INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix

Ibtihel Ben Gharbia — J. Charles Gilbert

N° 7160

19 décembre 2009

Thème NUM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7160--FR+ENG

Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix

Ibtihel Ben Gharbia[†], J. Charles Gilbert[†]

Thème NUM — Systèmes numériques
Projet Estime

Rapport de recherche n° 7160 — 19 décembre 2009 — 16 pages

Abstract: The plain Newton-min algorithm for solving the linear complementarity problem (LCP for short) $0 \leq x \perp (Mx + q) \geq 0$ can be viewed as a nonsmooth 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 M-matrix of order n, the algorithm is known to converge in at most n iterations. We show in this note that this result no longer holds when M is a P-matrix of order ≥ 3, since then the algorithm may cycle. 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 P-matrix of order 1 or 2.

Key-words: linear complementarity problem, Newton's method, nonconvergence, nonsmooth function, P-matrix.

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

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Non convergence de l'algorithme de Newton-min simple pour les problèmes de complémentarité linéaires avec P-matrice

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 cette note que ce résultat ne tient plus lorsque M est une P-matrice d'ordre $n \geq 3$; l'algorithme peut en effet cycler 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, P-matrice, problème de complémentarité linéaire.

Table of contents

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

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

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 [19, 6], i.e., a matrix with positive principal minors: $\det M_{II} > 0$ for all nonempty $I \subset \{1, \ldots, n\}$. Other classes of matrices M intervening in the discussion below are the one of Z-matrices (which have nonpositive off-diagonal elements: $Mij \leq 0$ for all $i \neq j$) and M-matrices (which are at once P- and Z-matrices; they are called K-matrices in [1, 6]).

Since the components of x and $Mx+q$ must be nonnegative in $\mathrm{LC}(M,q)$, the 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

$$x \cdot (Mx+q) = 0. \tag{1.1}$$

Recall that the Hadamard product $u \cdot v$ of two vectors u and v is the vector having its ith 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 matrix M 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 $\mathrm{LC}(M,q)$ is therefore a feasible node.

Many algorithms have been proposed to solve problem $\mathrm{LC}(M,q)$ [6]. They may be based on pivoting techniques [5], 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 [14, 1984] (see also [16, 1991] for one of the first accounts on the use of interior point methods for solving linear complementarity problems), and on nonsmooth Newton approaches [8], as the one considered here.

The algorithm we consider in this note maintains the complementarity condition (1.1), while feasibility is obtained at convergence. 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. Then there are two complementary subsets A and I of $\{1, \ldots, n\}$, such that $x_A = 0$ and $(Mx+q)_I = 0$ (A and I are not necessarily uniquely determined from x). The algorithm first defines the index sets A^+ and I^+ of the next iterate x^+; in its simplest form, it takes

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

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

$$x^+_{A^+} = 0 \qquad \text{and} \qquad (Mx^+ + q)_{I^+} = 0. \tag{1.3}$$

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^+: all the principal submatrices M_{II}, with $I \subset \{1, \dots, n\}$, must be nonsingular. The definition of the index sets A^+ and I^+ by (1.2) is partly motivated by the fact that it implies $x^+ = x$ if and only if x is a solution (lemma 4.1). Note that the algorithm has a principle quite different from the one used by an interior point approach, which generates strictly feasible point, while complementarity is obtained at the limit. In the local analysis of the method, it is important to allow the first iterate x^1 not to be a node, but in this paper x^1 will always be assumed to be a node. 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 (the algorithm is a Markov process).

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

$$\min(x, Mx + q) = 0, \tag{1.4}$$

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 $\mathrm{LC}(M, q)$. 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 $x_i = (Mx + q)_i = 0$) 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 [4] of the function in (1.4). This description makes it natural to call *Newton-min* the algorithm that updates x by the formulas (1.2)-(1.3).

