IJMMS 29:2 (2002) 99–113
PII. S0161171202007056
http://ijmms.hindawi.com

PROJECTIVE ALGORITHMS FOR SOLVING COMPLEMENTARITY PROBLEMS

CAROLINE N. HADDAD and GEORGE J. HABETLER

Received 16 February 2001

We present robust projective algorithms of the von Neumann type for the linear complementarity problem and for the generalized linear complementarity problem. The methods, an extension of Projections Onto Convex Sets (POCS) are applied to a class of problems consisting of finding the intersection of closed nonconvex sets. We give conditions under which convergence occurs (always in 2 dimensions, and in practice, in higher dimensions) when the matrices are P-matrices (though not necessarily symmetric or positive definite). We provide numerical results with comparisons to Projective Successive Over Relaxation (PSOR).

2000 Mathematics Subject Classification: 15A57, 65F99, 65K99, 90C33.

1. Introduction. Linear complementarity problems have been seen to arise in many areas. More specifically, equilibria in: Economics: Walrus law (Mathiesen [15], Tobin [20]), Leontief model (Ebiefung and Kostreva [6]), bearing lubrication (Cryer [3]), Mathematical Biology and Lotka-Volterra Systems (Takeuchi [19], Habetler and Haddad [9]). Applications of generalized linear complementarity problems include Bearing lubrication (Oh [17]), Mathematical Biology and Lotka-Volterra Systems (Habetler and Haddad [10]), Engineering and Economic Applications (Ferris and Pang [7], Vandenberghe et al. [21], Dirkse and Ferris [5]). In addition to being extensions of linear complementarity problems, these problems also have a rather useful geometrical interpretation in that they may be thought of as finding the solutions of piecewise linear systems, a generalization of linear systems. It was in considering them in this light, and in trying to find classes of nonconvex set (a piecewise linear function is a nonconvex set made up of convex pieces) where the method of Projections Onto Convex Sets (POCS) might work, that we discovered some simple projective algorithms for finding solutions of both linear complementarity problems (LCPs) and generalized linear complementarity problems (GLCPs).

In Section 2, we introduce the background necessary to the problem. In Sections 3 and 4, we present the projection algorithms for the LCP and in Section 5, the extension to the GLCP. Proofs of convergence for the 2-dimensional case are based on the geometry of the problem and fixed point arguments. They have been done in the thesis of Haddad [12] and are not included here. In practice, these algorithms work on higher-dimensional problems and always converge to the solution if the matrices involved are P-matrices (where a unique solution is guaranteed [1, 18] and there is no need to worry about cycling, i.e., iterations projecting between two or more vectors that are nonsolutions, [13]) independent of the choice of starting vector. They

also seem to work in cases where the matrices are not P-matrices (where a solution exists), as well, but not necessarily in every situation. When they do not work, they will be seen to cycle. Many other schemes, such as Cryer's mentioned below, require a strict subclass of P-matrices to guarantee convergence. In Section 6, we discuss ways of speeding up the convergence (which can be slow). Finally, in Section 7, we compare the method to another well-known robust iterative scheme, *Cryer's Projective Successive Over-Relaxation method (PSOR)* [3, 4]. We have coded and tested the algorithms (initially in FORTRAN and later as MATLAB M-files) and provide convergence results for several types of matrices. We include a pathological example where our method converges very quickly while it would take on the order of 2^n pivots for convergence to occur if one was using a pivoting scheme such as *Lemke's* [14] or *Murty's* [16] methods. In Section 8, we address possible future directions for research.

2. Preliminaries. Consider the linear complementarity problem: given A is an $n\times n$ real matrix, $\mathbf{b}\in\mathbb{R}^n$, find $\mathbf{w},\mathbf{x}\in\mathbb{R}^n$ such that

$$\mathbf{w}\geq 0,\ \mathbf{x}\geq 0,\quad \mathbf{w}=A\mathbf{x}-\mathbf{b},\quad \mathbf{w}^T\mathbf{x}=0\quad (\text{or } w_ix_i=0,\ i=1,\dots,n). \tag{2.1}$$

We will denote the above problem by LCP$(A,\mathbf{b})$.

Definition 2.1. An $n\times n$ matrix A is said to be a P-matrix ($A\in P$) if all of its principal minors are positive. $A\in P_o$ if all of its principal minors are nonnegative.

