A lower bound on the iterative complexity of the Harker and Pang globalization technique of the Newton-min algorithm for solving the linear complementarity problem

Jean-Pierre Dussault [†], Mathieu Frappier [‡], and Jean Charles Gilbert [§]

Friday 20$^{\text{th}}$ September, 2019 (13:54)

The plain Newton-min algorithm for solving the linear complementarity problem (LCP) "$0 \leqslant x \perp (Mx+q) \geqslant 0$" can be viewed as an instance of the plain semismooth Newton method on the equational version "$\min(x, Mx+q) = 0$" of the problem. This algorithm converges for any q when M is an **M**-matrix, but not when it is a **P**-matrix. When convergence occurs, it is often very fast (in at most n iterations for an **M**-matrix, where n is the number of variables, but often much faster in practice). In 1990, Harker and Pang proposed to improve the convergence ability of this algorithm by introducing a stepsize along the Newton-min direction that results in a jump over at least one of the encountered kinks of the min-function, in order to avoid its points of nondifferentiability. This paper shows that, for the Fathi problem (an LCP with a positive definite symmetric matrix M, hence a **P**-matrix), an algorithmic scheme, including the algorithm of Harker and Pang, may require n iterations to converge, depending on the starting point.

Keywords: iterative complexity, linear complementarity problem, Fathi and Murty problems, globalization, Harker and Pang algorithm, line search, Newton-min algorithm, nondegenerate matrix, **P**-matrix, semismooth Newton method.

AMS MSC 2010: 15B99, 47B99, 49M15, 65K15, 90C33.

Table of contents

[†] Département d'Informatique, Faculté des Sciences, Université de Sherbrooke, Canada. E-mail: `Jean-Pierre.Dussault@Usherbrooke.ca`.

[‡] Département de Mathématiques, Faculté des Sciences, Université de Sherbrooke, Canada. E-mail: `Mathieu.Frappier@usherbrooke.ca`.

[§] INRIA Paris, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France. E-mail: `Jean-Charles.Gilbert@inria.fr`.

1 Introduction

Let $n \geqslant 1$ be an integer, $M \in \mathbb{R}^{n\times n}$ be a real matrix, $q \in \mathbb{R}^n$ be a real vector, and $[1:n] := \{1,\dots,n\}$ be the set of the first n positive integers. The linear complementarity problem (LCP) consists in searching a vector $x \in \mathbb{R}^n$ such that

$$0 \leqslant x \perp (Mx+q) \geqslant 0. \tag{1.1}$$

This means that the sought x must satisfy $x \geqslant 0$, $Mx+q \geqslant 0$ (vectorial inequalities must be understood componentwise), and $x^{\mathsf{T}}(Mx+q) = 0$ (the exponent "$^{\mathsf{T}}$" is used to denote matrix transposition). The problem has a combinatorial aspect, which lies in this last equation, since, by the nonnegativity of x and $Mx+q$, it amounts to the set of n complementarity conditions $x_i(Mx+q)_i = 0$ for all indices $i \in [1:n]$. The term complementarity comes from the fact that, for all $i \in [1:n]$, either x_i or $(Mx+q)_i$ must vanish; these conditions may be realized in 2^n different ways. Actually, the problem of determining whether a particular instance of the LCP has a solution is strongly NP-complete [14], and NP-complete for a $\mathbf{P_0}$-matrix (i.e., when M has nonegative principal minors) [29].

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be the *min-function* associated with the LCP (1.1), which is the function that takes at $x \in \mathbb{R}^n$ the value

$$F(x) = \min(x, Mx+q). \tag{1.2}$$

The Newton-min algorithm can be viewed as an instance of the semismooth Newton method [40, 31] to solve the equational equivalent form of (1.1) [32, 33, 16] that reads $F(x) = 0$. To write compactly the algorithm, it is useful to introduce, for $I \subseteq [1:n]$ and its complement $A := [1:n] \setminus I$, the point $x^{(I)}$ defined by

$$x^{(I)}_A = 0 \qquad \text{and} \qquad (Mx^{(I)}+q)_I = 0.$$

This point is well defined when M is *nondegenerate*, meaning that the principal minors of M do not vanish. The plain *Newton-min algorithm* computes the next iterate by

$$\hat{x} := x^{(\mathcal{S}(x))}, \tag{1.3}$$

where the *index selector* $\mathcal{S} : \mathbb{R}^n \multimap [1:n]$ is the multifunction defined at $x \in \mathbb{R}^n$ by

$$\mathcal{S}(x) := \{i \in [1:n] : x_i > (Mx+q)_i\}. \tag{1.4}$$

In some versions of the algorithm, $\mathcal{S}(x)$ also contains some or all the indices in $\{i \in [1:n] : x_i = (Mx+q)_i\}$. See paragraph 7 of the introduction of [4] for more details on the origin of this algorithm and a discussion on the contributions from [12, 31, 23, 22, 8, 7, 24, 28]. When the current iterate $x \in \mathbb{R}^n$ is not on a kink of F, like in this paper, the Newton-min algorithm is identical to the Newton method to find a zero of F, which is then well defined.

Even though the Newton-min algorithm uses no globalization technique, like line searches or trust regions [9, 15], it may converge globally, i.e., from any starting point. This is due to the very particular piecewise linearity of F. For example, global convergence occurs, whatever q is, when M is an $\mathbf{M}$-matrix [1], which is a $\mathbf{P}$-matrix (i.e., with positive principal minors) with nonpositive off-diagonal elements. It also occurs when M is close enough to an **M**-matrix [24]. However, this global convergence property does not extend up to the larger class of **P**-matrices [4, 5, 17, 6]. This is unfortunate for the Newton-min algorithm, since **P**-matrices are exactly those ensuring the existence and uniqueness of the solution to the LCP, whatever q is [41, 16].

A natural idea to enlarge the class of matrices, for which the global convergence of the Newton-min algorithm can be guaranteed, is to introduce a line search on the associated *least-square merit function*, which is the function $\Theta : \mathbb{R}^n \to \mathbb{R}$ defined at $x \in \mathbb{R}^n$ by

$$\Theta(x) = \frac{1}{2}\|F(x)\|^2,$$

where $\|\cdot\|$ denotes the Euclidean norm. This least-square function is natural, since it has been used, often with success, for globalizing the Newton method when the function F is smooth [18, 15, 9, 26]. In the presence of nonsmoothness of F, like here, this technique is more difficult to implement since the Newton-min direction $d := \hat{x} - x$ may not be a descent direction of Θ at a kink of F [20] (this fact was already observed during the PhD thesis of I. Ben Gharbia [3; example 5.8]). To overcome this difficulty, Harker and Pang proposed in [23; p. 275] a method named the *Modified Damped-Newton Algorithm*, which consists in taking for the next iterate the point

$$x^+ := x + (\breve{\alpha}_1 + \varepsilon_{\text{HP}})d,$$

where $\breve{\alpha}_1 > 0$ is a stepsize so that $x + \breve{\alpha}_1 d$ is on the first kink of F encountered along d from x, and $\varepsilon_{\text{HP}} > 0$ is a number such that the new iterate x^+ is not on a kink of F and ensures a sufficient decrease of the least-square merit function Θ. Consequently, this algorithm avoids the points of nondifferentiability of F, generates descent directions of Θ, and forces Θ to decrease sufficiently at each iteration. To the best of our knowledge, the only convergence result for any line search algorithm using a semismooth Newton direction uses the assumption that $\liminf_k \alpha_k > 0$ [38], which is a very weak result since this strong assumption relates to the algorithm products rather than the problem's data.

In this research field, sparing of theoretical results, this paper provides the value n as a worse case lower bound on the number of iterations of the Harker and Pang algorithm when the extra stepsize ε_{HP} is taken sufficiently small, which is allowed by the description of the method given in [23; p. 275]. This lower bound is obtained on the Fathi problem for a set of starting points, including the one of [21], which is zero. To extend the applicability of this result, we describe an algorithmic scheme, for which this worse case lower bound is valid; a scheme that includes the Harker and Pang algorithm for sufficiently small positive ε_{HP}. In this scheme, the iterates avoid the kinks of F and the stepsizes are chosen arbitrarily between the first two break-stepsizes $\breve{\alpha}_1$ and $\breve{\alpha}_2$ (to be defined). Now, on many practical problems, an algorithm using the Newton-min direction and a stepsize that is not forced to be in $(\breve{\alpha}_1, \breve{\alpha}_2)$ usually finds a solution in much fewer iterations than n; in the experiments of [20], it is not uncommon to encounter LCPs having up to 10^5 variables that are solved in fewer than 10 iterations. Nevertheless, the Fathi problem remains a difficult instance of LCP for this family of methods, independently of the chosen stepsizes. To illustrate this, we show in the numerical experiment section that, surprisingly, doing exact line searches hardly modifies the iteration counter. Finally, this worse case lower bound and the numerical experiments of section 5 suggest us that it is unlikely that the improvement of the Newton-min algorithm can lie in a better determination of the stepsizes. This observation paves the way for the proposals made in [20].

