %dimension of the problem
M=[2,1,1,1;1,2,0,1;1,0,1,2;-1,-1,-2,0];%PSD matrix
q=[1;-2;4;10;2];   %constant vector q
%%%%% Problem 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=7;
Q=[1,0,-0.5,0;0,0.5,0,0;-0.5,0,1,0.5;0,0,0.5,0.5];
A=[1,2,1,1;3,1,2,-1;0,-1,-4,0];
M=[Q,A';-A,zeros(3,3)]
q=[-1;3;1;-1;5;6;1.5];
%%%%% Problem 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=input('give the value of size n=')
M=4*eye(n);
for i=1:n
    for j=1:n-1
        if(i==j)&&(i==1)
            M(i,j+1)=1;
            M(i,j+1)=-2*M(i,j+1);
        end
        if(i==j)&&(i>1)
            M(i,j+1)=1;
            M(i,j+1)=-2;
            M(i,j-1)=1;
            M(i,j-1)=-2;
        end
        if(i==j+1)&&(i==n)
            M(i,j)=1;
            M(i,j)=-2;
        end
    end
end
q=ones(n,1);q(1)=-1;q(n)=-1;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

e=ones(n,1);       %vector of all ones
N=eye(n);          %identity matrix of order n
x=ones(n,1)        %feasible initial starting vectors x and y
y=[M*x+q];
mu=0.5;
theta=1/sqrt(2*n+2) %barier parameter
v=sqrt((x.*y)/mu)
del=norm(v.^(-1)-v)  % the proximity measure
epsilon=10^-6;     %accuracy parameter
Iter=0;            %number of iteration
tic
while epsilon<=n*mu
    mu=(1-theta)*mu;
    X=diag(x);
    Y=diag(y);
    Z=[-M,N;Y,X];
    b=[zeros(n,1);(mu*e)-(x.*y)];
    [L,U]=lu(Z);
    d=inv(U)*(inv(L)*b);
    dx=d(1:n)
    dy=d(n+1:2*n)
    x=x+dx
    y=y+dy
    Iter=Iter+1
end
toc
```

```
%%%%% Problem 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=5;                %dimension of the problem
M=[6,6,4,3,2;8,21,14,10,12;4,14,13,5,9;4,10,5,6,5;3,12,8,4,10];%P
SD matrix
q=[-20.5;-64.5;-44.5;-29.5;-36.5];   %constant vector q
x=ones(n,1);        %feasible initial starting vectors x and y
y=[M*x+q];
%%%%% Problem 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=8
M=[8,9,13,13,5,11,9,10;8,10,15,15,7,12,10,12;13,15,26,26,10,20,13
,21;
13,15,26,26,10,20,12,20;5,7,10,10,5,9,5,8;
11,12,20,20,9,19,13,15;9,10,13,12,5,13,16,13;
10,12,21,20,8,15,13,22];
q=[-8.265;-9.3033;-14.835;-14.4633;-5.995;-12.4133;-10.015;-
12.3033]
x=[0.2233;0.1893;0.1207;0.1202;0.2758;0.1431;0.1961;0.1403]
y=[M*x+q];
%%%%% Problem 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=input('give the value of size n=')
M=ones(n,n);
for k=1:n
    M(k,:)=4*k-3+1;
    M(k,k)=4*k-3;
end
for k=1:n
    for s=1:n
        M(s,k)=M(k,s);
    end
end
e=ones(n,1);       %vector of all ones
q=-M*e+e
```

```
x=e
y=e
N=eye(n);          %identity matrix of order n
mu=0.5;
theta=1/(35*sqrt(2*n)) %barier parameter
v=sqrt((x.*y)/mu)
del=norm(v.^(-4)-v)  % the proximity measure
epsilon=10^-4;     %accuracy parameter
Iter=0;            %number of iteration
tic
while epsilon<=n*mu
    mu=(1-theta)*mu;
    X=diag(x);
    Y=diag(y);
    Z=[-M,N;Y,X];
    b=[zeros(n,1);(2*mu/5)*(((x.*y)/mu).^((-3)/2)-(x.*y)/mu)];
    [L,U]=lu(Z);
    d=inv(U)*(inv(L)*b);
    dx=d(1:n)
    dy=d(n+1:2*n)
    x=x+dx;
    y=y+dy;
    Iter=Iter+1 ;
end
toc
```

# Bibliography

[1] M. Achache, A weighted-path-following method for the linear complementarity problem, Studia Univ. Babeş-Bolyai. Ser. Informatica, 49(1), 61-73 (2004)

[2] M. Achache, A new primal-dual path-following method for convex quadratic programming, Computational and Applied Mathematics, 25(1), 97-110 (2006). https://doi.org/10.1590/S0101-82052006000100005

[3] M. Achache, H. Roumili, A. Keraghel, A numerical study of an infeasible primal-dual path-following algorithm for linear programming. Appl. Math Comput., 186(2), 1472-1479 (2007). https://doi.org/10.1016/j.amc.2006.07.135

[4] M. Achache, N. Boudiaf and A. Keraghel, Numerical experiments with two smoothing type-algorithms for nonlinear complementarity problems, Rev. Anal. Numér. Théor. Approx., 37(1), 3-16 (2008)

[5] M. Achache, Complexity analysis and numerical implementation of a short-step primal-dual algorithm for linear complementarity problems, Appl. Math. Comput., 216(7), 1889-1895 (2010). https://doi.org/10.1016/j.amc.2010.03.015