Result 2.2. Let A be an $n\times n$ matrix. Then LCP(A,b) has a unique solution for every $\mathbf{b}\in\mathbb{R}^n$ if and only if A is a P-matrix.

(See Cottle et al. [2, Theorem 3.37] and the discussion therein.)

We will assume that the matrix A has its rows normalized. Certainly if A is P, then it can be so normalized.

Definition 2.3. *A vertical block matrix* N of dimension $m\times n$, $m\geq n$, is said to be of *type* $(m_1,\dots,m_n)$ if it is partitioned row-wise into n blocks,

$$N=\begin{bmatrix}N^1\\ \vdots\\ N^n\end{bmatrix}, \tag{2.2}$$

where the jth block, N^j, is of dimension $m_j\times n$, and $m=\sum_{j=1}^n m_j$.

The generalized linear complementarity problem may be stated as follows: given N is an $m\times n$, real vertical block matrix, $m\geq n$, of type $(m_1,\dots,m_n)$, $\mathbf{b}\in\mathbb{R}^m$, find $\mathbf{w}\in\mathbb{R}^m$, $\mathbf{x}\in\mathbb{R}^n$ such that

$$\begin{gathered}\mathbf{w}=N\mathbf{x}-\mathbf{b},\quad \mathbf{w}\geq 0,\ \mathbf{x}\geq 0,\\ x_i\prod_{j=1}^{n}w_i^j=0,\quad i=1,\dots,n,\ w_i^j,\ \text{the } i\text{th entry of } \mathbf{w} \text{ from the } j\text{th block of } N.\end{gathered} \tag{2.3}$$

We will denote the above problem by GLCP(N,b).

DEFINITION 2.4. Let N be a vertical block matrix of type $(m_1,\dots,m_n)$. A submatrix A of N of size n is called a *representative submatrix* if its jth row is drawn from the jth block, N^j, of N. Note that there are $\prod_{j=1}^{n} m_j$ representative submatrices. Let this $n\times n$ representative submatrix of N be labeled $A^{(l)}$ with $\sqrt{\sum_{j=1}^{n} a_{kj;l}^2} = 1$ for each $k = 1,\dots,n$ and $l = 1,\dots,\prod_{j=1}^{n} m_j$. Furthermore, we assume that $\mathbf{b}$ is compatibly organized with entries, $b_{k;l}$, respectively.

RESULT 2.5. Let N be an $m\times n$ real vertical block matrix, $m \geq n$, of type $(m_1,\dots,m_n)$. Then GLCP$(N,\mathbf{b})$ has a unique solution for every $\mathbf{b}\in\mathbb{R}^m$ if and only if all of the representative submatrices of N are P-matrices.

(See Habetler and Szanc [11], and Cottle and Dantzig [1].)

DEFINITION 2.6. Let N be an $m\times n$, $m \geq n$, real vertical block matrix of type $(m_1,\dots,m_n)$. A *principal submatrix* of N is a principal submatrix of some representative submatrix and its determinant is referred to as a *principal minor*.

DEFINITION 2.7. The region defined by the complementarity problem where $x_k \geq 0$, $w_k^i \geq 0$, $i = 1,\dots,m_1$ for $k = 1$ and $i = 1+\sum_{q=1}^{k-1} m_q,\dots,\sum_{q=1}^{k} m_q$ for $k > 1$ will be referred to as the *kth feasible region* or *kth interior*. The complement of the *kth interior* is the *kth exterior*. The intersection of all of the feasible regions is the *interior*. The complement of the *interior* will be referred to as the *exterior*.

DEFINITION 2.8. For the linear complementarity problem where A is an $n\times n$ matrix, $\mathbf{b}\in\mathbb{R}^n$, *hyperplane k* refers to the set of points $\mathbf{x}\in\mathbb{R}^n$ such that $w_k = \sum_{j=1}^{n} a_{kj}x_j - b_k = 0$, for $k = 1,\dots,n$.

DEFINITION 2.9. For the generalized linear complementarity problem, *bent hyperplane k* (BHk) refers to the union of the m_k+1 hyperplane segments that forms the boundary of the kth feasible region. The *ith segment of the kth bent hyperplane* refers to the intersection of $w_k^i = 0$ with the kth feasible region.