To conclude this introduction, let us mention that there are a large number of contributions related to the complexity of algorithms for solving the LCP. Most of them are related to interior point methods and it is out of the scope of this paper to review them (they can be found by looking at those citing one of the first accounts on the subject, which is [29, 30]). Other approaches are sometimes qualified as noninterior path-following/continuation methods and are based on the smoothing of equational versions of the LCP: the function $(a, b) \in \mathbb{R}^2 \mapsto a + b - [(a-b)^2 + 4\mu^2]^{1/2}$ is considered in [13, 27] and the smooth Fisher-Burmeister function $(a, b) \in \mathbb{R}^2 \mapsto a + b - [a^2 + b^2 + 2\mu^2]^{1/2}$ is used in [27]. The complexity of these approaches have been studied in [10, 25, 11], for instance.

The paper is structured as follows. The algorithmic scheme, for which the lower bound on the iterative complexity is obtained, is presented in section 2. The Fathi problem and two properties of its matrix are given in section 3. The iterative complexity result is proved in section 4. Finally, some numerical experiments are reported in section 5 and the paper ends with the conclusion section 6.

Notation. For the $n \times n$ matrix M and index sets I and $J \subseteq [1:n]$, we denote by M_{IJ} the submatrix of M formed of its elements with row indices in I and column indices in J. We also define $M_{I:} := M_{I[1:n]}$ and $M_{II}^{-1} := (M_{II})^{-1}$.

2 The Newton-min-HP-ext algorithmic scheme

In [23; 1990, p. 275], Harker and Pang proposed a method to solve the LCP (1.1) that they named the *Modified Damped-Newton Algorithm*. It is grounded on Newton's iterations to find a zero of the function F defined in (1.2), and it is first recalled as algorithm 2.4 below. Next, we describe an algorithmic scheme (algorithm 2.5), slightly extending the Harker and Pang algorithm, with the goal of making it a framework encompassing more ways of determining the stepsizes, in particular the one of Harker and Pang. It is for this last scheme that the lower bound on the iterative complexity is established.

The concepts of break-stepsizes and break-points will play a major part in the considered algorithms. After the definition of these notions, we clarify their connection with the nondifferentiability of F.

Definitions 2.1 (break-stepsize and break-point) A *break-stepsize* at $x \in \mathbb{R}^n$ along a direction $d \in \mathbb{R}^n$ is a real number $\breve{\alpha} > 0$ such that there is an index $i \in [1:n]$ for which $x_i \neq (Mx+q)_i$ and $(x+\breve{\alpha}d)_i = (Mx+q+\breve{\alpha}Md)_i$. Then, $\breve{x} := x+\breve{\alpha}d$ is called a *break-point*.

Lemma 2.2 (kink of F at a break-point) *Let $\breve{\alpha}$ be a break-stepsize at x along the direction d. Then F is not differentiable at $x + \breve{\alpha}d$.*

PROOF. Denote by $\breve{x} := x + \breve{\alpha}d$ the break-point corresponding to $\breve{\alpha}$. Since $\breve{\alpha}$ is a break-stepsize, there is an index $i \in [1:n]$ such that $x_i \neq (Mx + q)_i$ and $\breve{x}_i = (M\breve{x} + q)_i$, which implies that $d_i \neq (Md)_i$. Now an easy computation provides (see also [38])

$$F_i'(\breve{x}; d) = \min(d_i, (Md)_i) \quad \text{and} \quad F_i'(\breve{x}; -d) = \min(-d_i, -(Md)_i),$$

so that

$$F_i'(\check{x};d) + F_i'(\check{x};-d) = \min(d_i,(Md)_i) - \max(d_i,(Md)_i) < 0,$$

because $d_i \neq (Md)_i$. Hence F is nondifferentiable at $\check{x}$. $\square$

Remark 2.3 Whilst F is nondifferentiable at a break-point, this is not necessary the case for Θ, as shown by the following example: $n = 1$, $M = 2$, $q = 0$, $x = -1$, and $d = 1$. Then $\check{\alpha} = 1$ is a break-stepsize because $-1 = x \neq Mx + q = -2$ and, for $\check{x} = x + \check{\alpha}d$, $\check{x} = M\check{x} + q = 0$. Since

$$F(x) = \begin{cases} 2x & \text{if } x \leqslant 0 \\ x & \text{otherwise} \end{cases} \qquad \text{and} \qquad \Theta(x) = \begin{cases} 2x^2 & \text{if } x \leqslant 0 \\ \frac{1}{2}x^2 & \text{otherwise,} \end{cases}$$

we see that F is nondifferentiable at $\check{x} = 0$, but that Θ is differentiable at the same point. This is in agreement with the strong Fréchet differentiability of Θ at a zero of F, proved in [39; prop. 1]. $\square$

This paper deals with the Newton-min algorithm [1], which is now described with more precision than in the introduction. The method is similar to the one of Kojima and Shindo [31] and has the flavor of a semismooth Newton method [40] for finding a zero of the nonsmooth function F defined by (1.2) [24]. At a point $x \in \mathbb{R}^n$, the indices in $[1\,{:}\,n]$ are partitioned into three subsets:

$$\begin{aligned} A_0(x) &:= \{i \in [1\,{:}\,n] : x_i < (Mx+q)_i\}, \\ E(x) &:= \{i \in [1\,{:}\,n] : x_i = (Mx+q)_i\}, \\ I_0(x) &:= \{i \in [1\,{:}\,n] : x_i > (Mx+q)_i\}. \end{aligned}$$

Since, for $i \in A_0(x) \cup I_0(x)$, F_i is differentiable at x, a Newton-like direction d should satisfy $F_i'(x)d = -F_i(x)$, which becomes $d_i = -x_i$ for $i \in A_0(x)$ and $M_{i:}d = -(Mx+q)_i$ for $i \in I_0(x)$, where $M_{i:}$ denotes the ith row of M. For $i \in E(x)$, F_i is usually nonsmooth at x; to reduce the size of the linear system to solve, these indices are dealt with like those in $A_0(x)$. In summary, the following index sets are introduced

$$A \equiv A(x) := A_0(x) \cup E(x), \qquad I \equiv I(x) := I_0(x), \tag{2.1}$$

and the *Newton-min direction* is defined by

$$d_A = -x_A \qquad \text{and} \qquad M_{I:}d = -(Mx+q)_I \equiv -M_{I:}x - q_I. \tag{2.2}$$

As a result, the point $\hat{x} := x + d$ targeted by the Newton-min algorithm satisfies

$$\hat{x}_A = 0 \qquad \text{and} \qquad (M\hat{x}+q)_I = 0. \tag{2.3}$$

The target point $\hat{x}$ is the one introduced by (1.3), since $\mathcal{S}(x) = I$ with the previous notation. The system (2.3) has a unique solution when M is nondegenerate, since its second identity also reads $M_{II}\hat{x}_I = -q_I$, which determines $\hat{x}_I = -M_{II}^{-1}q_I$ since then M_{II} is nonsingular.

The *plain Newton-min algorithm*, which takes $x^+ := \hat{x}$ as the iterate following the current one x, converges locally in one iteration when M is nondegenerate and x is in some neighborhood of a solution to the LCP [22]. It also converges globally if M is an **M**-matrix [1], but not if M is a **P**-matrix, since there are counter-examples in that case [4, 5, 6] (and even when M is a symmetric positive definite matrix [17]).

