

**Nonconvergence of the plain Newton-min algorithm for
linear complementarity problems with a P-matrix —
The full report.**

Ibtihel Ben Gharbia, Jean Charles Gilbert

► **To cite this version:**

Ibtihel Ben Gharbia, Jean Charles Gilbert. Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix — The full report.. [Research Report] RR-7160, INRIA. 2012, pp.19. inria-00442293v5

HAL Id: inria-00442293

<https://hal.inria.fr/inria-00442293v5>

Submitted on 17 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Nonconvergence of the plain Newton-min algorithm
for linear complementarity problems with a P-matrix
— The full report*

Ibtihel BEN GHARBA — J. Charles GILBERT

N° 7160

17 décembre 2012

Thème NUM

 *Report
de recherche*

Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a \mathbf{P} -matrix — The full report

Ibtihel BEN GHARBA[†], J. Charles GILBERT[†]

Thème NUM — Systèmes numériques
Équipe-Projet Estime

Rapport de recherche n° 7160 — 17 décembre 2012 — 19 pages

Abstract: The plain Newton-min algorithm to solve the linear complementarity problem (LCP for short) $0 \leq x \perp (Mx + q) \geq 0$ can be viewed as a semismooth Newton algorithm without globalization technique to solve the system of piecewise linear equations $\min(x, Mx + q) = 0$, which is equivalent to the LCP. When M is an \mathbf{M} -matrix of order n , the algorithm is known to converge in at most n iterations. We show in this paper that this result no longer holds when M is a \mathbf{P} -matrix of order ≥ 3 , since then the algorithm may cycle. \mathbf{P} -matrices are interesting since they are those ensuring the existence and uniqueness of the solution to the LCP for an arbitrary q . Incidentally, convergence occurs for a \mathbf{P} -matrix of order 1 or 2.

Key-words: linear complementarity problem, Newton's method, nonconvergence, nonsmooth function, \mathbf{M} -matrix, \mathbf{P} -matrix.

[†] INRIA Paris-Rocquencourt, team-project Estime, BP 105, F-78153 Le Chesnay Cedex (France); e-mails: Ibtihel.Ben_Gharbia@inria.fr, Jean-Charles.Gilbert@inria.fr.

Non convergence de l'algorithme de Newton-min simple pour les problèmes de complémentarité linéaires avec **P**-matrice — Le rapport complet

Résumé : L'algorithme Newton-min, utilisé pour résoudre le problème de complémentarité linéaire (PCL) $0 \leq x \perp (Mx + q) \geq 0$ peut être interprété comme un algorithme de Newton non lisse sans globalisation cherchant à résoudre le système d'équations linéaires par morceaux $\min(x, Mx + q) = 0$, qui est équivalent au PCL. Lorsque M est une **M**-matrice d'ordre n , on sait que l'algorithme converge en au plus n itérations. Nous montrons dans cet article que ce résultat ne tient plus lorsque M est une **P**-matrice d'ordre $n \geq 3$; l'algorithme peut en effet cycloper dans ce cas. On a toutefois la convergence de l'algorithme pour une **P**-matrice d'ordre 1 ou 2.

Mots-clés : fonction non-lisse, méthode de Newton, non-convergence, **M**-matrice, **P**-matrice, problème de complémentarité linéaire.

1 Introduction

The linear complementarity problem (LCP) consists in finding a vector $x \geq 0$ with n components such that $Mx + q \geq 0$ and $x^\top(Mx + q) = 0$. Here M is a real matrix of order n , q is a vector in \mathbb{R}^n , the inequalities have to be understood componentwise, and the sign $^\top$ denotes matrix transposition. The LCP is often written in compact form as follows

$$\text{LC}(M, q) : \quad 0 \leq x \perp (Mx + q) \geq 0.$$

This problem is known to have a unique solution for any $q \in \mathbb{R}^n$ if and only if M is a **P**-matrix [34, 11], i.e., a matrix with positive principal minors: $\det M_{II} > 0$ for all nonempty $I \subset \{1, \dots, n\}$. We denote by **P** the set of **P**-matrices. Other classes of matrices M intervening in the discussion below are the class **Z** of **Z**-matrices (which have nonpositive off-diagonal elements: $M_{ij} \leq 0$ for all $i \neq j$) and the class **M** := **P** \cap **Z** of **M**-matrices (they are called **K**-matrices in [11]).

Since the components of x and $Mx + q$ must be nonnegative in $\text{LC}(M, q)$, their perpendicularity with respect to the Euclidean scalar product required in the problem is equivalent to the nullity of the Hadamard product of the two vectors, that is

$$x \cdot (Mx + q) = 0. \quad (1.1)$$

Recall that the Hadamard product $u \cdot v$ of two vectors u and v is the vector having its i th component equal to $u_i v_i$. A point x such that (1.1) holds is here called a *node* or is said *to satisfy complementarity*. Since, for a node x , either x_i or $(Mx + q)_i$ vanishes, for all indices i , there are at most 2^n nodes for a *nondegenerate* matrix M , which is a matrix having nonsingular principal submatrices. On the other hand, a point x such that x and $Mx + q$ are nonnegative (resp. positive) is said to be *feasible* (resp. *strictly feasible*). A solution to $\text{LC}(M, q)$ is therefore a feasible node.

Many algorithms have been proposed to solve problem $\text{LC}(M, q)$ [32, 11]. They may be based on pivoting techniques [27, 10], which often suffer from the combinatorial aspect of the problem (i.e., the 2^n possibilities to realize (1.1)), on interior point methods, which originate from an algorithm introduced by Karmarkar in linear optimization [21; 1984] (see also [23; 1991] for one of the first accounts on the use of interior point methods to solve linear complementarity problems), and on nonsmooth Newton approaches [13], such as the one considered here. See [32, 11] for other iterative methods.

The algorithm we consider in this paper maintains the complementarity condition (1.1) at each iteration, while feasibility is obtained at convergence (when this one occurs). As a result, all the iterates are nodes, except possibly the first one, and the algorithm terminates as soon as it has found a feasible iterate. More specifically, suppose that the current iterate x is a node. The algorithm first defines index sets A^+ and I^+ associated with the next iterate x^+ ; in its simplest form, it takes

$$A^+ := \{i : x_i \leq (Mx + q)_i\} \quad \text{and} \quad I^+ := \{i : x_i > (Mx + q)_i\}. \quad (1.2)$$

Then it computes x^+ by solving the linear system formed of the equations

$$x_{A^+}^+ = 0 \quad \text{and} \quad (Mx^+ + q)_{I^+} = 0. \quad (1.3)$$

To have a well defined algorithm, an assumption on M is necessary so that this system has a solution for any choice of complementary sets A^+ and I^+ , and vector q : M must be nondegenerate.

The motivation sustaining the algorithm is that it can be viewed as a semismooth Newton method to solve the system of piecewise linear equations

$$\min(x, Mx + q) = 0, \quad (1.4)$$

in which the minimum operator ‘min’ acts componentwise: $[\min(x, y)]_i = \min(x_i, y_i)$. On the one hand, since, for a and $b \in \mathbb{R}$, $\min(a, b) = 0$ if and only if $0 \leq a \perp b \geq 0$, the system (1.4) is

indeed equivalent to problem $LC(M, q)$ (see [25; 1976] and [28; 1977, lemma 2.1] for instance). On the other hand, it is indeed clear that the function in (1.4) is differentiable at a point x without *doubly active index* (i.e., without index i such that $x_i = (Mx + q)_i$) and that its Jacobian matrix is the one used in the linear system (1.3); when there are doubly active indices, the Jacobian used in (1.3), determined by the choice (1.2), is an element of the Clarke generalized Jacobian [8] of the function in (1.4), so that the algorithm may indeed be viewed as a semismooth Newton method [33; 1993]. This description makes it natural to call *Newton-min* the algorithm that updates x by the formulas (1.2)–(1.3).

Here are some remarks about algorithm (1.2)–(1.3). First, note that the algorithm has a principle quite different from the one used by an interior point approach, which generates strictly feasible points, while complementarity is obtained at the limit. Note also that if, in the local analysis of the method, it is important to allow the first iterate x^1 not to be a node, in this paper, x^1 will always be assumed to be a node. Finally, observe that a consequence of the fact that the algorithm only generates nodes is that it is equivalent to say that it converges or that it converges in a finite number of iterations or that it does not cycle (algorithm (1.2)–(1.3) is a Markov process).