The following definition defines an orientation regarding the hyperplanes that will give us a unique definition for *angle* θ, and *normal* to the hyperplane (for the LCP and GLCP).

DEFINITION 2.10. Without loss of generality, we take the *normal*, $\mathbf{n}_i$, *to ith segment of the kth bent hyperplane* to point into the kth feasible region. The angle θ, formed between the $(i-1)$th and the ith segment of the kth bent plane is defined to be θ, such that

$$\cos\theta = \frac{\mathbf{n}_{i-1}^T\mathbf{n}_i}{||\mathbf{n}_{i-1}||\,||\mathbf{n}_i||}. \tag{2.4}$$

DEFINITION 2.11. A *kink* refers to a nonempty intersection of any two hyperplanes making up part of a bent hyperplane. It is a point in $\mathbb{R}^2$, a line in $\mathbb{R}^3$, and so forth.

The LCP may be restated as follows: for each $i = 1,\dots,n$, we have $x_i \geq 0$, $w_i \geq 0$, and $w_i x_i = 0$. Here, $w_i = (Ax^- b)_i$. As we saw, the boundary of the ith feasible region forms a bent hyperplane in n dimensions. The angle between the two half-spaces forming the bent hyperplane must be greater than 0° and less than or equal to 180°,

and hence the *angle* θ, is well defined. Contained in the set of possible solutions to LCP$(A,\mathbf{b})$ for given $\mathbf{b}$, is the set of intersection points of the n bent hyperplanes.

Similarly, the GLCP$(N,\mathbf{b})$ for a given $\mathbf{b}$, forms a set of n bent hyperplanes, possibly with multiple kinks, and the solution set of the GLCP$(N,\mathbf{b})$ is the set of intersection points of these n bent hyperplanes. If all the representative matrices of N are P-matrices (or in the case of LCP$(A,\mathbf{b})$, if the matrix A is a P-matrix), then the solution set consists of a single point.

Definition 2.12. Let T_k be the kth *cone* defined by $x_k \geq 0$ and $w_k \geq 0$. Then the *(negative) polar cone of* T_k, or *dual cone of* T_k is

$$T_k^- \equiv \{\mathbf{P} \in \mathbb{R}^n \mid \langle \mathbf{P},\ \mathbf{q}\rangle < 0,\ \forall \mathbf{q} \in T\}. \tag{2.5}$$

The *closed polar cone of* T_k is

$$\overline{T_k^-} \equiv \{\mathbf{P} \in \mathbb{R}^n \mid \langle \mathbf{P},\ \mathbf{q}\rangle \leq 0,\ \forall \mathbf{q} \in T\}. \tag{2.6}$$

Definition 2.13. A *projection of the point* x *onto a closed subset* C*, of a normed space* H is a point $\Pr_C \mathbf{x}$ in C, $\ni$

$$\left\|\mathbf{x} - \Pr_C \mathbf{x}\right\| = \inf_{y \in C} \|\mathbf{x} - \mathbf{y}\| = d(\mathbf{x}, C). \tag{2.7}$$

If in addition, the set is convex, then the projection is nonexpansive. If we have m closed such sets and we alternately project from one set to the next, in some fixed order, then we generate a sequence of iterates

$$\mathbf{x}^{k+1} = \Pr_m \Pr_{m-1} \cdots \Pr_1 \mathbf{x}^k, \tag{2.8}$$

where $\Pr_i$ is the projection onto the ith set. A complete set of projections from the first hyperplane to the nth hyperplane will be referred to as a *cycle* or a *cycle of iterates, not* to be confused later with iterates cycling. In general, we take $H = \mathbb{R}^n$ and $\|\cdot\|$ to be the usual Euclidean norm.

Assuming that the convex sets have a common intersection, Gubin et al. [8] showed that in $\mathbb{R}^n$ convergence of these iterates to a point in the intersection is guaranteed.

This method is referred to as POCS, *projection onto convex sets*. The nonexpansive property of the projections and the uniqueness of the projection is lost if the closed sets are not all convex or if the closed sets do not have an intersection. So, in general, the iterates will probably not converge. However, we prove in [12] that if the sets are the bent hyperplanes associated with an LCP$(A,\mathbf{b})$ with A, a P-matrix, then the method will always converge to a unique solution for $n = 2$. In practice, it converges for all n under the same conditions.