The purpose of the Harker and Pang algorithm [23; 1990, p. 275] is to improve the convergence properties of the plain Newton-min algorithm, as already mentioned in the introduction. For this, a stepsize $\alpha > 0$ is introduced along the Newton-min direction d, meaning that the iterate x^+ following x is computed by

$$x^+ = x + \alpha d.$$

The stepsize α has the very particular form

$$\alpha = \check{\alpha}_1 + \varepsilon_{\text{HP}},$$

where $\check{\alpha}_1$ is the first break-stepsize in $(0,1)$ at x along d and $\varepsilon_{\text{HP}} > 0$ is a positive number such that x^+ is not a break-point of F and $\Theta(x^+)$ is sufficienty smaller than $\Theta(x)$, in the sense that

$$\Theta(x^+) \leqslant (1 - 2\omega\alpha)\Theta(x), \tag{2.4}$$

for some $\omega \in (0, 1/2)$. Since, when $E(x) = \varnothing$ (this condition is satisfied recursively by all the iterates of the algorithm), the directional derivative of Θ at x along the Newton-min direction d takes the value $\Theta'(x; d) = -2\Theta(x)$, the previous inequality corresponds to the usual Armijo condition [2, 9]. This algorithm is summarized below. To the best of our knowledge, its global convergence has not been proved.

Algorithm 2.4 (Newton-min-HP algorithm) It is supposed that the current iterate x is not a solution to (1.1) and verifies $E(x) = \varnothing$. The next iterate x^+ also verifies $E(x^+) = \varnothing$ and is computed as follows.

1. *Index sets.* Compute A and I by (2.1).
2. *Direction.* Compute the direction d by (2.2).
3. *Stepsize.* Compute the smallest break-stepsize $\check{\alpha}_1$, if any. Then, determine the stepsize $\alpha > 0$ by the following rules.
   - 3.1. If there is no break-stepsize in $(0,1)$, take $\alpha = 1$ and terminate with $x + d$,
   - 3.2. Otherwise take $\alpha = \check{\alpha}_1 + \varepsilon_{\text{HP}}$, where $\varepsilon_{\text{HP}} > 0$ is such that
     - 3.2.1. α is not a break-stepsize,
     - 3.2.2. (2.4) holds.
4. *New iterate.* $x^+ = x + \alpha d$.

It is not difficult to see that if the condition in step 3.1 holds, $\hat{x} := x + d$ is a solution to (1.1), which justifies the termination. This is because the inequalities verified by x are preserved at $\hat{x}$, since there is no break-point in the open segment $(x, \hat{x})$:

$$\hat{x}_A \leqslant (M\hat{x} + q)_A \qquad \text{and} \qquad \hat{x}_I \geqslant (M\hat{x} + q)_I. \tag{2.5}$$

Now, by (2.3), $\hat{x}_A = 0$ and $(M\hat{x} + q)_I = 0$, so that $0 \leqslant \hat{x} \perp (M\hat{x} + q) \geqslant 0$ (we have used (2.5) and $A \cup I = [1\,{:}\,n]$), meaning that $\hat{x}$ is a solution to the LCP.

The next algorithm is the one that is studied in section 4. It differs from algorithm 2.4 by the way the stepsizes are determined along the Newton-min direction. Our goal in the design of algorithm 2.5 is not to make it efficient, but to make it as little binding as possible, in order to include as many variants of the Newton-min algorithm as possible. This way, the lower bound on its iterative complexity given in proposition 4.4 below will be valid for all the algorithms obeying the rules of algorithm 2.5.

Algorithm 2.5 (Newton-min-HP-ext scheme) It is supposed that the current iterate x is not a solution to (1.1) and verifies $E(x) = \varnothing$. The next iterate x^+ is then computed as follows.

1. *Index sets.* Compute A and I by (2.1).
2. *Direction.* Compute the direction d by (2.2).
3. *Stepsize.* Compute the two smallest distinct break-stepsizes $\breve{\alpha}_1$ and $\breve{\alpha}_2$, if any. Then, determine the stepsize $\alpha > 0$ by the following rules.
   - 3.1. If there is no break-stepsize in $(0, 1)$, take $\alpha = 1$ and terminate with $x + d$,
   - 3.2. If there is a single break-stepsize $\breve{\alpha}_1$ in $(0, 1)$, take α in $(\breve{\alpha}_1, 1]$,
   - 3.3. If there are at least two break-stepsizes $\breve{\alpha}_1$ and $\breve{\alpha}_2$ in $(0, 1)$, take α in $(\breve{\alpha}_1, \breve{\alpha}_2)$.
4. *New iterate.* $x^+ = x + \alpha d$.

Note that, in general, the Newton-min-HP algorithm is not a particular instance of algorithm 2.5, because it may occur that $\breve{\alpha}_1 + \varepsilon_{\text{HP}} > \breve{\alpha}_2$. Nevertheless, the scheme 2.5 includes the Newton-min-HP algorithm when $\varepsilon_{\text{HP}} > 0$ is sufficiently small and convergence of the iterates to a solution occurs. Indeed, when convergence occurs, it occurs in a finite number of iterations (by the above mentioned convergence in one step when the current iterate is in some neighborhood of a solution). Then the smallest value of $\breve{\alpha}_2 - \breve{\alpha}_1$ encoutered along the iterations (when both $\breve{\alpha}_1$ and $\breve{\alpha}_2$ exist) is > 0, implying that a sufficiently small positive ε_{HP} is in $(0, \breve{\alpha}_2 - \breve{\alpha}_1)$ or $\breve{\alpha}_1 + \varepsilon_{\text{HP}} \in (\breve{\alpha}_1, \breve{\alpha}_2)$.

3 The Fathi problem

As claimed in the abstract, the lower bound on the iterative complexity of the Newton-min-HP-ext scheme is shown thanks to the Fathi problem. This LCP has its matrix formed with the one of the Murty LCP, which is first presented.

The Murty problem

The Murty problem [36] is often considered to have the following data M and q, and starting point x:

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & \cdots & 0 \\ 2 & 2 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ 2 & 2 & 2 & & 1 \end{pmatrix}, \qquad q = -e, \qquad \text{and} \qquad x = 0, \tag{3.1}$$

where e is the vector of all ones. Other values of q are considered in [37; chapter 6]. The matrix M is clearly a **P**-matrix (its principal minors have the value 1), so that the problem has a unique solution, which is $\bar{x} = (1, 0, \dots, 0)$. This problem is extensively used for assessing algorithms [34, 23, 13], probably because some pivoting methods [35] are known to require an exponential number of iterations to solve it [37; theorem 6.4]. This problem is also relatively difficult to solve for the Newton-min algorithm, but not with the same severity [20].

The Fathi problem

In the Fathi problem [21; 1979], M, q, and the starting point are given by

$$M = \begin{pmatrix} 1 & 2 & 2 & 2 & \cdots & 2 \\ 2 & 5 & 6 & 6 & \cdots & 6 \\ 2 & 6 & 9 & 10 & \cdots & 10 \\ 2 & 6 & 10 & 13 & & 14 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 2 & 6 & 10 & 14 & \cdots & 4(n-1)+1 \end{pmatrix}, \qquad q = -e, \qquad \text{and} \qquad x = 0. \tag{3.2}$$

Since $M = LL^{\mathsf{T}}$, where L is the nonsingular lower triangular Murty matrix [36], M is symmetric positive definite, hence a **P**-matrix. The unique solution to the Fathi problem is the same one as for the Murty problem, namely $\bar{x} = (1, 0, \dots, 0)$. This problem was introduced in [21] to show the exponential iterative complexity of some pivot algorithms when the matrix of the LCP is symmetric positive definite.

The analysis of the Newton-min-HP-ext scheme in section 4 relies on the following two technical properties of the Fathi matrix. The first property determines the vector $v_I := M_{II}^{-1} e_I$, for some $I \subseteq [1\!:\!n]$, which, according to (2.3) and $q = -e$, are the nonzero components of the point $\hat{x}$ targeted by the Newton-min algorithm at any point x in $\{x \in \mathbb{R}^n : x_A < (Mx+q)_A, \text{ and } x_I > (Mx+q)_I\}$, where A and I form a partition of $[1\!:\!n]$. In this lemma, the indices of the vector v_I are numbered with the indices in I. A similar convention is adopted for the matrices M_{II} and M_{AI}, where A is some other index subset of $[1\!:\!n]$.