The algorithm sketched above and that we further explore in this paper can be traced back at least to the algorithm (6.2)–(6.4) of Aganagić in [1; 1984], who considers a linear complementarity problem that reads instead $0 \leq (Xx) \perp (Yx + q) \geq 0$, in which X and Y must be jointly \mathbf{M} -regular. Here, it is not important to be precise about the meaning of this property of joint \mathbf{M} -regularity, but to know that if $X = I$, as in $LC(M, q)$, then the property is equivalent to the \mathbf{M} -matrix property of $Y \equiv M$. When $X = I$ and $Y \equiv M$, the algorithm (6.2)–(6.4) of Aganagić [1] is exactly the algorithm (1.2)–(1.3) above, which is proven in [1; theorem 6.2] to be *monotonically* (in the sense that $x^k \leq x^{k+1}$ for $k \geq 2$) and *globally* (i.e., x^1 may be arbitrary) convergent to the unique solution of $LC(M, q)$, provided $M \in \mathbf{M}$ (in [1], the first iterate x^1 is supposed to be zero, but this assumption is not necessary). Under the conditions that $x^1 = 0$ and $M \in \mathbf{M}$, this algorithm is actually identical to the one proposed and analyzed earlier by Chandrasekaran [7; 1970] [1; remark 2]. Although not presented in that manner, the algorithm proposed by Bergounioux, Ito, and Kunisch [4; 1999] to solve a strongly convex quadratic optimal control problem, under the name of *primal-dual active set strategy* (PDAS), can be viewed as an extension of algorithm (1.2)–(1.3) to an *infinite* dimensional problem (see [3; 2000] for a comparison with interior point methods); the authors prove the convergence of the algorithm on a discretized version of the problem under some conditions. In [18; 2003], Hintermüller, Ito, and Kunisch consider a quadratic optimization problem with simple bounds, which can actually be written as a linear complementarity problem like $LC(M, q)$; they establish the equivalence between the PDAS strategy and the semismooth Newton method, i.e., algorithm (1.2)–(1.3); the algorithm is also shown to be *locally superlinearly* convergent when $M \in \mathbf{P}$, and to be *monotonically* (in the sense that $\sum_i x_i^k \leq \sum_i x_i^{k+1}$) and *globally* convergent when M is in the set that we denote here by

$$\mathbf{M}_\varepsilon := \{M : M \text{ is a matrix near an } \mathbf{M}\text{-matrix of the same order}\}. \quad (1.5)$$

In \mathbf{M}_ε , the level of proximity to an \mathbf{M} -matrix is left unprecise (and depends on the considered matrix M), but in the proof of [18; theorem 3.4], this proximity is sufficiently tight to imply $\mathbf{M}_\varepsilon \subset \mathbf{P}$. Another interesting property of algorithm (1.2)–(1.3), proved by Kanzow [20; 2004], is its convergence in at most n iterations when $M \in \mathbf{M}$ and the first iterate is a node. When applied to algorithm (1.2)–(1.3), the result of Fischer and Kanzow [15; 1996] shows that a solution to $LC(M, q)$, if any, is reached in one step, provided x^1 is sufficiently close to that solution and M is a nondegenerate matrix. To conclude this review of results, we would like to cite the quadratic local convergence of Newton's method for piecewise C^1 functions proved by Kojima and Shindo [24; 1986], which is related to the formulation (1.4) of problem $LC(M, q)$.

This paper presents examples of nonconvergence of the Newton-min algorithm when M is a \mathbf{P} -matrix. These counter-examples hold for the *plain* (or undamped) Newton-min algorithm, i.e., without the use of globalization techniques such as linesearch or trust regions. One may believe

that, as a Newton-like method to solve a nonlinear (and nonsmooth) system of equations, this is not a good strategy. We share this opinion, in general. However the algorithm deals with a piecewise linear function, and it has been shown to be convergent without globalization techniques when M is a \mathbf{P} -matrix sufficiently near an \mathbf{M} -matrix (see the discussion in the previous paragraph). Therefore, searching for the weakest assumptions for which these convergence properties hold seems to us a valid topic. The examples in this paper show that it is not enough to require the \mathbf{P} -matricity for M .

The paper is structured as follows. In the next section, we are more specific on the definition of the algorithm, by making it a slightly more flexible than in the description (1.2)–(1.3) above. Some elementary properties of the algorithm, useful in the sequel, are also given. Section 3 describes and analyzes the examples of nonconvergence of the plain Newton-min algorithm with a \mathbf{P} -matrix, when $n \geq 3$. These counter-examples work for both definitions of the algorithm, those of sections 1 and 2. In them, the algorithm can be forced to cycle and visit p nodes, with a p that can be chosen arbitrarily in $\{3, \dots, n\}$. We consider successively the cases when n is odd and even, which require a different analysis. These counter-examples readily imply that the convergence radius of the plain Newton-min algorithm for solving $\text{LC}(M, q)$ with $M \in \mathbf{P}$, which is known to be > 0 [18; 2003], can be arbitrarily small (corollary 3.8). In section 4, the plain Newton-min algorithm is claimed to converge for a \mathbf{P} -matrix when $n = 1$ or $n = 2$, a crumb of consolation. The paper concludes with the perspective section 5.

2 The algorithm

The plain Newton-min algorithm described in this section generates points that satisfy the complementarity conditions in (1.1), while the nonnegativity conditions in $\text{LC}(M, q)$ are satisfied when the solution is reached. The starting point x^1 may or may not satisfy this complementarity condition.

Algorithm 2.1 (plain Newton-min) Let $x^1 \in \mathbb{R}^n$.

For $k = 2, 3, \dots$, do the following.

1. If x^{k-1} is a solution to $\text{LC}(M, q)$, stop.
2. Choose complementary index sets $A^k := A_0^k \cup A_1^k$ and $I^k := I_0^k \cup I_1^k$, where

$$\begin{aligned} A_0^k &:= \{i : x_i^{k-1} < (Mx^{k-1} + q)_i\}, \\ A_1^k &\subset E^k := \{i : x_i^{k-1} = (Mx^{k-1} + q)_i\}, \\ I_0^k &:= \{i : x_i^{k-1} > (Mx^{k-1} + q)_i\}, \\ I_1^k &:= E^k \setminus A_1^k. \end{aligned}$$

3. Determine x^k as a solution to

$$\begin{cases} x_{A^k}^k = 0 \\ (Mx^k + q)_{I^k} = 0. \end{cases} \quad (2.1)$$

The algorithm is well defined if M is nondegenerate. This assumption will be generally reinforced by supposing that $M \in \mathbf{P}$. We recall from [14, 11] that

$$M \in \mathbf{P} \quad \iff \quad \text{any } x \text{ verifying } x \cdot (Mx) \leq 0 \text{ vanishes.} \quad (2.2)$$

Note that algorithm 2.1 is more flexible than the one presented in section 1, in that the doubly active indices (those in E^k) can be chosen to belong either to A^k or I^k . If this choice is not entirely determined by the current point x^k , the generated sequence $\{x^k\}$ may no longer be a Markov process. We will not be more precise here, however, since this choice has actually no impact on the counter-examples given in this paper, for which the algorithm never generates iterates with doubly active indices.

In this paper, we always assume that x^1 is a node; then there are two complementary subsets A^1 and I^1 of $\{1, \dots, n\}$, such that $x_{A^1}^1 = 0$ and $(Mx^1 + q)_{I^1} = 0$. In other contexts, in particular for studying the local convergence of the algorithm [18], it is better not to make this assumption.

By the selection of the index sets in step 2, one certainly has for $k \geq 2$:

$$x_{A^k}^{k-1} \leq (Mx^{k-1} + q)_{A^k} \quad \text{and} \quad x_{I^k}^{k-1} \geq (Mx^{k-1} + q)_{I^k}. \quad (2.3)$$

Therefore,

$$x_{A^k}^{k-1} \leq 0 \quad \text{and} \quad (Mx^{k-1} + q)_{I^k} \leq 0 \quad (2.4)$$

must hold [18, 20]. Indeed, for $i \in A^k$, $x_i^{k-1} \leq (Mx^{k-1} + q)_i$ by (2.3) and, since either $x_i^{k-1} = 0$ or $(Mx^{k-1} + q)_i = 0$ by complementarity, one necessarily has $x_i^{k-1} \leq 0$, which proves the first inequality in (2.4). A similar reasoning yields the second inequality in (2.4).

We denote by e^i the i th vector of the canonical basis of \mathbb{R}^n : its j th component is equal to 1 if $j = i$ and to 0 otherwise.