[6] M. Achache, R. Khebchache, A full-Newton step feasible weighted primal-dual interior point algorithm for monotone LCP, Afrika Matematika, 26, 139–151 (2015). https://doi.org/10.1007/s13370-013-0193-z

[7] M. Achache, Complexity analysis of an interior point algorithm for the semidefinite optimization based on a 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

[8] M. Achache, M. Goutali, Complexity analysis and numerical implementation of a full-Newton step interior-point for LCCO, J. Numer. Algoritms, 70, 393-405 (2015). https://doi.org/10.1007/s11075-014-9955-4

[9] M. Achache, A new parameterized kernel function for LO yielding the best known iteration bound for a large-update interior point algorithm, Afrika Matematika, 27(3), 591–601(2016). https://doi.org/10.1007/s13370-015-0363-2

[10] M. Achache, Complexity analysis of a weighted-full-Newton step interior-point algorithm for $P_{\star}(\kappa)$-LCP, RAIRO Operations Research, 50, 131-143 (2016)

[11] M. Achache, N. Tabchouche, Complexity analysis and numerical implementation of large-update interior-point methods for SDLCP based on a new parametric barrier kernel function, Optimization, Journal of Mathematical Programming and Operations Research, 67(8), 1211-1230 (2018). https://doi.org/10.1080/02331934.2018.1462356

[12] 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

[13] B. Alzalg, A logarithmic barrier interior- point method based on majorant functions for second-order cone programming. Optimization Letters, 14, 729-746 (2020). https://doi.org/10.1007/s11590-019-01404-1

[14] Y.Q. Bai, M. El Ghami and C. Roos, A new efficient large-update primal-dual interior-point method based on finite barrier, SIAM Journal on optimization, 13(3), 766-782 (2002) https://doi.org/10.1137/S1052623401398132

[15] Y. Bai, M. El Ghami, C. Roos, A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization, SIAMJ. Optim., 15(1), 101-128 (2004). https://doi.org/10.1137/S1052623403423114

[16] Y.Q. Bai, G. Lesaja, C. Roos, A new class of polynomial interior-point algorithms for $P^{\star}(\kappa)-$linear complementary problems. Pac. J. Optim., 4(1), 19–41 (2008)

[17] J.F. Bonnans, C.C. Gonzaga, Convergence of interior-point algorithms for monotone linear complementarity problem, Math. Oper. Res., 21(1), 1-15 (1996)

[18] L.B. Cherif, B. Merikhi, A penalty method for nonlinear programming. RAIRO-Operations Research. 53(1),29-38 (2019). https://doi.org/10.1051/ro/2018061

[19] E. Chouzenoux, S. Moussaoui, J. Idier, A majorize-minimize line search algorithm for barrier function optimization. EURASIP European Signal and Image Processing Conference. Glasgow, United Kingdom, 1379-1383 (2009)

[20] R.W. Cottle, J.S. Pang, R.E., Stone The Linear Complementarity Problem, Academic, San Diego, (1992)

[21] R. W. Cottle, F. Giannessi, and J. L.-Lions, eds., Variational Inequalities and Complementarity Problems: Theory and Applications, John Wiley & Sons, New York, NY, (1980)

[22] R. W. Cottle, S.M. Guu, Two characterizations of sufficient matrices, Systems Optimization Laboratory, Stanford University, California, (1990)

[23] J. Crouzeix, B. Merikhi, A logarithmic barrier method for semi-definite programming. RAIRO-Oper. Res. 142, 123-139 (2008). http://eudmi.org/doc/250394

[24] J. Czyzyk, S. Mehrotra, M. Wagner and S.J. Wright, PCx: An interior-point code for linear programming, Optimization methods and software, 11(1), 397-430 (1999) https://doi.org/10.1080/10556789908805757

[25] Z. Darvay, A new algorithm for solving self-dual linear programming problems, Studia Univ. Babeş-Bolyai, Ser. Informatica, 47(1), 15-26 (2002)

[26] Z. Darvay, New interior point algorithm in linear programming, Advanced Modeling and Optimization, 5(1), 51-92 (2003)

[27] Z. Darvay, I. M. Papp, P. R. Takács, Complexity analysis of a full-Newton step interior point method for linear optimization, Period. Math. Hung., 73(1), 27-42 (2016) https://doi.org/10.1007/s10998-016-0119-2

[28] Z. Darvay, P.R. Takács, New method for determining search directions for interior point algorithms in linear optimization, Optimization Letters, 12(5), 1099-1116 (2018). https://doi.org/10.1007/s11590-017-1171-4