Lemma 3.1 (two properties of the Fathi matrix) *Let $A := [2\!:\!k]$ for some $k \in [1\!:\!n]$ ($A = \varnothing$ if $k = 1$), $I := [1\!:\!n] \setminus A$, and e_A and e_I be the vectors of all ones, with indices taken in A and I respectively. Let M be the Fathi matrix given in (3.2). Then,*

1) *$v_I := M_{II}^{-1} e_I$ has its components, numbered by the indices of I, given by*

$$v_i = \begin{cases} \frac{2(2k-1)(n-k)+1}{4(k-1)(n-k)+1} & \text{if } i = 1, \\ (-1)^{i-k}\, \frac{2(n-i)+1}{4(k-1)(n-k)+1} & \text{if } i \in [k+1\!:\!n], \end{cases} \tag{3.3}$$

2) *$M_{AI} M_{II}^{-1} e_I > e_A$, when $A \neq \varnothing$.*

Proof. 1) We provide a short verification proof. Let us denote by L the lower triangular matrix of Murty of dimension n, denoted M in (3.1), so that $M_{II} = L_{I:}(L_{I:})^{\mathsf{T}}$. We only have to check that the vector v_I given by formula (3.3) satisfies $L_{I:}(L_{I:})^{\mathsf{T}} v_I = e_I$.

Let us simplify the notation by introducing the positive numbers

$$a := \frac{2(n-k)}{4(k-1)(n-k)+1} \qquad \text{and} \qquad b := \frac{1}{4(k-1)(n-k)+1}$$

and let us show that $w := (L_{I:})^{\mathsf{T}} v_I$ takes the value (the numbers on the right are the indices of the vector w)

$$w := \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ & 2 & 2 & \cdots & 2 \\ & \vdots & \vdots & \ddots & \vdots \\ & 2 & 2 & \cdots & 2 \\ & 1 & 2 & \ddots & \vdots \\ & & 1 & \ddots & 2 \\ & & & \ddots & 2 \\ & & & & 1 \end{pmatrix} v_I = \begin{pmatrix} 1 \\ -a \\ \vdots \\ -a \\ -b \\ b \\ \vdots \\ (-1)^{n-k} b \end{pmatrix}. \qquad \begin{array}{c} 1 \\ 2 \\ \vdots \\ k \\ k+1 \\ k+2 \\ \vdots \\ n \end{array} \tag{3.4}$$

We will use the fact that, for $p \in \mathbb{N}$, there holds

$$1 - 3 + 5 - 7 + 9 + \cdots + (-1)^p (2p+1) = (-1)^p (p+1). \tag{3.5}$$

Let us first compute, for $j \in [k+1:n]$, the sum

$$\begin{aligned} \textstyle\sum_{i=j}^n v_i &= b \textstyle\sum_{i=j}^n (-1)^{i-k} \big[2(n-i)+1\big] \qquad [(3.3)] \\ &= b(-1)^{k-n} \textstyle\sum_{i=j}^n (-1)^{n-i} \big[2(n-i)+1\big] \\ &= b(-1)^{k-n} \big[1 - 3 + 5 - 7 + 9 + \cdots + (-1)^{n-j}[2(n-j)+1]\big] \\ &= b(-1)^{k-n} \big[(-1)^{n-j}(n+1-j)\big] \qquad [(3.5) \text{ with } p = n-j] \\ &= (-1)^{j-k} b(n+1-j). \end{aligned} \tag{3.6}$$

The rows of (3.4) with index in $[2:k]$ now follow from the previous computation with $j = k+1$, since

$$2 \textstyle\sum_{i=k+1}^n v_i = -2b(n-k) = -a. \tag{3.7}$$

The first row of (3.4) is also verified since by the previous computation

$$v_1 + 2 \textstyle\sum_{i=k+1}^n v_i = \big[2(2k-1)(n-k)+1\big] b - a = 1.$$

The last $(n-k)$ rows of (3.4) are also verified since for $j \in [k+1:n]$ there holds, using (3.3) and (3.6):

$$\begin{aligned} v_j + 2 \textstyle\sum_{i=j+1}^n v_i &= (-1)^{j-k} \left(2(n-j)+1\right) b + (-1)^{j+1-k} 2b(n-j) \\ &= (-1)^{j-k} b \; [2(n-j) + 1 - 2(n-j)] \\ &= (-1)^{j-k} b. \end{aligned}$$

It remains to observe that $L_{I:}w = e_I$, which follows from

$$L_{I:}w = \begin{pmatrix} 1 &&&&&&& \\ 2 & 2 & \cdots & 2 & 1 &&& \\ 2 & 2 & \cdots & 2 & 2 & 1 && \\ \vdots & \vdots & \ddots & & \ddots & \ddots & \ddots & \\ 2 & 2 & \cdots & 2 & \cdots & 2 & 2 & 1 \end{pmatrix} w$$

$$= \begin{pmatrix} 1 \\ 2-2(k-1)a-b \\ 2-2(k-1)a-2b+b \\ \vdots \\ 2-2(k-1)a-2b+2b+\cdots+(-1)^{n-k}b \end{pmatrix} = e_I,$$

since $2(k-1)a+b=1$.

2) By the definitions of A and I, when $A \neq \varnothing$, M_{AI} has the form (the numbers on the right are the row indices of M_{AI})

$$\begin{array}{ll} M_{AI} = \begin{pmatrix} 2 & 6 & \cdots & 6 \\ 2 & 10 & \cdots & 10 \\ 2 & 14 & \cdots & 14 \\ \vdots & \vdots & & \vdots \\ 2 & 4k-2 & \cdots & 4k-2 \end{pmatrix}. & \qquad \begin{matrix} 2 \\ 3 \\ 4 \\ \vdots \\ k \end{matrix} \end{array}$$

Note that only the first column is present when $k = n$. Since $M_{II}^{-1}e_I$ is the vector v_I given by (3.3), the row with index $i \in [2:k]$ of $M_{AI}M_{II}^{-1}e_I$ reads

$$2v_1 + (4i-2)\sum_{j=k+1}^{n} v_j = 2v_1 - (4i-2)(n-k)b,$$

where we have used (3.6) with $j = k+1$ (see also (3.7)). Its smallest value is obtained for the largest i, that is $i = k$, and is, thanks to (3.3):

$$\begin{aligned} 2v_1 - (4k-2)(n-k)b &= 2[2(2k-1)(n-k)+1]b - (4k-2)(n-k)b \\ &= [(4k-2)(n-k)+2]b \\ &> 1, \end{aligned}$$

which is the stated result. $\square$

4 A lower bound on the iterative complexity

The goal of this section is to show that the Newton-min-HP-ext scheme (algorithm 2.5) converges in exactly n iterations on the instance of dimension n of the Fathi problem (3.2) when the algorithm starts at zero or in some neighborhood of zero. This gives a lower bound on the iterative complexity of the considered algorithmic scheme.

The proof of proposition 4.4 below consists in showing that, when the Newton-min-HP-ext scheme generates a sequence $\{x^k\}_{k\geqslant 0}$ with a starting point x^0 near zero (in the set $\mathcal{X}_1$ introduced below actually), there holds $x^k \in \mathcal{X}_{k+1}$, for $k \in [1:n]$, where $\mathcal{X}_k$ is defined by

$$\mathcal{X}_k := \Big\{x \in \mathbb{R}^n : \; x_{A_k} < (Mx+q)_{A_k} \text{ and } x_{I_k} > (Mx+q)_{I_k}, \\ \tfrac{(Mx-e-x)_i}{(Mx-e-x)_{i+2}} < \tfrac{(M^{-1}_{I_kI_k}e_{I_k})_i}{(M^{-1}_{I_kI_k}e_{I_k})_{i+2}}, \text{ for all } i \in [k+1:n-2]\Big\}, \tag{4.1}$$

with $A_k = [2:k]$ and $I_k := [2:k]^c$ (the complementary set of $[2:k]$ in $[1:n]$). In this definition, it is assumed that the integer interval $[i:j]$ is empty when $j < i$ (in particular, $A_1 = \varnothing$ and the strict inequalities after the second one are not present if $k \geqslant n-2$).

Remarks 4.1 1) There holds $0 \in \mathcal{X}_1$. Indeed, $A_1 = \varnothing$, $I_1 = [1:n]$, $0 > M0 + q = -e$ and, for $i \in [2:n-2]$:

$$\frac{(M0-e-0)_i}{(M0-e-0)_{i+2}} = 1 < \frac{2(n-i)+1}{2(n-i-2)+1} = \frac{(M^{-1}e)_i}{(M^{-1}e)_{i+2}},$$

where we have used (3.3). This observation also shows that $\mathcal{X}_1 \neq \varnothing$.

2) The fact that $\mathcal{X}_k \neq \varnothing$ for $k \in [2:n]$ will be a consequence of lemma 4.2 below.