3 Nonconvergence for $n \geq 3$

In this section, we show that the plain Newton-min algorithm described in section 2 may not converge if M is a \mathbf{P} -matrix and $n \geq 3$. We start in section 3.1 with the case when n is odd and provide an example of a \mathbf{P} -matrix M , a vector q , and a starting point, for which the algorithm makes cycles having n nodes. In section 3.2, we consider the case when n is even and construct another class of \mathbf{P} -matrices, which can be viewed as perturbations of the matrices in the first example, for which a cycle having n nodes is also possible. We conclude in section 3.3 by constructing examples for which the plain Newton-min algorithm makes cycles visiting p nodes, with p arbitrary in $\{3, \dots, n\}$.

3.1 Cycles with an odd number of nodes

In this section, we show the nonconvergence of the plain Newton-min algorithm for problems having the following features.

Example 3.1 Let $n \geq 2$. The matrix $M \in \mathbb{R}^{n \times n}$ and the vector $q \in \mathbb{R}^n$ are given by

$$M = \begin{pmatrix} 1 & & & \alpha \\ \alpha & \ddots & & \\ & \ddots & 1 & \\ & & \alpha & 1 \end{pmatrix} \quad \text{and} \quad q = \mathbf{1},$$

where $\mathbf{1}$ denotes the vector of all ones and the elements of M that are not represented are zeros. More precisely, $M_{ij} = 1$ if $i = j$, $M_{ij} = \alpha$ if $i = (j \bmod n) + 1$, and $M_{ij} = 0$ otherwise. Since $q \geq 0$, a solution to $\text{LC}(M, q)$ for these data is $x = 0$. \square

The matrix M and the vector q in example 3.1 have already been used by Morris [31] (in that paper, $\alpha = 2$, M is the transpose of the one here, and $q = -\mathbf{1}$), although we arrived at them in a

different manner, as explained in section 4. For completeness and precision, we start by studying the \mathbf{P} -matricity of the matrix M in example 3.1 (the result for n odd and $\alpha = 2$ was already claimed in [31] without a detailed proof).

Lemma 3.2 *Consider the matrix M in example 3.1. If n is even, $M \in \mathbf{P}$ if and only if $|\alpha| < 1$. If n is odd, $M \in \mathbf{P}$ if and only if $\alpha > -1$.*

PROOF. Observe first that if $\alpha \leq -1$, the nonzero vector $x = \mathbf{1}$ is such that $x \cdot (Mx) = (1 + \alpha)\mathbf{1} \leq 0$; hence $M \notin \mathbf{P}$.

Observe next that $M \in \mathbf{P}$ when $-1 < \alpha < 0$. Indeed, let $x \in \mathbb{R}^n$ be such that $x \cdot (Mx) \leq 0$ or equivalently

$$x_1(x_1 + \alpha x_n) \leq 0, \quad x_2(x_2 + \alpha x_1) \leq 0, \quad \dots, \quad x_n(x_n + \alpha x_{n-1}) \leq 0. \quad (3.1)$$

If $x_n > 0$, using (3.1) from right to left shows that all the components of x are positive and verify

$$0 < x_n \leq |\alpha|x_{n-1} \leq |\alpha|^2 x_{n-2} \leq \dots \leq |\alpha|^{n-1} x_1 \leq |\alpha|^n x_n.$$

Since $|\alpha| < 1$, this is incompatible with $x_n > 0$. Having $x_n < 0$ is not possible either (just multiply x by -1 and use the same argument). Hence $x_n = 0$ and using (3.1) from left to right shows that $x = 0$. The \mathbf{P} -matricity now follows from (2.2).

When $\alpha = 0$, M is the identity matrix and is therefore a \mathbf{P} -matrix.

Suppose now that n is even. If $\alpha \geq 1$, the nonzero vector x , defined by $x_i = (-1)^{i+1}$, is such that $x \cdot (Mx) = (1 - \alpha)\mathbf{1} \leq 0$; hence $M \notin \mathbf{P}$. Let us now show, by contradiction, that $M \in \mathbf{P}$ if $0 < \alpha < 1$, assuming that there is a nonzero x satisfying $x \cdot (Mx) \leq 0$, hence that (3.1) holds. Then, as above, all the components of x are nonzero and one can assume that $x_n > 0$. Starting with the rightmost inequality of (3.1), one obtains by induction for $i = 1, \dots, n - 1$:

$$x_{n-i} \leq -\frac{1}{\alpha^i} x_n < 0 \quad (\text{for } i \text{ odd}) \quad \text{and} \quad x_{n-i} \geq \frac{1}{\alpha^i} x_n > 0 \quad (\text{for } i \text{ even}).$$

Since n is even, $x_1 \leq -(1/\alpha^{n-1})x_n < 0$ and, using the first inequality of (3.1), $x_n \geq -(1/\alpha)x_1 \geq (1/\alpha^n)x_n$, which is in contradiction with $0 < \alpha < 1$ and $x_n > 0$.

Suppose finally that n is odd and $\alpha > 0$. Again, for proving that $M \in \mathbf{P}$, we argue by contradiction, assuming that there is a nonzero x such that $x \cdot (Mx) \leq 0$, which is equivalent to (3.1). As above, one can assume that $x_n > 0$. Starting with the rightmost inequality in (3.1), one can specify by induction the sign $\sigma(x_i)$ of the x_i 's:

$$\sigma(x_{n-1}) = -1, \quad \sigma(x_{n-2}) = 1, \quad \dots, \quad \sigma(x_1) = (-1)^{n-1} = 1,$$

since n is odd. Finally, the first inequality in (3.1) gives $\sigma(x_n) = -\sigma(x_1) = -1$, in contradiction with $x_n > 0$. Hence $x = 0$ and $M \in \mathbf{P}$ by (2.2). \square

Recall that e^k denotes the k th basis vector of \mathbb{R}^n .

Lemma 3.3 *Suppose that $n \geq 2$ and consider problem $\text{LC}(M, q)$ in which M and q are given in example 3.1 with $\alpha > 1$. When applied to that problem $\text{LC}(M, q)$ and started from $x^1 = -e^1$, algorithm 2.1 cycles by visiting in order the n nodes $x^k = -e^k$, $k = 1, \dots, n$.*

PROOF. The proof proceeds by induction. By assumption, $x^1 = -e^1$.

Suppose now that $x^{k-1} = -e^{k-1}$ for some $k \in \{2, \dots, n\}$ and let us show that $x^k = -e^k$. There hold

$$x^{k-1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{and} \quad Mx^{k-1} + q = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ 1 - \alpha \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

where the component -1 (resp. 0) is at position $k-1$ in x^{k-1} (resp. in $Mx^{k-1} + q$). The update rules of algorithm 2.1 and $\alpha > 1$ then imply that $I^k = \{k\}$ and $x^k = -e^k$.

We still have to show that $x^{n+1} = -e^1$. From $x^n = -e^n$, one deduces that

$$x^n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} \quad \text{and} \quad Mx^n + q = \begin{pmatrix} 1 - \alpha \\ 1 \\ \vdots \\ 1 \\ 0 \end{pmatrix}.$$

Again, the update rules of algorithm 2.1 and $\alpha > 1$ show that $I^{n+1} = \{1\}$ and $x^{n+1} = -e^1$. Therefore algorithm 2.1 cycles. \square

By combining lemmas 3.2 and 3.3, one can see that algorithm 2.1 may cycle when M is a \mathbf{P} -matrix of odd order $n \geq 3$: this is the case when the algorithm is started at $x^1 = -e^1$ and when M and q are given by example 3.1, with n odd and $\alpha > 1$. When n is even, the condition $\alpha > 1$ used in lemma 3.3 prevents the matrix M from being a \mathbf{P} -matrix. It is not difficult to show, however, that the algorithm can also cycle when n is even, $n \geq 4$, and $M \in \mathbf{P}$, by using the construction given in the proof of proposition 3.7: the cycle visits an odd number of nodes (hence $< n$).

3.2 Cycles with an even number of nodes

The next family of examples is obtained by perturbing the matrices of example 3.1 with a parameter β , in order to construct cycles having n nodes for an even order \mathbf{P} -matrix.