3. Algorithm I. Thus we propose the following alternating projection scheme to solve the LCP$(A,\mathbf{b})$ for A, a P-matrix. If we think of solving the LCP as finding the (unique) intersection point of n bent (nonconvex) hyperplanes, then the alternating

projection method applied here would yield the following algorithm:

(1) Choose a starting vector.
(2) Project directly onto BHi, for $i = 1,\dots,n$.
(3) Repeat until solution is reached or desired tolerance is achieved.

However, in practice it is the following.

Direct algorithm: (I) Given, A is an $n \times n$ matrix, $\mathbf{b} \in \mathbb{R}^n$. Choose a starting vector $\mathbf{x}^{(o,n)}$.

(II) Construct the alternating sequences, $\mathbf{x}^{(i,k)}$, for $i = 1,2,3,\dots$ in the following manner:

For $k = 1,2,\dots,n$

If $k = 1$

(i) Project $\mathbf{x}^{(i-1,n)}$ directly onto BH1:

If $\mathbf{x}^{(i-1,n)}$ is in the interior of BH1, then $x^{(i,1)}$ is the projection onto the nearest hyperplane segment, $x_1 = 0$ or $w_1 = 0$. If they are equidistant, then project to $x_1 = 0$.

If $\mathbf{x}^{(i-1,n)}$ is in the exterior of BH1, then

(1) if $\mathbf{x}^{(i-1,n)}$ is also in the dual cone of BH1, $\mathbf{x}^{(i,1)}$ is the projection onto the kink of BH1
(2) if $x^{(i-1,n)}$ is *not* in the dual cone of BH1, then $x^{(i,1)}$ is the projection onto the nearest hyperplane segment, $x_1 = 0$ or $w_1 = 0$.

If $k = 2,\dots,n$

(i) Project $\mathbf{x}^{(i,k-1)}$ directly onto BHk:

If $\mathbf{x}^{(i,k-1)}$is in the interior of BHk, then $\mathbf{x}^{(i,k)}$ is the projection onto the nearest hyperplane segment, $x_k = 0$ or $w_k = 0$

If $\mathbf{x}^{(i,k-1)}$ is in the exterior of BHk, then

(1) $\mathbf{x}^{(i,k-1)}$ is also in the dual cone of BHk, then $\mathbf{x}^{(i,k)}$ is the projection onto the kink of BHk
(2) if $\mathbf{x}^{(i,k-1)}$ is *not* in the dual cone of BHk, then $\mathbf{x}^{(i,k)}$ is the projection onto the nearest hyperplane segment, $x_k = 0$ or $w_k = 0$. If they are equidistant, then project to $x_k = 0$.

(III) Repeat step (II) until $\mathbf{x}^{(i,k)} \to \mathbf{z}$ (the unique solution) or until a suitable stopping criteria has been met.

An alternative approach to finding the solution to LCP(A,$\mathbf{b}$) is to find all possible intersections of the hyperplanes formed by the equations $w_k = 0$ for k in some subset of $\{1,2,\dots,n\}$ and $x_j = 0$ for j not in this subset, and then testing the nonnegativity constraints on the remaining variables. Since A is a P-matrix, only one of these intersection points will satisfy all the required constraints. This is extremely inefficient in most cases.

The main difficulty with the direct algorithm is that for each iterate, it is necessary to test to find out in which segment of the exterior of the feasible region the current iterate is located. This involves finding intersections of hyperplanes and testing to see where positivity constraints are satisfied. This is nearly as inefficient as finding all the intersection points in the alternative approach previously mentioned.

In addition, the projections here are not necessarily orthogonal. This is true when we are in the interior of the dual cone(s) or the region(s) of positivity. When we project from the exterior of the dual cone to the bent hyperplane, we are projecting onto a convex set. However, when we project to from the interior of the dual cone to the bent hyperplane, we are not projecting onto a convex set. Hence, we cannot use the theory of POCS to demonstrate convergence. In fact, convergence has only been shown for the 2-dimensional case [12]. This result is stated as follows.