The algorithm sketched above and that we further explore in this paper can be traced back at least to the work of Aganagić [1, 1984], who proposed a Newton-type algorithm for solving $\mathrm{LC}(M, q)$ when M is a *hidden Z-matrix*, which is a matrix that can be written $MX = Y$ where X and Y are Z-matrices having particular properties (an M-matrix is a particular instance of hidden Z-matrix). In his approach, the nonsmooth equation (1.4) to solve is then expressed in terms of a Z-matrix associated with M; a convergence result is established. The algorithm was then rediscovered in the form (1.2)–(1.3) by Bergounioux, Ito, and Kunisch [2, 1999] for solving quadratic optimal control problems under the name of *primal-dual active set strategy* (see also [12, 2008]). Hintermüller [10, 2003] uses the same algorithm for solving constrained optimal control problems, shows that the method does not cycle in the case of bilateral constraints, and proves convergence in finite and infinite dimension. It is shown in [11, 2003] that the algorithm can be viewed as a nonsmooth Newton method for solving (1.4), which motivates the name of the algorithm given in this note, and that it converges locally superlinearly if M is a P-matrix. Kanzow [13, 2004] proved its convergence in at most n iterations when M is an M-matrix.

This paper presents examples of nonconvergence of the Newton-min algorithm when M is a P-matrix. These counter-examples hold for the undamped Newton-min algorithm. One may believe that, as a Newton-like method for solving 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, as mentionned above, to be convergent without globalization techniques when M is an M-matrix [13, 2004]. Therefore, searching the weakest assumptions for which these convergence properties hold seems to us a valid question. The examples in this paper show that it is not enough to require the 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 being a slightly more flexible than in the description (1.2)–(1.3) above. Some elementary properties of the algorithm are also given. Section 3 describes and analyses the examples of nonconvergence of the plain Newton-min algorithm with a 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. In section 4, the plain Newton-min algorithm is shown to converge for a P-matrix when $n = 1$ or $n = 2$, a crumb of consolation. The paper concludes with a perspective section.

2 The algorithm

The Newton-min algorithm described in this section generates points that satisfy the complementarity conditions in (1.1), while the nonnegativity conditions in $\mathrm{LC}(M, q)$ 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 $k = 2, 3, \ldots$, do the following.

1. If x^{k-1} is a solution to $\mathrm{LC}(M, q)$, 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

$$\left\{ \begin{array}{l} x_{A^k}^k = 0 \\ (Mx^k + q)_{I^k} = 0. \end{array} \right. \tag{2.1}$$

The algorithm is well defined if all the principal submatrices of M are nonsingular. This assumption will be generally reinforced by supposing the P-matricity of M. We recall [6] that M is a P-matrix if and only if

$$\text{any } x \text{ verifying } x \cdot (Mx) \leq 0 \text{ vanishes.} \tag{2.2}$$

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. Actually, this choice has no impact on the counter-examples given in this paper, since in them the algorithm does not generate iterates with doubly active indices.

In this paper, we always assume the x^1 is a node, then there are two complementary subsets A_1 and I_1 of $\{1, \ldots, n\}$, 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 [11], it is better not to make this assumption.

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

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

Therefore, as observed in [13, 2004, page 317], there must hold

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

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).

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 $n \geq 3$. We start with the case when n is odd and provide an example of P-matrix M, vector q, and starting point for which the algorithm makes cycles having n nodes. Next, we consider the case when n is even and construct another class of matrices, which can be viewed as perturbation of the matrices in the first example, for which cycles having n nodes is also possible.

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

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

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 $q \geq 0$, the solution of LC(M, q) for these data is $x = 0$. □

The matrix M and the vector q in example 3.1 have already been used by Morris [18] (in that paper, $\alpha = 2$, M is the transposed of the one here, and $q = -\mathbf{1}$), although we arrived to them in a different manner, as explained in section 4. For completeness and precision, we start by studying the P-matricity of the matrix M in example 3.1 (the result for n odd and $\alpha = 2$ was already claimed in [18] without a detailed proof).

Lemma 3.2 *Consider the matrix M in example 3.1. If n is even, M is a P-matrix if and only if $|\alpha| < 1$. If n is odd, M is a P-matrix 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 is not a P-matrix.

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

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

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

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

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 P-matricity now follows from (2.2).