Example 3.4 Let $n \geq 3$. The matrix $M \in \mathbb{R}^{n \times n}$ and the vector $q \in \mathbb{R}^n$ are given by

$$M = \begin{pmatrix} 1 & & & \beta & \alpha \\ \alpha & \ddots & & & \beta \\ \beta & \ddots & 1 & & \\ & \ddots & \alpha & 1 & \\ & & \beta & \alpha & 1 \end{pmatrix} \quad \text{and} \quad q = \mathbf{1},$$

where the elements of M that are not represented are zeros or, more precisely, for $i, j \in \{1, \dots, n\}$:

$$M_{ij} = \begin{cases} 1 & \text{if } i = j \\ \alpha & \text{if } i = (j \bmod n) + 1 \\ \beta & \text{if } i = ((j + 1) \bmod n) + 1 \\ 0 & \text{otherwise.} \end{cases}$$

Since $q \geq 0$, a solution to $\text{LC}(M, q)$ for these data is $x = 0$. \square

Conditions for having an n -node cycle with algorithm 2.1 on problem $\text{LC}(M, q)$ with M and q given by example 3.4 are very simple to express.

Lemma 3.5 *Suppose that $n \geq 3$ and consider problem $\text{LC}(M, q)$ in which M and q are given in example 3.4 with $\alpha > 1$ and $\beta < 1$. When applied to that problem $\text{LC}(M, q)$ and started from $x^1 = -e^1$, algorithm 2.1 cycles by visiting in order the n nodes defined by $x^k = -e^k$, $k = 1, \dots, n$.*

PROOF. The proof is quite similar to the one of lemma 3.3 and proceeds by induction. By assumption, $x^1 = -e^1$.

Suppose now that $x^{k-1} = -e^{k-1}$ for some $k \in \{2, \dots, n-1\}$ and let us show that $x^k = -e^k$. There hold

$$x^{k-1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{and} \quad Mx^{k-1} + q = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ 1 - \alpha \\ 1 - \beta \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

where the component -1 (resp. 0) is at position $k-1$ in x^{k-1} (resp. in $Mx^{k-1} + q$). The update rules of algorithm 2.1, $\alpha > 1$, and $\beta < 1$ then imply that $I^k = \{k\}$ and $x^k = -e^k$.

Therefore, by induction

$$x^{n-1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \\ 0 \end{pmatrix} \quad \text{and} \quad Mx^{n-1} + q = \begin{pmatrix} 1 - \beta \\ 1 \\ \vdots \\ 1 \\ 0 \\ 1 - \alpha \end{pmatrix}.$$

The update rules of algorithm 2.1, $\alpha > 1$, and $\beta < 1$ then imply that $I^n = \{n\}$, so that

$$x^n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} \quad \text{and} \quad Mx^n + q = \begin{pmatrix} 1 - \alpha \\ 1 - \beta \\ 1 \\ \vdots \\ 1 \\ 0 \end{pmatrix}.$$

Again, the update rules of algorithm 2.1, $\alpha > 1$, and $\beta < 1$ show that $I^{n+1} = \{1\}$. Therefore, $x^{n+1} = -e^1$ and algorithm 2.1 cycles. \square

We have now to examine whether the conditions on α and β given by lemma 3.5 are compatible with the \mathbf{P} -matricity of the matrix M in example 3.4. Actually, the conditions on α and β ensuring the \mathbf{P} -matricity of that matrix M are nonlinear and much more complex to write than the simple conditions on α given in lemma 3.2; in particular, their number depends on the dimension n . The

shaded regions in figure 3.1 show the intersections with the box $[-2, 2] \times [-2, 2]$ of the sets \mathcal{P} formed of the (α, β) pairs for which the matrix M is a \mathbf{P} -matrix, when its order is $n = 3, 4, 5,$ and 6 . Of course, by lemma 3.2, \mathcal{P} contains the set $\{(\alpha, \beta) : -1 < \alpha < 1, \beta = 0\}$ when n is even and the set $\{(\alpha, \beta) : -1 < \alpha, \beta = 0\}$ when n is odd. It is not clear at this point, however, whether, for any $n \geq 3$, these regions will contain points with $\alpha > 1$ and $\beta < 1$, which are the conditions highlighted by lemma 3.5. Lemma 3.6 below shows that this is actually the case since \mathcal{P} always contains the interiors of the nonconvex polyhedron and the ellipse represented in figure 3.1, which are independent of n .

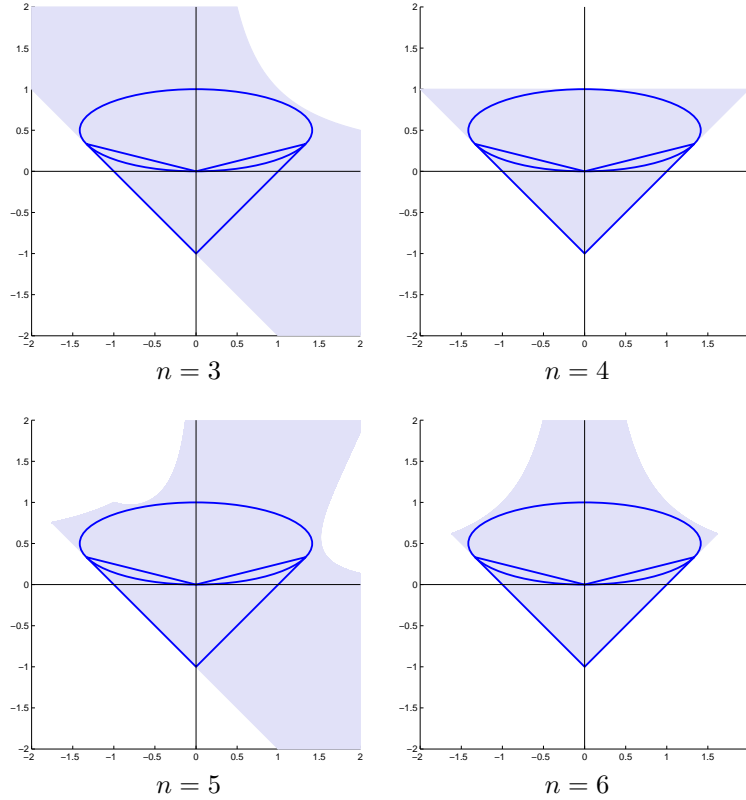


Figure 3.1: The shaded regions are formed of the (α, β) pairs in $[-2, 2] \times [-2, 2]$ for which the matrix M of example 3.4 is a \mathbf{P} -matrix, when its order is $n = 3, 4, 5,$ and 6 ; these regions are nonconvex but star-shaped with respect to $(0, 0)$, which is the point corresponding to the identity matrix. According to lemma 3.6, the interiors of the represented nonconvex polyhedron and ellipse, which are independent of n , are always contained in these regions, for any $n \geq 3$.

To prepare the proof of lemma 3.6, we write the circulant matrix M in example 3.4 as follows

$$M = I + \beta J^{n-2} + \alpha J^{n-1},$$

where I denotes the $n \times n$ identity matrix and J is the elementary circulant $n \times n$ matrix

$$J = \begin{pmatrix} 0 & 1 & & 0 \\ & 0 & \ddots & \\ & & \ddots & 1 \\ 1 & & & 0 \end{pmatrix}.$$

More precisely, $J_{ij} = 1$ if $j = (i \bmod n) + 1$ and $J_{ij} = 0$ otherwise. It is well known [16; formula (4.7.10)] that J is diagonalizable on \mathbb{C} , meaning that there is a diagonal matrix $D \in \mathbb{C}^{n \times n}$ and a nonsingular matrix $P \in \mathbb{C}^{n \times n}$ such that $J = PDP^{-1}$; in addition its eigenvalues are the n th roots of unity: $D := \text{Diag}(1, w^1, \dots, w^{n-1})$, where $w = e^{2\pi i/n}$ and i is the imaginary unit.

Lemma 3.6 *The set of (α, β) pairs ensuring that $M \in \mathbf{P}$ in example 3.4 contains the set $\{(\alpha, \beta) : |\alpha| - 1 < \beta < |\alpha|/4 \text{ or } \alpha^2 + 8(\beta - \frac{1}{2})^2 < 2\}$.*

PROOF. We only have to show that when α and β satisfy

$$|\alpha| - 1 < \beta < \frac{|\alpha|}{4} \quad \text{or} \quad \alpha^2 + 8\left(\beta - \frac{1}{2}\right)^2 < 2, \quad (3.2)$$

the matrix $M + M^\top$ is positive definite, since then M is clearly a \mathbf{P} -matrix [22; p. 175].