[29] Z. Darvay, T. Illés, J. Povh, P. R. Rigó, Feasible corrector-predictor interior-point algorithm for $P_\star(\kappa)$-linear complementarity problems based on a new search directions. SIAM Journal on Optimization, 30(3), 2628-2658 (2020)

[30] Z. Darvay, T. Illés, C. Majoros, Interior-point algorithm for sufficient LCPs based on the technique of algebraically equivalent transformation, Optimization Letters, 15, 357-376 (2021). https://doi.org/10.1007/s11590-020-01612-0

[31] M. Fiedler and V. Ptak, On matrices with non-positive off-diagonal elements and positive principle minors, Czechoslovak Mathematical Journal, 12, 382-400 (1962)

[32] M. Fiedler and V. Ptak, Some generalizations of positive definiteness and monotonicity, Numerische Mathematik, 9, 163-172 (1966)

[33] A. Fischer, A special Newton-type optimization method, Optimization, 24(3-4), 269-284 (1992). 10.1080/02331939208843795

[34] A. Fischer, On the local superlinear convergence of a Newton-type method for LCP under weak conditions, Optimization methods and software, 6(2),83-107 (1995) 10.1080/10556789508805627

[35] M. C. Ferris, O. L. Mangasarian, and J.-S. Pang, editors, Complementarity: Applications, Algorithms and Extensions, Kluwer Academic Publishers, Dordrecht, (2001)

[36] M.C. Ferris, J.S. Pang, Engineering and economic applications of complementarity problems, SIAM Revue, 39, 669–713 (1997)

[37] D. Gay, Electronic mail distribution of linear programming test problems. Math. Program. Soc. COAL Newsl. 3, 10–12 (1985) http://www.netlib.org/lp

[38] W. Grimes, Path-following interior-point algorithm for monotone linear complementarity problems, Asian-European Journal of Mathematics, 15(9) (2021). https://doi.org/S1793557122501704

[39] W. Grimes, M. Achache, An Infeasible Interior-point Algorithm for Monotone Linear Complementarity Problems, International Journal of Informatics and Applied Mathematics, 04 (02), 53–59 (2021). https://doi.org/10.53508/ijiam.1036022

[40] M. Haddou, T. Migot, J. Omer, A Generalized Direction in Interior Point Method for Monotone Linear Complementarity Problems, Optimization Letters, 13, 35-53 (2019) https://doi.org/10.13140/RG.2.2.14536.72963

[41] M. Haddou, P. Maheux, Smoothing Methods for Nonlinear Complementarity Problems, Journal of Optimization Theory and Applications, Springer Verlag, 160(3), 711-729 (2014)

[42] P.T. Harker, J.S Pang, Finite dimensional variational inequality and nonlinear complementarity problems. A survey of theory, algorithms and applications, Mathematical programming, 48, 35-53 (1990)

[43] S. Karamardian, The complementarity problem, Mathematical programming, 2 (1972)

[44] N. K. Karmarkar, A new polynomial-time algorithm for linear programming, Combinatorica, 4, 373-395 (1984)

[45] B. Kheirfam, A predictor-corrector interior-point algorithm for $P_{\star}(k)$-horizontal linear complementarity problem, Numerical Algorithms, 66(2), 349-361 (2014) https://doi.org/10.1007/s11075-013-9738-3

[46] B. Kheirfam, M. Haghighi, A full-Newton step feasible interior-point algorithm for $P_{\star}(k)$-LCP based on a new search direction, Croat. Oper. Rev., 7(2), 277-290 (2016) https://doi.org/10.17535/crorr.2016.0019

[47] B. Kheirfam, A. Nasrollahi, A full-Newton step Interior-point Method based on a Class of Specific Algebra Transformation, Fundamenta Informaticae, 163,325-337 (2018)

[48] B. Kheirfam, A new search direction for full-Newton step infeasible interior-point method in linear optimization, arXiv:2102.07223v1 [math.OC] (2021)

[49] M.-K. Kim, y.-y. Cho, and G.-M. Cho, New path-following interior-point algorithms for $P_{\star}(\kappa)$-nonlinear complementarity problems, Nonlinear Analysis: Real World Applications, 14, 718–733 (2013)

[50] E. De Klerk and M. E.-Nagy, On the complexitiy of computing the handicap of a sufficient matrix, Math. Program., 129, 383-402 (2011)

[51] M. Kojima, S. Mizuno and A. Yoshise, A polynomial algorithm-time for a class of linear complementarity problems, Math. Program., 44(1-3) (1989) 1-26.