3) The set $\mathcal{X}_n = \{x \in \mathbb{R}^n : x_{[2:n]} < (Mx+q)_{[2:n]} \text{ and } x_1 > (Mx+q)_1\}$ is the one to which belongs the solution to the Fathi problem, namely $\bar{x} = (1, 0, \dots, 0)$.

4) By the strict inequalities in their definition, the sets $\mathcal{X}_k$ are *open* (more precisely they are interiors of polyhedrons), so that they are not reduced to a single point. By the first two strict inequalities in their definition, these sets are also two by two *disjoint*.

5) The last group of inequalities in the definition (4.1) of $\mathcal{X}_k$ is essential for our analysis. Let $\mathcal{X}'_k := \{x \in \mathbb{R}^n : \; x_{A_k} < (Mx+q)_{A_k} \text{ and } x_{I_k} > (Mx+q)_{I_k}\}$ be the set $\mathcal{X}_k$ without these inequalities.

- A first observation is that the last group of inequalities in $\mathcal{X}_k$ is not redundant. For example, if $n = 4$, $x^0 := e/31$ belongs to $\mathcal{X}'_1$, since $x^0 > (Mx^0 + q)$, but not to $\mathcal{X}_1$.
- Another observation is that the iterate following one in $\mathcal{X}'_k$ is not necessarily in $\mathcal{X}'_{k+1}$, hence the usefulness of working with $\mathcal{X}_k$ instead of $\mathcal{X}'_k$. For example, again for $n = 4$, we have seen that $x^0 = e/31$ is in $\mathcal{X}'_1$, but the next iterate x^1 computed by our code implementing algorithm 2.5 satisfies $x^1_4 < (Mx^1+q)_4$ and $x^1_{\{1,2,3\}} > (Mx^1+q)_{\{1,2,3\}}$, so that x^1 is not in $\mathcal{X}'_2$. □

We start with the following fundamental lemma (fundamental for our purpose, since it contains the essential idea of the proof), which shows that if the current iterate x of the Newton-min-HP-ext scheme is in $\mathcal{X}_k$, the next iterate $x^+ = x + \alpha d$ will be in $\mathcal{X}_{k+1}$. In its proof, for positive integers i, s, and j, we use the notation

$$[i:s:j] := \{i, i+s, i+2s, \dots, i + \lfloor (j-i)/s \rfloor s\},$$

where $\lfloor\cdot\rfloor$ is the floor operator ($\lfloor r \rfloor$ is the integer number i such that r is in $[i, i+1)$); hence $[i:1:j] = [i:j]$.

Lemma 4.2 (one iteration from x to x^+) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Suppose that the current iterate x of the Newton-min-HP-ext scheme is in $\mathcal{X}_k$ for some $k \in [1:n-1]$. Then, the next iterate $x^+ = x + \alpha d$ is in $\mathcal{X}_{k+1}$ and, when $k \leqslant n-3$, the stepsize α is in $(0,1)$.*

Proof. Let $k \in [1:n-1]$, $x \in \mathcal{X}_k$, and set $A \equiv A_k := [2:k]$ and $I \equiv I_k := [1:n] \setminus A$, so that

$$x_A < (Mx+q)_A \qquad \text{and} \qquad x_I > (Mx+q)_I.$$

The next iterate is then defined by $x^+ := x + \alpha d$, where

$$d = \begin{pmatrix} 0_A \\ M_{II}^{-1} e_I \end{pmatrix} - x \tag{4.2}$$

and the stepsize α is chosen as described in step 3 of algorithm 2.5. We have to prove that for some $\breve{\alpha} \in (0,1)$ there hold

$$(x+td)_A < (Mx+q+tMd)_A, \quad \text{for all } t \in [0,\breve{\alpha}], \tag{4.3a}$$

$$(x+\breve{\alpha}d)_{k+1} = (Mx+q+\breve{\alpha}Md)_{k+1}, \tag{4.3b}$$

$$(x+td)_{I\setminus\{k+1\}} > (Mx+q+tMd)_{I\setminus\{k+1\}}, \quad \text{for all } t \in [0,\breve{\alpha}], \tag{4.3c}$$

$$\text{if } k \leqslant n-3, \text{ then } \alpha \in (\breve{\alpha},1), \tag{4.3d}$$

$$\tfrac{(Mx^+-e-x^+)_i}{(Mx^+-e-x^+)_{i+2}} < \tfrac{(M_{I^+I^+}^{-1}e_{I^+})_i}{(M_{I^+I^+}^{-1}e_{I^+})_{i+2}}, \quad \text{for all } i \in [k+2:n-2], \tag{4.3e}$$

where $I^+ := I_{k+1} = [1:n] \setminus [2:k+1]$. Indeed, if (4.3) is shown:

- by (4.3a)–(4.3c), the first break-stepsize $\breve{\alpha}_1$ is $\breve{\alpha} \in (0,1)$ and this break-stepsize is due to the index $k+1$,
- since $x \in \mathcal{X}_k$, it follows, using also (4.3b), that $(x+td)_{k+1} > (Mx+q+tMd)_{k+1}$ for $t < \breve{\alpha}$, so that the reverse inequality holds for $t > \breve{\alpha}$, implying that $k+1 \in A(x^+)$,
- since the stepsize α taken by algorithm 2.5 is less than the possible next break-stepsize $\breve{\alpha}_2 > \breve{\alpha}_1$, the inequalities (4.3a) and (4.3c) hold at $x+\alpha d = x^+$; hence $A(x^+) = [2, k+1]$ and $I(x^+) = [1:n] \setminus [2, k+1]$,
- now with (4.3e), x^+ is in $\mathcal{X}_{k+1}$.

This implies that the first two strict inequalities in the definition of $\mathcal{X}_{k+1}$ hold. The last group of inequalities is just (4.3e). Finally, (4.3d) shows indeed that $\alpha \in (0,1)$. Let us now prove (4.3).

The equality $(x+td)_i = (Mx+q+tMd)_i$ is equivalent to $t(d-Md)_i = (Mx+q-x)_i$ or, using (4.2) and the value of $q = -e$, this identity can be rewritten

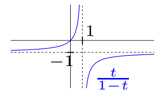
$$t\left[\begin{pmatrix} 0_A \\ M_{II}^{-1} e_I \end{pmatrix} - \begin{pmatrix} M_{AI} M_{II}^{-1} e_I \\ e_I \end{pmatrix} + Mx - x\right]_i = [Mx-e-x]_i\,. \tag{4.4}$$

Consider the indices $i \in A$. By (4.4), the equality $(x+td)_i = (Mx+q+tMd)_i$ is equivalent to

$$t\left[e_A - M_{AI}M_{II}^{-1}e_I + (Mx-e-x)_A\right]_i = [Mx-e-x]_i$$

or to

$$\frac{t}{1-t} = \frac{(Mx - e - x)_i}{(e_A - M_{AI} M_{II}^{-1} e_I)_i}.$$



Observe that the left-hand side is nonnegative if and only if $t \in [0, 1)$. Furthermore, the right-hand side is negative, since the numerator is positive by the assumption on x and the index $i \in A$, while the denominator is negative by point 2 of lemma 3.1. This implies that this identity cannot be realized by some $t \in [0, 1]$. Consequently

$$\forall\, t \in [0, 1]: \quad (x + td)_A < (Mx + q + tMd)_A$$

and (4.3a) holds, provided we show that $\breve{\alpha} \leqslant 1$.

Consider now the indices $i \in I$. By (4.4), the equality $(x + td)_i = (Mx + q + tMd)_i$ is now equivalent to

$$t\left[M_{II}^{-1} e_I + (Mx - e - x)_I\right]_i = [Mx - e - x]_i$$

or to

$$\frac{t}{1-t} = \frac{(Mx - e - x)_i}{(M_{II}^{-1} e_I)_i}. \tag{4.5}$$

For $i \in I$, the numerator of the fraction in the right-hand side is negative, so that the right-hand side is positive when $(M_{II}^{-1} e)_i$ is also negative, that is for $i \in [k+1:2:n]$ according to (3.3). By the monotonicity of the map $t \mapsto t/(1-t)$, the smallest break-stepsize at x along d is due to the index i giving the smallest fraction in the right-hand side. Since $x \in \mathcal{X}_k$, the third inequality in the definition (4.1) of $\mathcal{X}_k$ and the negativity of $(Mx - e - x)_i$ and $(M_{II}^{-1} e_I)_i$ for $i \in [k+1:2:n]$ tell us that this occurs for the smallest index $i \in [k+1:2:n]$, that is for $k+1$ (note that we use here only half of these third inequalities in the definition of $\mathcal{X}_k$; the others will be used below for getting (4.3e)). Therefore, we have shown (4.3b) and (4.3c) for the unique solution $\breve{\alpha}$ of

$$\frac{\breve{\alpha}}{1 - \breve{\alpha}} = \frac{(Mx - e - x)_{k+1}}{(M_{II}^{-1} e_I)_{k+1}},$$

which is in $(0, 1)$.