For any integer p , $(J^p)^\top = J^{n-p}$. Therefore $M + M^\top = 2I + \alpha J + \beta J^2 + \beta J^{n-2} + \alpha J^{n-1}$ and, using $J = PDP^{-1}$, we obtain

$$M + M^\top = P(2I + \alpha D + \beta D^2 + \beta D^{n-2} + \alpha D^{n-1})P^{-1}.$$

This identity shows that the eigenvalues of the symmetric matrix $M + M^\top$ are the (necessarily real) numbers

$$\lambda_k := 2 + \alpha e^{2k\pi i/n} + \beta e^{4k\pi i/n} + \beta e^{2k(n-2)\pi i/n} + \alpha e^{2k(n-1)\pi i/n},$$

for $k = 0, \dots, n-1$. Using $e^{2kp\pi i/n} + e^{2k(n-p)\pi i/n} = 2 \cos(2kp\pi/n)$ (for p integer) and $\cos 2\theta = 2 \cos^2 \theta - 1$, we obtain

$$\lambda_k = 2 + 2\alpha \cos(2k\pi/n) + 4\beta \cos^2(4k\pi/n) - 2\beta.$$

We see that the desired positivity of the eigenvalues λ_k depends on the positivity of the following polynomial on $[-1, 1]$:

$$t \mapsto \varphi(t) = 2\beta t^2 + \alpha t + (1 - \beta).$$

We denote by $t_\pm := [-\alpha \pm (\alpha^2 - 8\beta(1 - \beta))^{1/2}]/(4\beta)$ the roots of φ when $\beta \neq 0$ and consider in sequence the three possible cases, identifying in each case conditions that ensure the positivity of φ on $[-1, 1]$.

- Case $\beta = 0$. Then φ is positive on $[-1, 1]$ if $|\alpha| < 1$.
- Case $\beta > 0$. There are two subcases.
 - If φ has no real root, i.e., if $\alpha^2 < 8\beta(1 - \beta)$ or equivalently $\alpha^2 + 8(\beta - \frac{1}{2})^2 < 2$, then φ is clearly positive on \mathbb{R} .
 - If φ has two (possibly equal) real roots t_\pm , i.e., if $\alpha^2 \geq 8\beta(1 - \beta)$, then these verify $t_- \leq t_+$ and φ is positive on $[-1, 1]$,
 - either if $1 < t_-$, which is ensured if $-\alpha - 1 < \beta < -\alpha/4$,
 - or if $t_+ < -1$, which is ensured if $\alpha - 1 < \beta < \alpha/4$.
- Case $\beta < 0$. Then, φ has two real roots t_\pm , which verify $t_+ < t_-$, and φ is positive on $[-1, 1]$ if both $t_+ < -1$ and $1 < t_-$ holds, which is ensured if $|\alpha| - 1 < \beta < 0$.

By gathering the above conditions, we obtain (3.2). \square

3.3 Nonconvergence

The nonconvergence result is summarized in proposition 3.7, in which it is shown that cycles made of p nodes are possible when $n \geq 3$ and $p \in \{3, \dots, n\}$. It is then shown with proposition 3.8 that when $n \geq 3$, although the plain Newton-min algorithm is known to converge locally, i.e., when the starting point is near the solution to $\text{LC}(M, q)$, the radius of convergence can be made as small as desired by modifying q ; this makes the convergence of the plain Newton-min algorithm unlikely.

Proposition 3.7 (nonconvergence for $n \geq 3$) *When $n \geq 3$, algorithm 2.1 may fail to converge when trying to solve $\text{LC}(M, q)$ with a \mathbf{P} -matrix M . A cycle made of p nodes is possible, for an arbitrary $p \in \{3, \dots, n\}$.*

PROOF. Since the plain Newton-min algorithm visits only a finite number of nodes, it fails to converge if and only if it cycles.

When $n \geq 3$ is odd, a cycle made of n nodes is possible on problem $\text{LC}(M, q)$ with M and q given by example 3.1, and $\alpha > 1$: lemma 3.2 shows that $M \in \mathbf{P}$ and lemma 3.3 shows that a cycle is possible.

When $n \geq 4$ is even, a cycle made of n nodes is possible on problem $\text{LC}(M, q)$ with M and q given by example 3.4, and α and β satisfying $\alpha > 1$ and $\alpha - 1 < \beta < \alpha/4$ (hence $\beta < 1/3$): lemma 3.5 shows that a cycle is possible and lemma 3.6 shows that $M \in \mathbf{P}$.

When $n \geq 3$ and $p \in \{3, \dots, n\}$, consider a $p \times p$ matrix $\tilde{M} \in \mathbf{P}$, a vector $\tilde{q} \in \mathbb{R}^p$, and a starting point $\tilde{x}^1 \in \mathbb{R}^p$, such that algorithm 2.1 applied to problem $\text{LC}(\tilde{M}, \tilde{q})$ and starting at \tilde{x}^1 generates iterates \tilde{x}^k forming a cycle made of p nodes (this is possible by what has just been proven). With obvious notation, define

$$M = \begin{pmatrix} \tilde{M} & 0_{p \times (n-p)} \\ 0_{(n-p) \times p} & I_{n-p} \end{pmatrix}, \quad q = \begin{pmatrix} \tilde{q} \\ 0_{n-p} \end{pmatrix}, \quad \text{and} \quad x^1 = \begin{pmatrix} \tilde{x}^1 \\ 0_{n-p} \end{pmatrix}.$$

The \mathbf{P} -matrixity of M is clear, by observing that $x \cdot (Mx) \leq 0$ implies $x = 0$. Denote by x^k the k th iterate generated by algorithm 2.1 on $\text{LC}(M, q)$ starting from x^1 . Observe first that when an index $i > p$, there holds $x_i^k = 0$, whenever $i \in I^k$ or $i \in A^k$; therefore the generated iterates $x^k \in \mathbb{R}^p \times \{0_{n-p}\}$. Hence, if the same rule as the one used by algorithm 2.1 on \mathbb{R}^p is used to decide whether an index $i \in E^k$ will be considered as being in I^k or A^k , the iterates x^k will be $(\tilde{x}^k, 0_{n-p})$. Obviously, as the \tilde{x}^k 's, these iterates also form a cycle made of p nodes. \square

To make the above nonconvergence result more concrete, we provide two examples of \mathbf{P} -matrices M_n , of order $n = 3$ and $n = 4$ respectively, which make algorithm 2.1 fail with an n -cycle, when it starts at $x^1 = (-1, 0, \dots, 0)$ to solve problem $\text{LC}(M_n, \mathbf{1})$:

$$M_3 := \begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix} \quad \text{and} \quad M_4 := \begin{pmatrix} 1 & 0 & 1/2 & 4/3 \\ 4/3 & 1 & 0 & 1/2 \\ 1/2 & 4/3 & 1 & 0 \\ 0 & 1/2 & 4/3 & 1 \end{pmatrix}. \quad (3.3)$$

We have used the lemmas 3.2 and 3.3 for constructing M_3 and the lemmas 3.5 and 3.6 for designing M_4 .

The *convergence radius* of an iterative algorithm for finding a solution \bar{x} to a given problem is the largest $\rho > 0$ such that the algorithm converges to \bar{x} when the initial iterate is in the ball of center \bar{x} and radius ρ . The plain Newton-min algorithm is known to be locally convergent [18; 2003]. By scaling the variables in the previous examples, however, one can make the convergence radius of the plain Newton-min algorithm as small as desired. Even though this observation is straightforward,

we highlight it in the following corollary to stress the fact that without modification the plain Newton-min algorithm may have little chance to converge.

Corollary 3.8 (small convergence radius for $n \geq 3$) *When $n \geq 3$, the convergence radius of algorithm 2.1 to solve $\text{LC}(M, q)$ with a \mathbf{P} -matrix M may be arbitrarily small.*

PROOF. Take an example of matrix $M \in \mathbf{P}$ and vector $q \geq 0$ such that algorithm 2.1 cycles when it tries to solve $\text{LC}(M, q)$ from some nonzero x^1 (this is possible by using one of the problems considered in the proof of proposition 3.7). The convergence radius of algorithm 2.1 for that problem is therefore less than $\|x^1\|$ (the solution is 0).

Now, algorithm 2.1 starting at $\tilde{x}^1 = \varepsilon x^1$, for some $\varepsilon > 0$, for solving $\text{LC}(M, \varepsilon q)$ generates the iterates $\tilde{x}^k = \varepsilon x^k$ and therefore cycles. The convergence radius for the plain Newton-min algorithm on this new problem is less than $\varepsilon \|x^1\|$, which can be made arbitrarily small by letting $\varepsilon \downarrow 0$. \square

4 Convergence for $n = 1$ or 2

In this section, we prove the convergence of the plain Newton-min algorithm, algorithm 2.1, when M is a \mathbf{P} -matrix of order 1 or 2. The proof for $n = 1$ is straightforward. The one presented for $n = 2$ is indirect but highlights the origin of the counter-example 3.1 and allows us to present some properties of the plain Newton-min algorithm.