Suppose now that n is even. If $\alpha \geq 1$, the nonzero vector x, defined by $x_i = 1$ for i odd and by $x_i = -1$ for i even, is such that $x \cdot (Mx) = (1 - \alpha)\mathbf{1} \leq 0$; hence M is not a P-matrix. Let us now show, by contradiction, that M is a P-matrix if $0 < \alpha < 1$, assuming that there is a nonzero x satisfying $x \cdot (Mx) \leq 0$, hence (3.1). 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 $i = 1, \ldots, n-1$:

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

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 the P-matricity of M, we argue by contradiction, assuming that there is a nonzero x such that $x \cdot (Mx) \leq 0$, which yields (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 of the x_i's:

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

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 is a P-matrix by (2.2). □

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

Proof. Let us show by induction that

$$I^k = \{k\}, \quad \text{for } k = 1, \ldots, n.$$

By assumption, this statement holds for $k = 1$. Suppose now that it holds for some $k \in \{1, \ldots, n-1\}$ and let us show that it holds for $k+1$. From $I^k = \{k\}$, one deduces that

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

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

Now from $I^n = \{n\}$, one deduces that

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

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

By combining lemmas 3.2 and 3.3, one can see that algorithm 2.1 may cycle when $n \geq 3$ is odd. When n is even, the condition $\alpha > 1$ used in lemma 3.3 prevents the matrix M from being a P-matrix. It is not difficult to show, however, that the algorithm can also cycle when n is even, $n \geq 4$, and M is a P-matrix, by using the construction of the proof of proposition 3.7: the cycle visits an odd number of nodes (hence $< n$).

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 oder 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

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

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 $q \geq 0$, the solution of $\mathrm{LC}(M, q)$ for these data is $x = 0$. □

Conditions for having an n-node cycle with algorithm 2.1 on problem $\mathrm{LC}(M, q)$ 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 $\mathrm{LC}(M, q)$ in which M and q are given in example 3.4 with $\alpha > 1$ and $\beta < 1$. When applied to that problem $\mathrm{LC}(M, q)$, and started from $I^1 = \{1\}$, algorithm 2.1 cycles by visiting in order the n nodes defined by $I^k = \{k\}$, $k = 1, \dots, n$.*

PROOF. The proof is quite similar to the one of lemma 3.3, so that we only sketch it. When $I^k = \{k\}$, $1 \leq k \leq n-2$, there holds

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

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

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

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

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

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

We have now to examine whether the conditions on α and β given by lemma 3.5 are compatible with the P-matricity of the matrix M in example 3.4. Actually, the conditions on α and β ensuring the 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. Figure 3.1 shows in gray-blue the intersection with the box $[-2,2] \times [-2,2]$ of the regions $\mathcal{P}$ formed

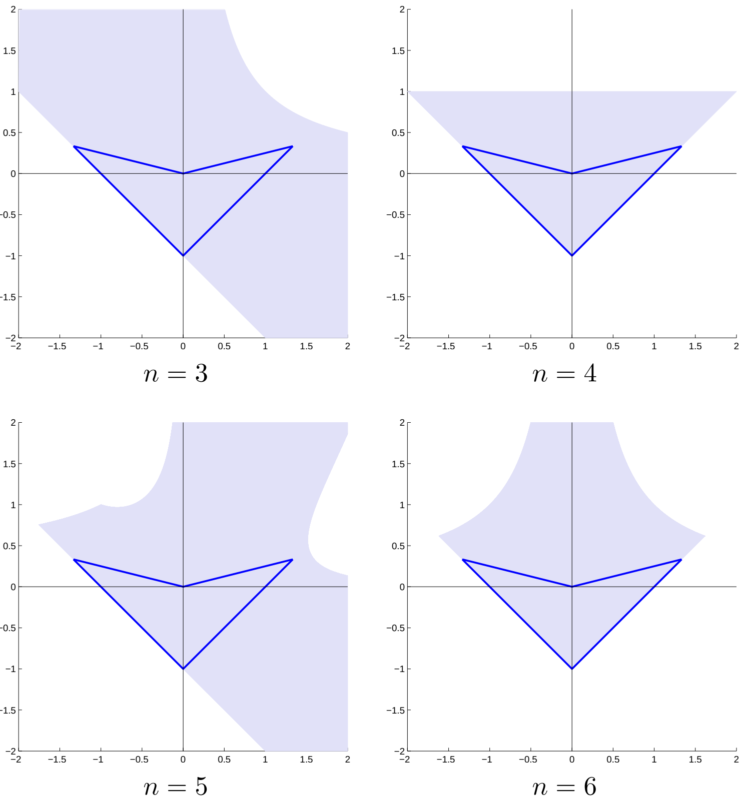

Figure 3.1: Regions formed of the (α, β) pairs in $[-2,2] \times [-2,2]$ for which the matrix M of example 3.4 is a P-matrix, when its order is $n = 3$, 4, 5, and 6; these regions are nonconvex but star-shaped with respect to $(0,0)$, which is the point corresponding to the identity matrix. According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