Theorem 3.1. *Let A be a 2×2 P-matrix. Given the 2×2 linear complementarity problem, if we project between the 2 bent lines as determined by the* LCP *using the two-step projections as defined in the Direct algorithm, then the iteration will converge to the unique complementarity point, $\mathbf{z}$, for any starting value, $\mathbf{x}_0$.*

In addition, when projecting from the interior of a dual cone, the projections may not be uniquely defined. The points along the hyperplane that bisects the angle between the two segments forming the bent hyperplane are equidistant to each segment forming the bent hyperplane. In this case, for the projection onto the kth hyperplane, our default is to project to the segment formed by the kth axis, $x_k = 0$.

This method seems to converge for all starting values when A is a P-matrix. In fact, in practice, the distance from the iterates along each bent hyperplane to the solution always decreases. The primary use of Algorithm I has been in using it to prove convergence (see Haddad [12] for proof) of Algorithm I for $n = 2$, which is presented in the next section.

4. Algorithm II. The direct algorithm of Section 3 requires too much a priori knowledge of the LCP, hence another approach was deemed necessary. We present a method based on alternating projections, where the only information necessary is the information provided in the statement of the LCP$(A,\mathbf{b})$, namely A and $\mathbf{b}$.

Two-step algorithm: (I) Given A is an $n\times n$ matrix, $\mathbf{b}\in\mathbb{R}^n$. Choose a starting vector $\mathbf{x}^{(o,n)}$.

(II) Construct the alternating sequences, $\mathbf{x}^{(i,1)}$, for $i = 1,2,3,\ldots$ in the following manner:

For $k = 1,2,\ldots,n$

If $k = 1$:

(i) $\hat{\mathbf{x}}^{(i,1)}$ is the projection of $x^{(i-1,n)}$ directly onto $x_1 \geq 0$.

(ii) $\tilde{\mathbf{x}}^{(i,1)}$ is the projection of $\hat{\mathbf{x}}^{(i,1)}$ directly onto $w_1 \geq 0$.

(iii) $\mathbf{x}^{(i,1)}$ is the projection of $\tilde{x}^{(i,1)}$ directly onto the closest of $x_1 = 0$ or $w_1 = 0$.

If they are equidistant, then project to $x_1 = 0$.

If $k = 2,\ldots,n$:

(i) $\hat{\mathbf{x}}^{(i,k)}$ is the projection of $\mathbf{x}^{(i,k-1)}$ directly onto $x_k \geq 0$.

(ii) $\tilde{\mathbf{x}}^{(i,k)}$ is the projection of $\hat{\mathbf{x}}^{(i,k)}$ directly onto $w_k \geq 0$.

(iii) $\mathbf{x}^{(i,k)}$ is the projection of $\tilde{\mathbf{x}}^{(i,k)}$ onto the closest of $x_k = 0$ or $w_k = 0$.

If they are equidistant, then project to $x_k = 0$.

(III) Repeat step II until $\mathbf{x}^{(i,k)} \to \mathbf{z}$ (the unique solution) or until a suitable stopping criteria has been met.

For each bent hyperplane k, this algorithm projects, first onto the region where the axis, x_k, is nonnegative, then onto the region where the kth hyperplane, w_k, is nonnegative, and then onto the nearest plane, $x_k = 0$ or $w_k = 0$. Notice that the last projection is not orthogonal and that there will be at most *two* projections made for each hyperplane at each iteration (hence the name).

The main advantage this has over the other method is that it only requires as input, A, b, and possibly a solution for comparison. Another benefit is that it is easily programmed.

Even though the projection labeled (iii) is not an orthogonal projection, this does not seem to pose a problem in terms of convergence, especially when A is a P-matrix. Whenever A is a P-matrix, the distance from the iterates along each bent hyperplane to the solution always seems to be decreasing for all n. This seems to be true regardless of the starting vector, as is also the case with the direct algorithm. This result is summarized as follows for $n = 2$.

THEOREM 4.1. *Let A be a 2×2 P-matrix. Given the 2×2 linear complementarity problem, if we project between the 2 bent lines as determined by the* LCP *using direct projections as defined in the two-step algorithm, then the iteration will converge to the unique complementarity point,* $\mathbf{z}$*, for any starting value,* $\mathbf{x}_0$.

Currently we have a proof for this theorem (see Haddad [12]). We first prove that the direct algorithm converged for all starting values using the geometry of the problem and a fixed-point argument. The proof for the two-step algorithm then uses *Theorem 3.1* and geometry to prove convergence for all starting values, provided that A is a P-matrix.