The convergence of the plain Newton-min algorithm when $n = 1$ is a direct consequence of the following reassuring and elementary property, already proven in [4; theorem 2.1] in the case where the complementarity problem expresses the optimality conditions of an infinite dimensional quadratic optimization problem.

Lemma 4.1 (stagnation at a solution) *Suppose that M is nondegenerate. Then, a node is a solution to $\text{LC}(M, q)$ if and only if the plain Newton-min algorithm, without its stopping test in step 1, starting at that node, takes the same node as the next iterate.*

PROOF. Let x be a node of problem $\text{LC}(M, q)$. We denote by A^+ and I^+ the index sets determined in step 2 of algorithm 2.1 and by x^+ the next iterate.

If x is a solution, then $x \geq 0$ and $Mx + q \geq 0$. There hold $x_{A^+}^+ = 0$ by (2.1) and $x_{A^+} = 0$ by (2.4) and the nonnegativity of x , so that $x_{A^+}^+ = x_{A^+}$. Similarly, there hold $(Mx^+ + q)_{I^+} = 0$ by (2.1) and $(Mx + q)_{I^+} = 0$ by (2.4) and the nonnegativity of $Mx + q$, so that $0 = (M(x^+ - x))_{I^+} = M_{I^+ I^+} (x^+ - x)_{I^+}$ [since $(x^+ - x)_{A^+} = 0$] and therefore $(x^+ - x)_{I^+} = 0$ by the nonsingularity of $M_{I^+ I^+}$. We have shown that $x^+ = x$.

Conversely, assume that $x^+ = x$. By (2.3), there hold $x_{A^+} \leq (Mx + q)_{A^+}$ and $x_{I^+} \geq (Mx + q)_{I^+}$. Since $x_{A^+} = x_{A^+}^+ = 0$ [by (2.1)] and $(Mx + q)_{I^+} = (Mx^+ + q)_{I^+} = 0$ [by (2.1)], we get $x \geq 0$ and $Mx + q \geq 0$; hence x is a solution. \square

We consider now the case when $n = 1$ and M is nondegenerate. For such a trivial problem, a direct proof of convergence is obviously possible. The proof given below uses instead lemma 4.1, which is nevertheless useful elsewhere.

Proposition 4.2 (convergence for $n = 1$) *Suppose that $n = 1$, that M is nondegenerate ($M \neq 0$), and that $\text{LC}(M, q)$ has a solution. Then the plain Newton-min algorithm converges.*

PROOF. Without restriction, it can be assumed that the first iterate x^1 is a node. If x^1 is a solution, the algorithm stops at that point (by step 1 of algorithm 2.1). If x^1 is not the solution, there is another node that is solution (by assumption). Since there are no more than 2 nodes (since $n = 1$), the algorithm takes the solution as the next iterate (by lemma 4.1) and stops there. \square

Since a \mathbf{P} -matrix is nondegenerate, the elementary preceding result applies when $M \in \mathbf{P}$. When $M \neq 0 \in \mathbb{R}$ but $\text{LC}(M, q)$ has no solution, the algorithm cycles by visiting alternatively the 2 nodes of the problem, i.e., $x = -q/M < 0$ and $x = 0$.

We start the study of the convergence of the plain Newton-min algorithm when $n = 2$ by showing that the algorithm cannot do cycles made of 2 nodes (lemma 4.3). We have seen with examples 3.1 and 3.4 that the algorithm can do a 3-cycle when $n \geq 3$, but this implies some conditions that are highlighted by lemma 4.4. We finally show that these conditions cannot be satisfied when $n = 2$ and, as a result, that the algorithm must converge (proposition 4.5).

Lemma 4.3 (no 2-cycle) *If M is a \mathbf{P} -matrix, then the plain Newton-min algorithm does not make cycles formed of 2 distinct nodes.*

PROOF. We argue by contradiction, assuming that the algorithm visits in order the following nodes $x^1 \rightarrow x^2 \rightarrow x^1$, with $x^1 \neq x^2$. Since the algorithm goes from x^1 to x^2 and from x^2 to x^1 , the definition of the sets A^k and I^k in step 2 of the algorithm implies that

$$x_{A^2}^1 \leq (Mx^1 + q)_{A^2} \quad \text{and} \quad x_{I^2}^1 \geq (Mx^1 + q)_{I^2}, \quad (4.1)$$

$$x_{A^1}^2 \leq (Mx^2 + q)_{A^1} \quad \text{and} \quad x_{I^1}^2 \geq (Mx^2 + q)_{I^1}. \quad (4.2)$$

After possible rearrangement of the component order, we get

$$x^2 - x^1 = \begin{pmatrix} 0_{A^1 \cap A^2} \\ x_{A^1 \cap I^2}^2 \\ 0_{I^1 \cap A^2} \\ x_{I^1 \cap I^2}^2 \end{pmatrix} - \begin{pmatrix} 0_{A^1 \cap A^2} \\ 0_{A^1 \cap I^2} \\ x_{I^1 \cap A^2}^1 \\ x_{I^1 \cap I^2}^1 \end{pmatrix} = \begin{pmatrix} 0_{A^1 \cap A^2} \\ x_{A^1 \cap I^2}^2 \\ -x_{I^1 \cap A^2}^1 \\ (x^2 - x^1)_{I^1 \cap I^2} \end{pmatrix}.$$

Observe that the components of $x^2 - x^1$ with indices in $A^1 \cap I^2$ are nonpositive since $x_{A^1 \cap I^2}^2 \leq (Mx^2 + q)_{A^1 \cap I^2}$ [by (4.2)₁] = 0 [by (2.1)₂] and that the components with indices in $I^1 \cap A^2$ are nonnegative since $-x_{I^1 \cap A^2}^1 \geq -(Mx^1 + q)_{I^1 \cap A^2}$ [by (4.1)₁] = 0 [by (2.1)₂]. On the other hand, by (2.1)₂, there holds

$$M(x^2 - x^1) = \begin{pmatrix} (Mx^2)_{A^1 \cap A^2} \\ -q_{A^1 \cap I^2} \\ (Mx^2)_{I^1 \cap A^2} \\ -q_{I^1 \cap I^2} \end{pmatrix} - \begin{pmatrix} (Mx^1)_{A^1 \cap A^2} \\ (Mx^1)_{A^1 \cap I^2} \\ -q_{I^1 \cap A^2} \\ -q_{I^1 \cap I^2} \end{pmatrix} = \begin{pmatrix} (M(x^2 - x^1))_{A^1 \cap A^2} \\ -(Mx^1 + q)_{A^1 \cap I^2} \\ (Mx^2 + q)_{I^1 \cap A^2} \\ 0 \end{pmatrix}.$$

In this vector, the components with indices in $A^1 \cap I^2$ are nonnegative since $-(Mx^1 + q)_{A^1 \cap I^2} \geq -x_{A^1 \cap I^2}^1$ [by (4.1)₂] = 0 [by (2.1)₁] and the components with indices in $I^1 \cap A^2$ are nonpositive since $(Mx^2 + q)_{I^1 \cap A^2} \leq x_{I^1 \cap A^2}^2$ [by (4.2)₂] = 0 [by (2.1)₁]. Therefore

$$(x^2 - x^1) \cdot M(x^2 - x^1) \leq 0.$$

Since $M \in \mathbf{P}$, there holds $x^1 = x^2$ by (2.2), contradicting the initial assumption. \square

Lemma 4.4 (necessary conditions for a 3-cycle) *Suppose that M is a \mathbf{P} -matrix and that the plain Newton-min algorithm cycles by visiting in order the three distinct nodes $x^1 \rightarrow x^2 \rightarrow x^3$. Then the following three sets of indices must be nonempty*

$$\begin{cases} (A^1 \cap I^2 \cap I^3) \cup (I^1 \cap A^2 \cap A^3) \\ (I^1 \cap A^2 \cap I^3) \cup (A^1 \cap I^2 \cap A^3) \\ (I^1 \cap I^2 \cap A^3) \cup (A^1 \cap A^2 \cap I^3). \end{cases} \quad (4.3)$$