[52] M. Kojima, N. Megiddo, T. Noma, A. Yoshise, A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems. Lecture Notes in Computer Science, vol. 538. Springer-Verlag, Berlin, (1991)

[53] M. Kojima, N. Megiddo, S. Mizuno and A. Yoshise, A Unified Approach to Interior Point Algorithms for linear complementarity problems, in: Lecture Notes in Computer Science, vol. 538, Springe-Verlag, Berlin, Germany, (1991)

[54] C.E. Lemke, Bimatrix equilibrium points and mathematical programming, Management Science II, 681-689 (1965)

[55] A. Leulmi, B. Merikhi, D. Benterki, Study of a logarithmic barrier approach for Linear semidefinite programming. Journal of Siberian Federal University. Mathematics and Physics. 11(3), 300-312 (2018). https://doi.org/10.17516/1997-1397-2018-11-3-300-312

[56] O.L. Mangasarian, R.R. Meyer, Absolute value equations, Linear algebra appl., 419, 359-367 (2006)

[57] H. Mansouri, S. Asadi, A Quadratically Convergent $\mathcal{O}((1+\kappa)\sqrt{n})$ Interior-Point Algorithm for the $P_\star(\kappa)$-Matrix Horizontal Linear Complementarity Problem, Journal of Sciences, Islamic Republic of Iran, 23(3), 237-244 (2012)

[58] L. Menniche, Benterki D., A logarithmic barrier approach for linear programming. Journal of Computational and Applied Mathematics, 312(1), 267-275 (2017). https://doi.org/10.1016/j.cam.2016.05.025

[59] T. Migot, Contributions aux méthodes numériques pour les problèmes de complémentarité et problèmes d'optimisation sous contraintes de complémentarité. Doctoral dissertation, Université Bretagne Loire, France (2017)

[60] R.D.C. Monteriro J.S. Pang and T. Wang, A algorithm for nonlinear complementarity problem, SIAM J. Opti., 5(1), 129-148 (1995)

[61] N. Moussaoui, M. Achache, A weighted-path following interior-point algorithm for convex quadratic optimization based on modified search directions, Statistics, optimization & information omputing, 10(3), 873-889 (2022). 10.19139/soic-23-10-5070-1385

[62] S. Pan, X. Li, S. He, An infeasible primal-dual interior point algorithm for linear programs based on logarithmic equivalent transformation, Mathematical Analysis and Applications, 314, 644-660 (2006). https://doi.org/10.1016/j.jmaa.2005.04.028

[63] J. Peng, C. Roos, T. Terlaky, Self-Regular Functions: A New Paradigm for primal-dual Interior-Point Methods, Princeton University Press, Princeton, (2002)

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

[65] J. Peypouquet, Convex optimization in normed spaces, theory, methods and examples. Springer; (2015)

[66] P. R. Rigó, New trends in algebraic equivalent transformation of the central path and its applications, PhD thesis, Budapest University of Technology and Economics, Institute of Mathematics, Hungary, 2020

[67] C. Roos, T. Terlaky and J.P. Vial, Theory and algorithms for Linear Optimization. An Interior Point Approach, Jhon-Wiley. Sons, Chichester, UK, (1997)

[68] G. Q. Wang, Y. Q. Bai, A new primal-dual path-following interior point algorithm for semidefinite programming, Journal of Mathematical Analysis and Applications, 353, 339-349 (2009). https://doi.org/10.1016/j.jmaa.2008.12.016

[69] G. Q. Wang, Y. Q. Bai, A primal-dual interior-point algorithm for second-order cone optimization with full-Nesterov-Todd step, Applied Mathematics and computation, 215, 1047-1061 (2009). https://doi.org/10.1016/j.amc.2009.06.034