The reasonings in the previous two paragraphs also tell us that there are $\lceil (n-k)/2 \rceil$ break-stepsizes in the interval $(0, 1)$, which are due to the indices $[k+1:2:n]$. Therefore, when $k \leqslant n - 3$, there are at least two break-stepsizes in $(0, 1)$ and, by the step 3.3 of algorithm 2.5, there holds $\breve{\alpha}_1 < \alpha < \breve{\alpha}_2 < 1$, showing that α is in $(0, 1)$. This shows (4.3d).

We still have to prove (4.3e) at the next iterate $x^+ = x + \alpha d$, where the stepsize $\alpha > 0$ is determined in step 3 of algorithm 2.5. Observe first that, for $i \in [k+2:n-2] \subseteq I^+ \subseteq I$, lemma 3.1 ensures the following identity on the ratio in the right-hand side of (4.3e):

$$\frac{(M_{I^+I^+}^{-1} e_{I^+})_i}{(M_{I^+I^+}^{-1} e_{I^+})_{i+2}} = \frac{2(n-i)+1}{2(n-i-2)+1} = \frac{(M_{II}^{-1} e_I)_i}{(M_{II}^{-1} e_I)_{i+2}}.$$

Therefore, since x^+ is defined using $M_{II}^{-1} e_I$, instead of (4.3e), it is easier to prove the following equivalent inequality:

$$\frac{(Mx^+ - e - x^+)_i}{(Mx^+ - e - x^+)_{i+2}} < \frac{(M_{II}^{-1} e_I)_i}{(M_{II}^{-1} e_I)_{i+2}}, \quad \text{for all } i \in [k+2:n-2]. \tag{4.6}$$

Observe now that the numerators (and the denominators) in (4.6) are linked by

$$\begin{aligned}(Mx^+ - e - x^+)_I &= (Mx - e - x)_I + \alpha(Md - d)_I \\ &= (Mx - e - x)_I + \alpha(e_I - (Mx)_I - M_{II}^{-1}e_I + x_I) \qquad [(4.2)] \\ &= (1-\alpha)(Mx - e - x)_I - \alpha M_{II}^{-1}e_I. \end{aligned} \tag{4.7}$$

Take now $i \in [k+2 : n-2]$. Then $i \in I$, $k \leqslant n-4$, so that $\alpha \in (0,1)$ by (4.3d). Note also that $i+2 \in I$. Using (4.7), the quotient in the left-hand side of (4.6) becomes

$$\frac{(Mx^+ - e - x^+)_i}{(Mx^+ - e - x^+)_{i+2}} = \frac{-\alpha(M_{II}^{-1}e_I)_i + (1-\alpha)(Mx - e - x)_i}{-\alpha(M_{II}^{-1}e_I)_{i+2} + (1-\alpha)(Mx - e - x)_{i+2}}. \tag{4.8}$$

The quotient in the right-hand side of (4.8) can be written $\frac{a+s}{b+t}$ with the notation

$$\begin{aligned} a &:= -\alpha(M_{II}^{-1}e_I)_i, & s &:= (1-\alpha)(Mx - e - x)_i, \\ b &:= -\alpha(M_{II}^{-1}e_I)_{i+2}, & t &:= (1-\alpha)(Mx - e - x)_{i+2}. \end{aligned}$$

Observe that

- $t = (1-\alpha)(Mx - e - x)_{i+2} < 0$, because $\alpha < 1$ and $i+2 \in I$,
- $b + t = (Mx^+ - e - x^+)_{i+2} < 0$, because $i+2 \in I^+$ by (4.3a)-(4.3c),
- $\frac{s}{t} < \frac{a}{b}$, because

  $$\frac{s}{t} = \frac{(Mx - e - x)_i}{(Mx - e - x)_{i+2}} < \frac{(M_{II}^{-1}e_I)_i}{(M_{II}^{-1}e_I)_{i+2}} = \frac{a}{b},$$

  where the strict inequality comes from the fact that $x \in \mathcal{X}_k$ and $\{i, i+2\} \subseteq I$.

It follows that $\frac{a+s}{b+t} < \frac{a}{b}$. Therefore (4.8) becomes (4.6), from which (4.3e) follows directly. □

By the previous lemma, if the initial iterate x^0 belongs to $\mathcal{X}_1$, after $n-1$ iterations, the iterate x^{n-1} belongs to

$$\mathcal{X}_n := \{x \in \mathbb{R}^n : x_{[2:n]} < (Mx + q)_{[2:n]} \text{ and } x_1 > (Mx + q)_1\}, \tag{4.9}$$

to which the solution $\bar{x} = e^1$ belongs. Hence the question arises to know whether one can have $x^{n-1} = \bar{x}$ and therefore converge in $n-1$ iterations. The next lemma invalidates this possibility.

Lemma 4.3 ($x^{n-1} \neq \bar{x}$) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Then, algorithm 2.5, starting at a point $x \in \mathcal{X}_{n-1}$ finds a point $x^+ \in \mathcal{X}_n$ that differs from the solution $\bar{x} = e^1$ to the LCP problem (1.1).*

Proof. Let us simplify the notation by setting $A := [2 : n-1]$ and $I = \{1, n\}$. Then

$$\mathcal{X}_{n-1} = \{x \in \mathbb{R}^n : x_A < (Mx + q)_A \text{ and } x_I > (Mx + q)_I\}.$$

By algorithm 2.5, the iterate following $x \in \mathcal{X}_{n-1}$ satisfies

$$x^+ = x + \alpha((0_A, v_I) - x) = (1-\alpha)x + \alpha(0_A, v_I), \tag{4.10}$$

where v_I is given by (3.3) with the index set I introduced above (see the comment before lemma 3.1) and $\alpha > 0$ is the stepsize. We want to show that $x^+ \neq \bar{x}$.

We proceed by contradiction, assuming that $x^+ = \bar{x}$. Then, $x_1^+ = 1$, $x_A^+ = 0$, and $x_n^+ = 0$. According to (4.10) and (3.3) with $k = n-1$, the first and third conditions read

$$(1-\alpha)x_1 + \alpha\,\frac{4n-5}{4n-7} = 1 \qquad \text{and} \qquad (1-\alpha)x_n + \alpha\,\frac{-1}{4n-7} = 0. \tag{4.11}$$

By the second identity in (4.11),

$$\alpha \neq 1. \tag{4.12}$$

Then (4.10) and $x_A^+ = 0$ imply that

$$x_A = 0. \tag{4.13}$$

Furthermore, adding the first identity in (4.11) and twice the second yields $(1-\alpha)(x_1+2x_n) + \alpha = 1$, which, thanks to (4.12), implies that

$$x_1 + 2x_n = 1. \tag{4.14}$$

Now, since $x \in \mathcal{X}_{n-1}$, there hold $x_1 > (Mx+q)_1$ and $x_n > (Mx+q)_n$. Therefore

$$\begin{aligned} x_1 + x_n &> (Mx+q)_1 + (Mx+q)_n \\ &= \big[x_1 + 2x_n - 1\big] + \big[2x_1 + (4n-3)x_n - 1\big] \qquad [(3.2) \text{ and } (4.13)] \\ &= 3x_1 + (4n-1)x_n - 2 \end{aligned}$$

or

$$(x_1 + 2x_n) + (2n-3)x_n < 1.$$

Using (4.14) and $n \geqslant 2$, we get $x_n < 0$, which is in contradiction with $\alpha \in [0,1]$ and the second identity in (4.11). $\square$

The restriction on $n \geqslant 2$ in lemma 4.3 is necessary, since when $n = 1$ the set $\mathcal{X}_{n-1}$ appearing in its statement does not exist.