PROOF. We use the same technique as in the proof of lemma 4.3. Since the algorithm goes from x^1 to x^2 and from x^2 to x^3 , one has from step 2 of the algorithm that

$$x_{A^2}^1 \leq (Mx^1 + q)_{A^2} \quad \text{and} \quad x_{I^2}^1 \geq (Mx^1 + q)_{I^2} \quad (4.4)$$

$$x_{A^3}^2 \leq (Mx^2 + q)_{A^3} \quad \text{and} \quad x_{I^3}^2 \geq (Mx^2 + q)_{I^3}. \quad (4.5)$$

Using (2.1)₁, there holds

$$x^2 - x^1 = \begin{pmatrix} 0_{A^1 \cap A^2 \cap A^3} \\ 0_{A^1 \cap A^2 \cap I^3} \\ x_{A^1 \cap I^2 \cap A^3}^2 \\ x_{A^1 \cap I^2 \cap I^3}^2 \\ 0_{I^1 \cap A^2 \cap A^3} \\ 0_{I^1 \cap A^2 \cap I^3} \\ x_{I^1 \cap I^2 \cap A^3}^2 \\ x_{I^1 \cap I^2 \cap I^3}^2 \end{pmatrix} - \begin{pmatrix} 0_{A^1 \cap A^2 \cap A^3} \\ 0_{A^1 \cap A^2 \cap I^3} \\ 0_{A^1 \cap I^2 \cap A^3} \\ 0_{A^1 \cap I^2 \cap I^3} \\ x_{I^1 \cap A^2 \cap A^3}^1 \\ x_{I^1 \cap A^2 \cap I^3}^1 \\ x_{I^1 \cap I^2 \cap A^3}^1 \\ x_{I^1 \cap I^2 \cap I^3}^1 \end{pmatrix} = \begin{pmatrix} 0_{A^1 \cap A^2 \cap A^3} \\ 0_{A^1 \cap A^2 \cap I^3} \\ x_{A^1 \cap I^2 \cap A^3}^2 \\ x_{A^1 \cap I^2 \cap I^3}^2 \\ -x_{I^1 \cap A^2 \cap A^3}^1 \\ -x_{I^1 \cap A^2 \cap I^3}^1 \\ (x^2 - x^1)_{I^1 \cap I^2 \cap A^3} \\ (x^2 - x^1)_{I^1 \cap I^2 \cap I^3} \end{pmatrix} \cdot \begin{matrix} [0] \\ [0] \\ [-] \\ [+] \\ [+] \\ [+] \\ [0] \\ [0] \end{matrix}$$

The extra column on the right gives the sign of each component, when appropriate; this one is deduced by arguments similar to those in the proof of lemma 4.3, that is

$$x_{A^1 \cap I^2 \cap A^3}^2 \leq (Mx^2 + q)_{A^1 \cap I^2 \cap A^3} = 0 \quad [\text{by (4.5)}_1 \text{ and (2.1)}_2]$$

$$x_{A^1 \cap I^2 \cap I^3}^2 \geq (Mx^2 + q)_{A^1 \cap I^2 \cap I^3} = 0 \quad [\text{by (4.5)}_2 \text{ and (2.1)}_2]$$

$$x_{I^1 \cap A^2}^1 \leq (Mx^1 + q)_{I^1 \cap A^2} = 0 \quad [\text{by (4.4)}_1 \text{ and (2.1)}_2].$$

On the other hand, using (2.1)₂, there holds

$$M(x^2 - x^1) = \begin{pmatrix} (Mx^2)_{A^1 \cap A^2 \cap A^3} \\ (Mx^2)_{A^1 \cap A^2 \cap I^3} \\ -q_{A^1 \cap I^2 \cap A^3} \\ -q_{A^1 \cap I^2 \cap I^3} \\ (Mx^2)_{I^1 \cap A^2 \cap A^3} \\ (Mx^2)_{I^1 \cap A^2 \cap I^3} \\ -q_{I^1 \cap I^2 \cap A^3} \\ -q_{I^1 \cap I^2 \cap I^3} \end{pmatrix} - \begin{pmatrix} (Mx^1)_{A^1 \cap A^2 \cap A^3} \\ (Mx^1)_{A^1 \cap A^2 \cap I^3} \\ (Mx^1)_{A^1 \cap I^2 \cap A^3} \\ (Mx^1)_{A^1 \cap I^2 \cap I^3} \\ -q_{I^1 \cap A^2 \cap A^3} \\ -q_{I^1 \cap A^2 \cap I^3} \\ -q_{I^1 \cap I^2 \cap A^3} \\ -q_{I^1 \cap I^2 \cap I^3} \end{pmatrix} = \begin{pmatrix} (M(x^2 - x^1))_{A^1 \cap A^2 \cap A^3} \\ (M(x^2 - x^1))_{A^1 \cap A^2 \cap I^3} \\ -(Mx^1 + q)_{A^1 \cap I^2 \cap A^3} \\ -(Mx^1 + q)_{A^1 \cap I^2 \cap I^3} \\ (Mx^2 + q)_{I^1 \cap A^2 \cap A^3} \\ (Mx^2 + q)_{I^1 \cap A^2 \cap I^3} \\ 0_{I^1 \cap I^2 \cap A^3} \\ 0_{I^1 \cap I^2 \cap I^3} \end{pmatrix} \cdot \begin{matrix} [+] \\ [+] \\ [+] \\ [-] \\ [0] \\ [0] \end{matrix}$$

The sign of the components given in the extra column on the right is justified as follows:

$$(Mx^1 + q)_{A^1 \cap I^2} \leq x_{A^1 \cap I^2}^1 = 0 \quad [\text{by (4.4)}_2 \text{ and (2.1)}_1]$$

$$(Mx^2 + q)_{I^1 \cap A^2 \cap A^3} \geq x_{I^1 \cap A^2 \cap A^3}^2 = 0 \quad [\text{by (4.5)}_1 \text{ and (2.1)}_1]$$

$$(Mx^2 + q)_{I^1 \cap A^2 \cap I^3} \leq x_{I^1 \cap A^2 \cap I^3}^2 = 0 \quad [\text{by (4.5)}_2 \text{ and (2.1)}_1].$$

Taking the Hadamard product of the two vectors now gives

$$(x^2 - x^1) \cdot M(x^2 - x^1) = \begin{pmatrix} 0_{A^1 \cap A^2 \cap A^3} \\ 0_{A^1 \cap A^2 \cap I^3} \\ -x_{A^1 \cap I^2 \cap A^3}^2 \cdot (Mx^1 + q)_{A^1 \cap I^2 \cap A^3} \\ -x_{A^1 \cap I^2 \cap I^3}^2 \cdot (Mx^1 + q)_{A^1 \cap I^2 \cap I^3} \\ -x_{I^1 \cap A^2 \cap A^3}^1 \cdot (Mx^2 + q)_{I^1 \cap A^2 \cap A^3} \\ -x_{I^1 \cap A^2 \cap I^3}^1 \cdot (Mx^2 + q)_{I^1 \cap A^2 \cap I^3} \\ 0_{I^1 \cap I^2 \cap A^3} \\ 0_{I^1 \cap I^2 \cap I^3} \end{pmatrix}, \quad \begin{matrix} [0] \\ [0] \\ [-] \\ [+] \\ [+] \\ [-] \\ [0] \\ [0] \end{matrix}$$

where the extra column on the right gives the sign of each component. Therefore, if the index set $(A^1 \cap I^2 \cap I^3) \cup (I^1 \cap A^2 \cap A^3)$ were empty, we would have $(x^2 - x^1) \cdot M(x^2 - x^1) \leq 0$, which, with the \mathbf{P} -matricity of M and (2.2), would imply that $x_2 = x_1$, in contradiction with the initial assumption. We have proven that the first index set in (4.3) is nonempty.

To show that the second index set in (4.3) is nonempty, we use the fact that the algorithm goes from x^2 to x^3 and from x^3 to x^1 . Therefore, by cycling the indices in the result just obtained, we see that $(A^2 \cap I^3 \cap I^1) \cup (I^2 \cap A^3 \cap A^1) \neq \emptyset$; this corresponds to the second index set in (4.3). By cycling the indices again, we obtain $(A^3 \cap I^1 \cap I^2) \cup (I^3 \cap A^1 \cap A^2) \neq \emptyset$; this corresponds to the third index set in (4.3). \square

Example 3.1 was actually obtained for $n = 3$, by forcing the 3 sets in (4.3) to be nonempty by setting

$$I^1 \cap A^2 \cap A^3 = \{1\}, \quad A^1 \cap I^2 \cap A^3 = \{2\}, \quad \text{and} \quad A^1 \cap A^2 \cap I^3 = \{3\},$$

which yields $I^k = \{k\}$.