[70] G. Q. Wang, Y. Q. Bai, Polynomial interior-point algorithms for horizontal linear complementarity problem, Journal of Computational and Applied Mathematics, 233, 248-263 (2009). https://doi:10.1016/j.cam.2009.07.014

[71] G. Q. Wang, X.J. Fan, D.T. Zhu, D. Z. Wang, New complexity analysis of a full-Newton step feasible interior-point algorithm for $P_\star(k)$-LCP, Optimization Letters, 9(6), 1105-1119 (2015). https://doi.org/10.10007/s11590-014-0800-4

[72] G.Q. Wang, C.J. YU and K.L. Teo, A full-Newtopn step feasible interior-point algorithm for $P_\star(\kappa)-$ linear complementarity problem, Journal of Global Optimization, 59, 81–99 (2014)

[73] S.J. Wright, Primal-Dual Interior Point Methods, SIAM, University city, (1997)

[74] Y. Ye, M.J. Todd, and S. Mizuno, An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm, Mathematics of Operations Research, 19(1), 53-67 (1994). http:/www.jstor.org/stable/3690376

[75] Y. Ye, On homogeneous and self-dual algorithm for LCP. Mathematical Programming, 76(1), (1997) 211-222. https://doi.org/10.1007/BF02614384

[76] Y. Ye, Interior-point algorithms. Theory and Analysis, Jhon-Wiley. Sons, Chichester, UK, (1997)

[77] L. Yong, Iterative method for a class of linear complementarity problems, Int. Conf. Inf. Comput. Appl. 1, 390–398 (2010)

[78] L. Zhang and Y. Xu, A full-Newton step primal dual interior-point algorithm for linear complementarity problem, J. Inf. Comput. Sci., 5(4), 305–313 (2010)

[79] M. Zhang, Y. Bai, G. ,Wang, A new primal–dual path-following interior-point algorithm for linearly constrained convex optimization, J. Shanghai Univ. 12(6), 475–480 (2008)

[80] Y.B. Zhao, J. Han, Two interior-point methods for nonlinear $P_{\star}(\tau)-$complementarity problems, Journal of Optimization Theory and Applications, 102, 659–679 (1999)

## ملخص

في هذه الأطروحة قدمنا عدة خوارزميات ذات النقطة الداخلية و المسار المركزي من أجل حل مسائل التتام (الخطي و غير الخطي)، لهذا الغرض ركزنا على حساب اتجاهات نيوتن القديمة والحديثة واقتراح استراتيجية جديدة لحساب الخطوة. هذه الدراسة أدت إلى حساب الكلفة الحدودية للخوارزميات المعرفة و نتائج عددية فعالة.

**الكلمات المفتاحية :** مسائل التتام (الخطي و غير الخطي)، طريقة النقطة الداخلية، المسار المركزي، اتجاهات نيوتن، الكلفة الحدودية.

## Résumé

Dans cette thèse, nous proposons quelque algorithmes de point-intérieur de type trajectoire centrale pour résoudre les problèmes de complémentarité (linéaire et non linéaire), dans ce but, nous avons concentré sur le calcul des directions de Newton classiques et nouvelles. En outre, nous avons proposé de nouveaux pas de déplacement sur ces directions. Cette étude, conduit à calculer la complexité polynomiale de ces algorithmes, ainsi que des résultats numériques efficaces.

**Mots clés :** Problèmes de complémentarités (linéaire et non linéaire), méthode de point-intérieur, trajectoire centrale, direction de Newton, complexité polynomiale.

## Abstract

In this thesis, we have proposed some path-following interior-point algorithms for solving complementarity problems (linear and nonlinear), for this purpose, we have concentrated on the computing of classical and new Newton’s search directions. Further, we have proposed some new step-sizes on these directions. This study, leads to compute the polynomial complexity for these algorithms, as well as efficient numerical results.

**Key words:** Complementarity problems (linear and nonlinear), interior-point method, path-following, Newton’s direction, polynomial complexity