of the (α, β) pairs for which the matrix M is a P-matrix, when its order is $n = 3$, 4, 5, and 6. Of course, by lemma 3.2, the regions $\mathcal{P}$ contain 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 and that the regions $\mathcal{P}$ always contain the nonconvex polyhedron represented in figure 3.1, which is independent of n.

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

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

where I_n 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 [9, formula (4.7.10)] that J is diagonalizable, meaning that there is a diagonal matrix D and a nonsingular matrix P such that $J = PDP^{-1}$; in addition its eigenvalues are the nth roots of unity: $D := \mathrm{Diag}(1, w^1, \ldots, w^{n-1})$, where $w = \mathrm{e}^{2i\pi/n}$.

Lemma 3.6 *The set of (α, β) pairs ensuring the P-matricity of M in example 3.4 contains the set $\{(\alpha, \beta) : |\alpha| - 1 < \beta < |\alpha|/4\}$.*

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

$$|\alpha| - 1 < \beta < \frac{|\alpha|}{4}, \tag{3.2}$$

the matrix $M + M^\top$ is positive definite, since then M is clearly a P-matrix [15, p. 175].

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

$$M + M^\top = P\left(2I_n + \alpha D + \beta D^2 + \beta D^{n-2} + \alpha D^{n-1}\right)P^{-1},$$

This identity shows that the eigenvalues of the symmetric matrix $M + M^\top$ are the (necessarily real) numbers $\lambda_k := 2 + \alpha e^{2ik\pi/n} + \beta e^{4ik\pi/n} + \beta e^{2ik(n-2)\pi/n} + \alpha e^{2ik(n-1)\pi/n}$, for $k = 0, \ldots, n-1$. Using $e^{2ikp\pi/n} + e^{2ik(n-p)\pi/n} = 2\cos(2kp\pi/n)$ (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 the conditions that ensure the positivity of φ on $[-1, 1]$.

- Case $\beta = 0$. Then the condition is $|\alpha| < 1$.

- Case $\beta > 0$. Then, $t_- \leq t_+$ and the condition is either $1 < t_-$, which is equivalent to $-\alpha - 1 < \beta < -\alpha/4$, or $t_+ < -1$, which is equivalent to $\alpha - 1 < \beta < \alpha/4$.
- Case $\beta < 0$. Then, $t_+ \leq t_-$ and the conditions are both $t_+ < -1$ and $1 < t_-$, which are equivalent to $|\alpha| - 1 < \beta < 0$.

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

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

Proof. Since the 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 $\mathrm{LC}(M, q)$ with M and q given by example 3.1, and $\alpha > 1$; then use lemma 3.2, which shows that M is a P-matrix, and lemma 3.3, which shows that a cycle is possible.

When $n \geq 4$ is even, a cycle made of n nodes is possible on problem $\mathrm{LC}(M, q)$ with M and q given by example 3.4, and α and β satisfying $\alpha > 1$ and $\alpha - 1 < \beta < \alpha/4$; then use lemma 3.5, which shows that a cycle is possible, and lemma 3.6, which shows that M is a P-matrix.

When $n \geq 3$ and $p \in \{3, \dots, n\}$, consider a $p \times p$ P-matrix $\tilde{M}$, a vector $\tilde{q} \in \mathbb{R}^p$, and a starting point $\tilde{x}^0 \in \mathbb{R}^p$, such that algorithm 2.1 applied to problem $\mathrm{LC}(\tilde{M}, \tilde{q})$ and starting at $\tilde{x}^0$ 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

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

The P-matricity of M is clear, by observing that $x \cdot (Mx) \leq 0$ implies $x = 0$. Denote by x^k the iterates generated by algorithm 2.1 on $\mathrm{LC}(M, q)$ 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 form also a cycle made of p nodes. □

To conclude this section and to make the discussion more concrete, we provide two examples of 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 $x^1 = (-1, 0, \dots, 0)$ for solving problem $\mathrm{LC}(M_n, \mathbf{1})$:

$$M_3 := \begin{pmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix} \qquad \text{and} \qquad M_4 := \begin{pmatrix} 1 & 0 & 1/4 & 7/6 \\ 7/6 & 1 & 0 & 1/4 \\ 1/4 & 7/6 & 1 & 0 \\ 0 & 1/4 & 7/6 & 1 \end{pmatrix}. \tag{3.3}$$

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.