Proposition 4.5 (convergence for $n = 2$) *Suppose that M is a \mathbf{P} -matrix and that $n = 2$. Then the plain Newton-min algorithm converges.*

PROOF. We know that the algorithm converges if it does not make a p -cycle, a cycle made of $p \geq 2$ distinct nodes. It cannot make a 2-cycle by lemma 4.3. By lemma 4.4, to make a 3-cycle, the three sets in (4.3) must be nonempty; but these sets are disjoint; since $n = 2$, one of them must be empty, so that the algorithm does not make a 3-cycle. Therefore, the algorithm will visit a 4th node if it has not found the solution on the first 3 visited nodes. This last node is then the solution, since the solution exist ($M \in \mathbf{P}$) and there are at most $2^n = 4$ distinct nodes ($M \in \mathbf{P}$). \square

5 Perspectives

The work presented in this paper can be pursued along at least two directions. One possibility is to better mark out the set of matrices for which the plain Newton-min method converges. According to [18; 2003, theorem 3.4] and the counter-examples of the present paper, this set contains the set \mathbf{M}_ε of matrices sufficiently near an \mathbf{M} -matrix but not all of the larger set \mathbf{P} of \mathbf{P} -matrices, see the discussion around (1.5). Such a study may result in the identification of an analytically well defined set of matrices or it may be a long process with an endless refinement. In the first case, it would be nice to see whether membership in that new matrix class can be determined in polynomial time, knowing that recognizing a \mathbf{P} -matrix is a co-NP-complete problem [12, 37].

Another possibility is to modify the algorithm to force its convergence for \mathbf{P} -matrices or an even larger class of matrices. Being able to deal with \mathbf{P} -matrices is important for at least three reasons.

First, linear complementarity problems with a \mathbf{P} -matrix are encountered in practice [35; 2004]. Next, this is exactly the class of matrices that ensure the existence and uniqueness of the solution to the LCP [34, 11], which forces us to pay attention to these matrices. Finally, the possibility to find a polynomial algorithm to solve the LCP with a \mathbf{P} -matrix still seems to be an open question. Some authors argue that such an algorithm might exist; see Morris [31; 2002], who refers to a contribution by Megiddo [29; 1988], himself citing an unpublished related note of Solow, Stone and Tovey [36; 1987]. The possibility that a modified version of the plain Newton-min algorithm might have the desired polynomiality property cannot be excluded. A natural remedy would be to add a globalization technique (linesearch or trust regions, see [9, 5] for example) to the plain Newton-min algorithm in order to force its convergence; see [17, 19; 1990, 2009], for contributions along that direction. This globalization is not straightforward since the Newton-min direction may not be a descent direction of the standard ℓ_2 merit function associated with (1.4) [2].

Acknowledgments

We would like to thank P. Knabner and S. Kräutle for having drawn our attention on the Newton-min algorithm and its efficiency on practical problems [6, 26; 2009, 2010], and M. Hintermüller, Ch. Kanzow, N. Metla [30; 2008], and A. Griewank for various exchanges of mails and conversations on the topics. Also, we wish to thank the two referees for their helpful comments. This work was partially supported by the MoMaS group (PACEN/CNRS, ANDRA, BRGM, CEA, EDF, IRSN).

References

- [1] M. Aganagić (1984). Newton's method for linear complementarity problems. *Mathematical Programming*, 28, 349–362. 4
- [2] I. Ben Gharbia, J.Ch. Gilbert (2011). In preparation. 17
- [3] 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. 4
- [4] M. Bergounioux, K. Ito, K. Kunisch (1999). Primal-dual strategy for constrained optimal control problems. *SIAM Journal on Control and Optimization*, 37, 1176–1194. 4, 13
- [5] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, C. Sagastizábal (2006). *Numerical Optimization – Theoretical and Practical Aspects* (second edition). Universitext. Springer Verlag, Berlin. 17
- [6] H. Buchholzer, Ch. Kanzow, P. Knabner, S. Kräutle (2009). Solution of reactive transport problems including mineral precipitation-dissolution reactions by a semismooth Newton method. Technical Report 288, Institute of Mathematics, University of Würzburg, Würzburg. 17
- [7] R. Chandrasekaran (1970). A special case of the complementary pivot problem. *Opsearch*, 7, 263–268. 4
- [8] F.H. Clarke (1983). *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York. 4
- [9] A.R. Conn, N.I.M. Gould, Ph.L. Toint (2000). *Trust-Region Methods*. MPS-SIAM Series on Optimization 1. SIAM and MPS, Philadelphia. 17
- [10] R.W. Cottle, G.B. Dantzig (1968). Complementarity pivot theory of mathematical programming. *Linear Algebra and its Applications*, 1, 103–125. 3
- [11] R.W. Cottle, J.-S. Pang, R.E. Stone (2009). *The linear complementarity problem*. Classics in Applied Mathematics 60. SIAM, Philadelphia, PA, USA. 3, 5, 17
- [12] G.E. Coxson (1994). The P -matrix problem is co-NP-complete. *Mathematical Programming*, 64, 173–178. 16
- [13] F. Facchinei, J.-S. Pang (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems* (two volumes). Springer Series in Operations Research. Springer. 3

-
- [14] M. Fiedler, V. Pták (1962). On matrices with nonpositive off-diagonal elements and principal minors. *Czechoslovak Mathematics Journal*, 12, 382–400. [5](#)
- [15] A. Fischer, C. Kanzow (1996). On finite termination of an iterative method for linear complementarity problems. *Mathematical Programming*, 74, 279–292. [4](#)
- [16] G.H. Golub, C.F. Van Loan (1996). *Matrix Computations* (third edition). The Johns Hopkins University Press, Baltimore, Maryland. [11](#)
- [17] 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. [17](#)
- [18] 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. [4](#), [5](#), [6](#), [12](#), [16](#)
- [19] K. Ito, K. Kunisch (2009). On a semi-smooth Newton method and its globalization. *Mathematical Programming*, 118, 347–370. [17](#)
- [20] Ch. Kanzow (2004). Inexact semismooth Newton methods for large-scale complementarity problems. *Optimization Methods and Software*, 19, 309–325. [4](#), [6](#)
- [21] N. Karmarkar (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, 4, 373–395. [3](#)
- [22] R.B. Kellogg (1972). On complex eigenvalues of M and P matrices. *Numerische Mathematik*, 19, 170–175. [11](#)
- [23] 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-Verlag, Berlin. [3](#)
- [24] 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. [4](#)
- [25] M.M. Kostreva (1976). *Direct algorithms for complementarity problems*. PhD Thesis, Rensselaer Polytechnic Institute, Troy, New York. [4](#)
- [26] S. Kräutle (2010). The semismooth Newton method for multicomponent reactive transport with minerals. Technical report, Department of Mathematics, University of Erlangen-Nuremberg, Erlangen, Germany. [17](#)
- [27] C.E. Lemke (1965). Bimatrix equilibrium points and mathematical programming. *Management Science*, 11, 681–689. [3](#)
- [28] O.L. Mangasarian (1977). Solution of symmetric linear complementarity problems by iterative methods. *Journal of Optimization Theory and Applications*, 22, 465–485. [4](#)
- [29] N. Megiddo (1988). A note on the complexity of P -matrix LCP and computing an equilibrium. Technical Report RJ 6439 (62557). IBM Research, Almaden Research Center, 650 Harry Road, San Jose, CA, USA. [17](#)
- [30] N. Metla (2008). *The Sequential Quadratic Programming Method for Elliptic Optimal Control Problems with Mixed Control-State Constraints*. PhD Thesis, Johann Radon Institute for Computational and Applied Mathematics, Johannes Kepler Universität, Linz, Austria. [17](#)
- [31] W. Morris (2002). Randomized pivot algorithms for P -matrix linear complementarity problems. *Mathematical Programming*, 92A, 285–296. [6](#), [7](#), [17](#)
- [32] K.G. Murty (1988). *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, Berlin. [3](#)
- [33] L. Qi (1993). Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research*, 18, 227–244. [4](#)
- [34] 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. [3](#), [17](#)
- [35] U. Schäfer (2004). A linear complementarity problem with a P -matrix. *SIAM Review*, 46, 189–201. [17](#)

- [36] D. Solow, R. Stone, C.A. Tovey (1987). Solving LCP on P -matrices is probably not NP-hard. Unpublished note. [17](#)
- [37] P. Tseng (2000). Co-NP-completeness of some matrix classification problems. *Mathematical Programming*, 88, 183–192. [16](#)



Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399