Proposition 4.4 (worse case lower bound of the Newton-min-HP-ext scheme) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Then, algorithm 2.5, starting at a point $x \in \mathcal{X}_k$, for some $k \in [1\!:\!n-1]$, finds the solution to the problem in exactly $n-k+1$ iterations. In particular, when started at $x \in \mathcal{X}_1$ or at $x = 0$, algorithm 2.5 finds the solution in exactly n iterations.*

Proof. The first claim comes from the fact that in one iteration the algorithm finds a point in $\mathcal{X}_{k+1}$ (by lemma 4.2). Applying this argument repetitively, we see that the algorithm finds a point on $\mathcal{X}_{n-1}$ in $n-k-1$ iterations. By lemma 4.3, the algorithm finds next a point in $\mathcal{X}_n$ in one more iteration, but this point is not the solution. Hence, one more iteration is necessary to get the solution and this is what algorithm 2.5 does. Indeed, if an iterate $x \in \mathcal{X}_n$, then there holds $x_{[2:n]} < (Mx+q)_{[2:n]}$ and $x_1 > (Mx+q)_1$ by (4.9), so that the next iterate x^+ satisfies $x^+_{[2:n]} = 0$ and $x_1^+ = 1$ if a unit stepsize is taken, which is indeed the choice of the algorithm. Hence x^+ is the solution.

The second claim can be deduced from the first claim with $k = 1$ and the use of the fact that $0 \in \mathcal{X}_1$ (see remark 4.1(1)). $\square$

Proposition 4.4 is not valid for $n = 1$. Indeed, in that case, $\mathcal{X}_1 = \mathcal{X}_n = \mathbb{R}$ and an initial iterate $x^0 \in \mathcal{X}_1$ can be the solution $\bar{x} = 1$, hence requiring no iteration to converge.

Corollary 4.5 (worse case lower bound of the Newton-min-HP algorithm) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Then, algorithm 2.4 with $\varepsilon_{\text{HP}} > 0$ sufficiently small, starting at a point $x \in \mathcal{X}_k$, for some $k \in [1:n-1]$, finds the solution to the problem in exactly $n-k+1$ iterations. In particular, when started at $x \in \mathcal{X}_1$ or at $x = 0$, algorithm 2.4 finds the solution in exactly n iterations.*

Proof. This is because, when $\varepsilon_{\text{HP}} > 0$ is sufficiently small, the stepsizes α are in $(\check{\alpha}_1, \check{\alpha}_2)$ (see the comment given after the statement of algorithm 2.5) and proposition 4.4 applies. $\square$

The behavior of algorithm 2.4 may be different from the one described in the previous corollary, when $\varepsilon_{\text{HP}} > 0$ is not taken sufficiently small and that the condition $\alpha \in (\check{\alpha}_1, \check{\alpha}_2)$ is not satisfied at some iterations. In particular, it could be more (or less) efficient. Nevertheless, the experiment of the next section suggests us that it could be much less efficient with a larger value of $\varepsilon_{\text{HP}} > 0$ (compare columns 3 and 4 in table 5.1).

5 Numerical experiments

We have written a piece of software in Matlab, called Nmhp [19], which implements 3 methods.

- (M_1) The first method is the Harker and Pang algorithm (algorithm 2.4), in which the extra stepsize $\varepsilon_{\text{HP}} > 0$ is determined from an initial value $\varepsilon^0_{\text{HP}} > 0$ prescribed by the user. In the numerical experiments reported below, we have taken the latter small ($\varepsilon^0_{\text{HP}} := 10^{-7}$ or 10^{-5}), while $\varepsilon_{\text{HP}} := \varepsilon^0_{\text{HP}}/2^i$, where i is the smallest nonnegative integer such that the two conditions in step 3.2 of algorithm 2.4 are satisfied. This is always possible since the number of break-stepsizes is finite and the Armijo condition (2.4) is satisfied with strict inequality for $\alpha = \check{\alpha}_1$ thanks to the choice of $\omega \in (0, 1/2)$.
- (M_2) The second method is the extended version of the Harker and Pang algorithm (algorithm 2.5), in which the stepsize is fixed to $\alpha = (\check{\alpha}_1 + \check{\alpha}_2)/2$. According to lemma 4.2 on algorithm 2.5, the results would not be modified on the Fathi problem by taking any stepsize in $(\check{\alpha}_1, \check{\alpha}_2)$.
- (M_3) The third method is a variant of the Newton-min algorithm with exact line search (meaning that $x^+ := x + \alpha d$ where $\alpha > 0$ is such that $\Theta(x^+) = \min\{\Theta(x + \alpha' d) : \alpha' > 0\}$). With exact line search, it is no longer guaranteed that $E(x) = \varnothing$ at all iterate x. This implies that a descent direction of Θ must be determined even when $E(x) \neq \varnothing$. We have chosen the Newton-min-hybrid direction defined in [20]. In this approach, an index i is chosen to be in $E(x)$ when $|x_i - (Mx + q)_i| \leqslant 10^{-11}$, it is in $A_0(x)$ when $x_i < (Mx + q)_i - 10^{-11}$, and in $I_0(x)$ when $x_i > (Mx + q)_i + 10^{-11}$.

These methods have been run on various instances of the Fathi problem, taking zero for initial iterate. The numbers of iterations are gathered in table 5.1, together with those given

	Number of iterations					
	Harker-Pang algorithm			Algorithm 2.5 in Nmhp	Exact line search
n	In [23]	Algorithm 2.4 in Nmhp with $\varepsilon_{\text{HP}}^0 = 10^{-7}$	$\varepsilon_{\text{HP}}^0 = 10^{-5}$		
8	8	8	8	8	8
16	16	16	16	16	16
32	32	32	32	32	32
64	65	64	64	64	64
128	63	128	128	128	128
256	-	256	256	256	256
512	-	512	524	512	513
1024	-	1024	6367	1024	1025
2048	-	2048	16337	2048	2049

Table 5.1: Comparison of the number of iterations required to solve the Fathi problem of dimension n (1st column) starting at zero by several algorithms: the 2nd column gives the results of Harker and Pang in [23], the 3rd and 4th column gives the results of our implementation in Nmhp of algorithm 2.4 with $\varepsilon_{\text{HP}}^0 = 10^{-7}$ and 10^{-5}, the 5th column are those of algorithm 2.5 in Nmhp, and the last column gives the results of the exact line search Newton-min-hybrid algorithm.

by Harker and Pang in [23; table 5, example 2]. The first column gives the dimension n of the Fathi problem.

Here are some observations on the reported statistics (see table 5.1).

(O_1) The results obtained by algorithm 2.5 of Nmhp (5th column) are in accordance with proposition 4.4: the number of iterations is n.

(O_2) The results given by Harker and Pang in [23] (2nd column) differ from n, for $n = 64$ and 128, and are not given for larger dimensions. The differences with algorithm 2.5 can only come from the stepsize $\alpha > 0$ taken along the Newton-min direction. The results of [23] for $n = 64$ and 128 could be explained by invoking rounding errors in the piece of software producing these results or, according to the proof of lemma 4.2, by the fact that $\breve{\alpha}_1 + \varepsilon_{\text{HP}} > \breve{\alpha}_2$ at some iterations when $n = 64$ and 128.

(O_3) Nevertheless, we have not been able to reproduce the results of Harker and Pang [23] with our implementation of algorithm 2.4: in accordance with corollary 4.5, when $\varepsilon_{\text{HP}}^0$ is sufficiently small one recovers the n iterations to find the solution ($\varepsilon_{\text{HP}}^0 = 10^{-7}$ is small enough for the considered dimensions, see the 3rd column in table 5.1), but when $\varepsilon_{\text{HP}}^0$ is larger, the number of iterations has a tendency to increase (this is the case for $\varepsilon_{\text{HP}}^0 = 10^{-5}$, see the 4th column in table 5.1), not to decrease as in the results of [23].

(O_4) The results obtained with the exact line search Newton-min-hybrid algorithm (last column) are surprising: the number of iterations differs from n by at most one unit. In other words, having a line search determining the best possible decrease of Θ does not improve the iteration counter (note that a modification of the stepsize changes the following direction). Proving this result would certainly be more difficult than the one shown in this paper, because the output of the code indicates that the change in the index sets (A, I) along the iterations does not follow the simple mechanism highlighted by lemma 4.2. Nevertheless, this last experiment supports the conclusion that any progress in the efficiency of the Newton-min is unlikely to come from a better line search procedure.