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 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 [2, 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 all the principal minors of M do not vanish. Then, a node is a solution to* $\mathrm{LC}(M, q)$ *if and only if the plain Newton-min algorithm, without its step 1, starting at that node, takes the same node as the next iterate.*

PROOF. Let x be a node of problem $\mathrm{LC}(M, q)$. 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 $0 = (M(x^+ - x))_{I^+} = M_{I^+I^+}(x^+ - x)_{I^+} = 0$ [since $(x^+ - x)_{A^+} = 0$] and $(x^+ - x)_{I^+} = 0$ [since $M_{I^+I^+}$ is nonsingular]. 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. □

Proposition 4.2 (convergence for $n = 1$) *Suppose that M is a P-matrix and that $n = 1$. 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 (since M is a P-matrix). 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. □

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 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 $x^1 \to x^2 \to x^1$, 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

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

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

After possible rearrangement of the component order, we get

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

Observe that the components of $x^2 - x^1$ with indices in $A^1 \cap I^2$ are nonpositive since $x^2_{A^1\cap I^2} \leq (Mx^2+q)_{A^1\cap I^2}$ [by $(4.2)_1$] $= 0$ [by $(2.1)_2$] and that the components with indices in $I^1 \cap A^2$ are nonnegative since $-x^1_{I^1\cap A^2} \geq -(Mx^1+q)_{I^1\cap A^2}$ [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 $-(Mx^1+q)_{A^1\cap I^2} \geq -x^1_{A^1\cap I^2}$ [by $(4.1)_2$] $= 0$ [by $(2.1)_1$] and the components with indices in $I^1 \cap A^2$ are nonpositive since $(Mx^2+q)_{I^1\cap A^2} \leq x^2_{I^1\cap A^2}$ [by $(4.2)_2$] $= 0$ [by $(2.1)_1$]. Therefore

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

Since M is a P-matrix, there holds $x^1 = x^2$ by (2.2), contradicting the initial assumption. □

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

$$\left\{ \begin{array}{l} (A^1\cap I^2\cap I^3)\cup(I^1\cap A^2\cap A^3) \\ (I^1\cap A^2\cap I^3)\cup(A^1\cap I^2\cap A^3) \\ (I^1\cap I^2\cap A^3)\cup(A^1\cap A^2\cap I^3). \end{array} \right. \tag{4.3}$$

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

$$x^1_{A^2} \leq (Mx^1+q)_{A^2} \quad \text{and} \quad x^1_{I^2} \geq (Mx^1+q)_{I^2} \tag{4.4}$$
$$x^2_{A^3} \leq (Mx^2+q)_{A^3} \quad \text{and} \quad x^2_{I^3} \geq (Mx^2+q)_{I^3}. \tag{4.5}$$

Using $(2.1)_1$, there holds

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

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

$$\begin{aligned}
&x^2_{A^1\cap I^2\cap A^3} \leq (Mx^2+q)_{A^1\cap I^2\cap A^3} = 0 && \text{[by } (4.5)_1 \text{ and } (2.1)_2\text{]}\\
&x^2_{A^1\cap I^2\cap I^3} \geq (Mx^2+q)_{A^1\cap I^2\cap I^3} = 0 && \text{[by } (4.5)_2 \text{ and } (2.1)_2\text{]}\\
&x^1_{I^1\cap A^2} \leq (Mx^1+q)_{I^1\cap A^2} = 0 && \text{[by } (4.4)_1 \text{ and } (2.1)_2\text{].}
\end{aligned}$$