5. Extensions to GLCP: Algorithm III. The GLCP is a natural extension of the LCP. The objective is still to find the intersection of a number of bent hyperplanes; the only difference is that there may be multiples bends in each hyperplane. One may view solving the LCP or GLCP as solving a system of piecewise linear equations. Thus it is possible to think of LCPs and GLCPs as natural extensions to solving linear systems. This may be further generalized, since the nonlinear complementarity problem (NCP) is yet another extension of both the generalized complementarity problems and systems of nonlinear equations. We will not deal with NCPs in this paper and we will not provide computational results for any GLCPs (some are presented in [12]).

An advantageous feature of Algorithms I and II is that they easily extend to solving GLCPs. The key to guaranteed success is that the representative matrices must all be P-matrices. However, the direct algorithm for the GLCP is even more cumbersome to program in higher dimensions than its LCP counterpart, hence we will omit presenting it here. It is primarily the same approach as the direct algorithm: alternately project directly onto each region of positivity as defined by the GLCP$(A, \mathbf{b})$. We do prove the following convergence result in [12], which is used to prove Theorem 5.2.

THEOREM 5.1. *Suppose that all of the representative matrices for the* GLCP$(N, \mathbf{b})$ *defined above for $n = 2$ are P-matrices. Suppose we alternatively project onto the bent lines, BH1 and BH2, formed by the* GLCP$(N, \mathbf{b})$*. Then the iteration will converge to the unique complementarity point,* $\mathbf{z}$ *for any starting value,* $\mathbf{x}_0$.

We now present the multi-step algorithm for the GLCP, which is an extension of the two-step algorithm for the LCP. Here, the number of steps depends upon the number of segments in each bent hyperplane.

Consider GLCP$(N,\mathbf{b})$ where N is a vertical block matrix of dimension $m\times n$, $m\geq n$, of *type* $(m_1,\dots,m_n)$ and whose representative matrices are all P-matrices.

Multi-step algorithm. (I) Given N is an $m\times n$ matrix, $b\in\mathbb{R}^n$. Choose a starting vector $\mathbf{x}^{(o,n,m_n)}$.

(II) Construct the alternating sequences, $\mathbf{x}^{(i,n,m_n)}$, for $i=1,2,3,\dots$ in the following manner:

For $k=1,2,\dots,n$

If $k=1$:

(i) $\tilde{\mathbf{x}}^{(i,1,o)}$ is the projection of $\mathbf{x}^{(i-1,n,m_n)}$ directly onto $x_1\geq 0$.

(ii) For $l=1,\dots,m_1$:

$\tilde{\mathbf{x}}^{(i,1,l)}$ is the projection of $\tilde{\mathbf{x}}^{(i,1,l-1)}$ directly onto $w_1^l\geq 0$.

(iii) $\mathbf{x}^{(i,1,m_1)}$ is the projection of $\tilde{\mathbf{x}}^{(i,1,m_1)}$ directly onto the closest segment of BH1.

(If more than one segment is closest, we default to projecting onto the first of these hyperplane segments encountered.)

If $k=2,\dots,n$:

(i) $\tilde{\mathbf{x}}^{(i,k,o)}$ the projection of $\mathbf{x}^{(i,k-1,m_{k-1})}$ directly onto $x_k\geq 0$.

(ii) For $l=1,\dots,m_k$:

$\tilde{\mathbf{x}}^{(i,k,l)}$ is the projection of $\tilde{\mathbf{x}}^{(i,k,l-1)}$ directly onto $w_k^l\geq 0$.

(iii) $\mathbf{x}^{(i,k,m_k)}$ is the projection of $\tilde{\mathbf{x}}^{(i,k,m_k)}$ directly onto the closest segment of BHk.

(If more than one segment is closest, we default to projecting onto the first of these hyperplane segments encountered.)

(III) Repeat step (II) until $\mathbf{x}^{(i,n,m_n)}\to\mathbf{z}$ (the unique solution) or until a suitable stopping criteria has been met.

This algorithm is unusual in that, for a time, the iterates *may not actually be points on the bent hyperplane*. What is happening here is that the projection may be onto a *ghost segment*. If you consider the GLCP in another way, the region of positivity of, say, the kth bent hyperplane is the intersection of what would be the interiors of the singly bent hyperplanes $x_k=0,\ w_k^1=0;\dots;x_k=0,\ w_k^{m_k}=0$. A segment of any of these singly bent hyperplanes that is not part of the bent hyperplane of the resulting GLCP is referred to as a *ghost segment*. In spite of this, the distances from the kth iterate to the solution associated with projecting onto the kth bent hyperplane is always decreasing. We prove the following theorem in Haddad [12].