6 Conclusion

This paper is a contribution to the better understanding of the Newton-min algorithm with line search on the least-square merit function for solving the linear complementarity problem. It examines in detail the behavior of the Harker and Pang globalization of the algorithm on the Fathi problem. It is mathematically proved and numerically observed that, if the first iterate is in some open polyhedral neighborhood of zero, then the algorithm requires exactly n iterations to find the solution to the problem (n is the number of variables). Whilst this is not disastrous, for very large problems, it is not as attractive as the best path-following algorithms (interior or non-interior), whose iterative complexity is in $O(n^{1/2})$, and it does not reflect the excellent behavior of the Newton-min algorithm on many large-scale problems coming from concrete applications [20]. Nevertheless, the realized precise computation of the number of iterations for the Fathi problem provides a lower bound on the provable iterative complexity of the Harker and Pang version of the Newton-min algorithm with line search, on a class of problems containing the Fathi problems. Numerical experiments suggest that this worse case lower bound could also be valid if an exact line search is performed.

Acknowledgment

We have benefited from a careful reading of the manuscript by the referee, who is warmly thanked.

References

[1] M. Aganagić (1984). Newton's method for linear complementarity problems. *Mathematical Programming*, 28, 349–362. [doi]. 2, 5, 6

[2] L. Armijo (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16, 1–3. [journal]. 6

[3] I. Ben Gharbia (2012). *Résolution de Problèmes de Complémentarité – Application à un Écoulement Diphasique Dans un Milieu Poreux*. PhD Thesis, Université Paris-Dauphine. 3

[4] I. Ben Gharbia, J.Ch. Gilbert (2012). Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix. *Mathematical Programming*, 134, 349–364. [doi]. 2, 3, 6

[5] I. Ben Gharbia, J.Ch. Gilbert (2013). An algorithmic characterization of P-matricity. *SIAM Journal on Matrix Analysis and Applications*, 34(3), 904–916. [doi]. 3, 6

[6] I. Ben Gharbia, J.Ch. Gilbert (2019). An algorithmic characterization of P-matricity II: adjustments, refinements, and validation. *SIAM Journal on Matrix Analysis and Applications*, 40(2), 800–813. [doi]. 3, 6

[7] M. Bergounioux, M. Haddou, M. Hintermüller, K. Kunisch (2000). A comparison of a Moreau-Yosida-based active set strategy and interior point methods for constrained optimal control problems. *SIAM Journal on Optimization*, 11, 495–521. [doi]. 2

[8] M. Bergounioux, K. Ito, K. Kunisch (1999). Primal-dual strategy for constrained optimal control problems. *SIAM Journal on Control and Optimization*, 37, 1176–1194. [doi]. 2

[9] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, C. Sagastizábal (2006). *Numerical Optimization – Theoretical and Practical Aspects* (second edition). Universitext. Springer Verlag, Berlin. [authors] [editor] [doi]. 2, 3, 6

[10] J.V. Burke, S. Xu (1999). A polynomial time interior-point path-following algorithm for LCP based on Chen-Harker-Kanzow smoothing techniques. *Mathematical Programming*, 86(1), 91–103. [doi]. 4

[11] J.V. Burke, S. Xu (2002). Complexity of a noninterior path-following method for the linear complementarity problem. *Journal of Optimization Theory and Applications*, 112(1), 53–76. [doi]. 4

[12] R. Chandrasekaran (1970). A special case of the complementary pivot problem. *Opsearch*, 7, 263–268. 2

[13] B. Chen, P.T. Harker (1993). A non-interior continuation method for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 14, 1168–1190. [doi]. 4, 8

[14] S.J. Chung (1989). NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60, 393–399. [doi]. 2

[15] A.R. Conn, N.I.M. Gould, Ph.L. Toint (2000). *Trust-Region Methods*. MPS-SIAM Series on Optimization 1. SIAM and MPS, Philadelphia. [doi]. 2, 3

[16] R.W. Cottle, J.-S. Pang, R.E. Stone (2009). *The Linear Complementarity Problem*. Classics in Applied Mathematics 60. SIAM, Philadelphia, PA, USA. 2, 3

[17] F.E. Curtis, Z. Han, D.P. Robinson (2015). A globally convergent primal-dual active-set framework for large-scale convex quadratic optimization. *Computational Optimization and Applications*, 60(2), 311–341. [doi]. 3, 6

[18] J.E. Dennis, R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs. 3

[19] J.-P. Dussault, M. Frappier, J.Ch. Gilbert (2018, 6). Nmhp, version 1.0. https://who.rocq.inria.fr/Jean-Charles.Gilbert/codes/nmhp/nmhp.zip. 16

[20] J.P. Dussault, M. Frappier, J.Ch. Gilbert (2019). A globally convergent polyhedral Newton-min algorithm for solving a nonlinear complementarity problem. Technical report. (to appear). 3, 4, 8, 16, 18

[21] Y. Fathi (1979). Computational complexity of LCPs associated with positive definite symmetric matrices. *Mathematical Programming*, 17, 335–344. 3, 8

[22] A. Fischer, C. Kanzow (1996). On finite termination of an iterative method for linear complementarity problems. *Mathematical Programming*, 74, 279–292. [doi]. 2, 6

[23] P.T. Harker, J.-S. Pang (1990). A damped-Newton method for the linear complementarity problem. In E.L. Allgower, K. Georg (editors), *Computational Solution of Nonlinear Systems of Equations*, Lecture in Applied Mathematics 26. AMS, Providence, RI. 2, 3, 4, 6, 8, 17

[24] M. Hintermüller, K. Ito, K. Kunisch (2003). The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13, 865–888. [doi]. 2, 3, 5

[25] K. Hotta, M. Inaba, A. Yoshise (2000). A complexity analysis of a smoothing method using CHKS-functions for monotone linear complementarity problems. *Computational Optimization and Applications*, 17(2-3), 183–201. [doi]. 4

[26] A.F. Izmailov, M.V. Solodov (2014). *Newton-Type Methods for Optimization and Variational Problems.* Springer Series in Operations Research and Financial Engineering. Springer. 3

[27] Ch. Kanzow (1996). Some noninterior methods for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 17(4), 851–868. [doi]. 4

[28] Ch. Kanzow (2004). Inexact semismooth Newton methods for large-scale complementarity problems. *Optimization Methods and Software*, 19, 309–325. [doi]. 2

[29] M. Kojima, N. Megiddo, T. Noma, A. Yoshise (1991). *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems.* Lecture Notes in Computer Science 538. Springer, Berlin. [doi]. 2, 4

[30] M. Kojima, N. Megiddo, T. Noma, A. Yoshise (1991). A unified approach to interior point algorithms for linear complementarity problems: A summary. *Operations Research Letters*, 10(5), 247–254. [doi]. 4

[31] M. Kojima, S. Shindo (1986). Extension of Newton and quasi-Newton methods to systems of PC^1 equations. *Journal of Operations Research Society of Japan*, 29, 352–375. 2, 5

[32] M.M. Kostreva (1976). *Direct algorithms for complementarity problems.* PhD Thesis, Rensselaer Polytechnic Institute, Troy, New York. 2

[33] O.L. Mangasarian (1977). Solution of symmetric linear complementarity problems by iterative methods. *Journal of Optimization Theory and Applications*, 22, 465–485. [doi]. 2

[34] S. Mizuno, A. Yoshise, T. Kikuchi (1989). Practical polynomial time algorithms for linear complmentarity problems. *Journal of Operations Research Society of Japan*, 32(1), 75–92. [doi]. 8

[35] K.G. Murty (1974). Note on a Bard-type scheme for solving the complementarity problem. *Opsearch*, 11, 123–130. 8

[36] K.G. Murty (1978). Computational complexity of complementarity pivot methods. *Mathematical Programming Study*, 7, 61–73. 7, 8

[37] K.G. Murty (1988). *Linear Complementarity, Linear and Nonlinear Programming* (Internet edition, prepared by Feng-Tien Yu, 1997). Heldermann Verlag, Berlin. 8

[38] J.-S. Pang (1990). Newton's method for B-differentiable equations. *Mathematics of Operations Research*, 15, 311–341. [doi]. 3, 4

[39] J.-S. Pang (1991). A B-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems. *Mathematical Programming*, 51, 101–131. [doi]. 5

[40] L. Qi, J. Sun (1993). A nonsmooth version of Newton's method. *Mathematical Programming*, 58, 353–367. [doi]. 2, 5

[41] H. Samelson, R.M. Thrall, O. Wesler (1958). A partition theorem for the Euclidean n-space. *Proceedings of the American Mathematical Society*, 9, 805–807. [editor]. 3