On the other hand, using $(2.1)_2$, there holds

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

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

$$\begin{aligned}
&(Mx^1+q)_{A^1\cap I^2} \leq x^1_{A^1\cap I^2} = 0 && \text{[by } (4.4)_2 \text{ and } (2.1)_1\text{]}\\
&(Mx^2+q)_{I^1\cap A^2\cap A^3} \geq x^2_{I^1\cap A^2\cap A^3} = 0 && \text{[by } (4.5)_1 \text{ and } (2.1)_1\text{]}\\
&(Mx^2+q)_{I^1\cap A^2\cap I^3} \leq x^2_{I^1\cap A^2\cap I^3} = 0 && \text{[by } (4.5)_2 \text{ and } (2.1)_1\text{].}
\end{aligned}$$

Taking the Hadamard product of the two vectors now gives

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

where the extra column on the right gives the sign of each components. 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 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 $(A^2\cap I^3\cap I^1)\cup(I^2\cap A^3\cap A^1)\neq\varnothing$; this corresponds to the second index set in (4.3). By cycling the indices again, we obtain $(A^3\cap I^1\cap I^2)\cup(I^3\cap A^1\cap A^2)\neq\varnothing$; this corresponds to the third index set in (4.3). □

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 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 is a P-matrix) and there are at most $2^n = 4$ distinct nodes (M is a P-matrix). □

5 Perspective

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 [13, theorem 3.2] and the counter-examples of the present paper, this set is located between the one of M-matrices and the larger one of P-matrices. 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 by bracketing sets, whose extreme points are now known to be the M-matrices and P-matrices. In the first case, it would be nice to see whether membership to that new matrix class can be determined in polynomial time, knowing that recognizing a P-matrix is a co-NP-complete problem [7].

Another possibility is to modify the algorithm to force its convergence for P-matrices or an even larger class of matrices. Being able to deal with P-matrices is important for at least two reasons. On the one hand, this is exactly the class of matrices that ensure the existence and uniqueness of the solution to the LCP [19, 6], which forces us to pay attention to these matrices. On the other hand, the possibility to find a polynomial algorithm for solving the LCP with a P-matrix still seems to be an open question. Some authors argue that such an algorithm might exist; see Morris [18], who refers to a contribution by Megiddo [17], himself citing an unpublished related note of Solow, Stone and Tovey [20]. The possibility that a modified version of the 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) to the plain Newton-min algorithm in order to force its convergence. It can be shown, indeed, that this algorithm is a descent method on the ℓ_2 merit function, provided the set A_1^k in step 2 of algorithm 2.1 is carefully chosen.

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 [3], and M. Hintermüller and Ch. Kanzow for various exchanges of mails and conversations on the topics. 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. 3, 4

[2] 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, 12

[3] 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. 15

[4] F.H. Clarke (1983). *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York. 4

[5] R.W. Cottle, G.B. Dantzig (1968). Complementarity pivot theory of mathematical programming. *Linear Algebra and its Applications*, 1, 103–125. 3

[6] 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, 15

[7] G.E. Coxson (1994). The P-matrix problem is co-NP-complete. *Mathematical Programming*, 64, 173–178. 15

[8] F. Facchinei, J.-S. Pang (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems* (two volumes). Springer Series in Operations Research. Springer. 3

[9] G.H. Golub, C.F. Van Loan (1996). *Matrix Computations* (third edition). The Johns Hopkins University Press, Baltimore, Maryland. 10

[10] M. Hintermüller (2003). A primal-dual active set algorithm for bilaterally control constrained optimal control problems. *Quaterly of Applied Mathematics*, 1, 131–160. 4

[11] 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

[12] K. Ito, K. Kunisch (2008). *Lagrange Multiplier Approach to Variational Problems and Applications*. Advances in Design and Control. SIAM Publication, Philadelphia. 4

[13] Ch. Kanzow (2004). Inexact semismooth Newton methods for large-scale complementarity problems. *Optimization Methods and Software*, 19, 309–325. 4, 5, 15

[14] N. Karmarkar (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, 4, 373–395. 3

[15] R.B. Kellogg (1972). On complex eigenvalues of M and P matrices. *Numerische Mathematik*, 19, 170–175. 10

[16] 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

[17] 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. 15

[18] W. Morris (2002). Randomized pivot algorithms for P-matrix linear complementarity problems. *Mathematical Programming*, 92A, 285–296. 6, 15

[19] 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, 15

[20] D. Solow, R. Stone, C.A. Tovey (1987). Solving LCP on P-matrices is probably not NP-hard. Unpublished note. 15

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

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

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