Theorem 5.2 (see [12])**.** *Suppose that all of the representative matrices for the* GLCP$(N,\mathbf{b})$ *defined above for* $n=2$ *are P-matrices. Suppose we alternatively project onto the bent lines, BH1 and BH2, formed by* GLCP$(N,\mathbf{b})$ *using the multi-step algorithm defined above. Then the iteration will converge to the unique complementarity point,* $\mathbf{z}$ *for any starting value,* $\mathbf{x}_o$.

6. Improving convergence rates. Unfortunately, as with POCS, convergence of these algorithms may be extremely slow in some cases. This occurs primarily when the angle between (the normals of) segments of adjacent hyperplanes is small. If this

is the case for a linear system, we would say that the matrix involved is *ill-conditioned*. In the case of the LCP$(A,\mathbf{b})$, this results from the matrix A, or possibly any other matrix where the columns of A are interchanged with the columns of the $n\times n$ identity matrix, I, being ill-conditioned. For the GLCP$(N,\mathbf{b})$, this may happen if any of the representative matrices of N, or the matrices formed by interchanging columns of the representative matrices with columns of I are ill-conditioned.

One way to (dramatically) increase convergence rates is with relaxation techniques. The projection operator in each algorithm becomes $x^{j+1} = T_m T_{m-1}\cdots T_1 x^j$, where $T_i = I + \lambda_i(\mathrm{Pr}_i - I)$, I is the identity operator, Pr_i is as defined in (2.8), and λ_i is a relaxation parameter between 0 and 2. Typically if we project onto some axis, $x_k = 0$, then we take λ_k to be 1, as this leads to the best results. Different λ_i could be used for each hyperplane, but in practice we always use the same λ for each.

In the case of LCP$(A,\mathbf{b})$, it is sometimes possible to increase the rate of convergence by taking advantage of orthogonality. This may be achieved by changing the order of the projections onto each hyperplane. It may also make more sense to project to the hyperplanes first, rather than the axes. This can significantly improve convergence in some orthogonal P-matrices.

7. Examples and numerical results. There are many applied areas in which complementarity has been successfully used. The examples mentioned in the introduction are but a few. For example, we used generalized complementarity to model and solve a Lotka-Volterra system both in two dimensions [9], and again in n dimensions with diffusion among patches [10]. For a more extensive listing of the different systems that we used to test our algorithm, see Haddad [12].

7.1. Computational details. We converted all of our FORTRAN programs into MATLAB 5.1 M-files on a Power PC 9500 (120 MH, 32 K RAM). We present results for $n \le 500$ only because of the restriction of the Power PC. While using FORTRAN, we did tests on much larger systems (see [12]).

Several types of error criteria were used to measure accuracy of the computed solution. An *iteration* onto a (bent) hyperplane refers to projecting from one hyperplane to the next. A *cycle* refers to projecting to every (bent) hyperplane once. The most useful error criterion was the relative error using the Euclidean norm when we happened to know the solution a priori. This was specified to be less than some prescribed error tolerance, usually 10^{-6}. If we did not know the solution, the distance between two consecutive iterates on a given hyperplane (usually the nth) divided by the most recent iterate (after one cycle of projections) was used as an indication of *relative error*. If the sum of the distances between consecutive hyperplanes of one cycle is compared to the sum for the next cycle and the difference was less than some tolerance, then this was used as another stopping criteria. However, if cycling (the projecting back and forth between the same sequence of points in two consecutive cycles) is going to occur (which may happen if A is not a P-matrix), then comparing two consecutive sums at the end of two cycles could be used as an indication thereof. In the absence of knowing the actual solution, the complementarity conditions may also be checked using the "solution" generated by the algorithm.

The solution, if it exists, always occurs in the nonnegative 2^n-tant. In general, the starting vector that was specified was either $-99\,\mathbf{e} = [-99,\ldots,-99]^T$ or $\mathbf{0} = [0,\ldots,0]^T$, whichever yields convergence soonest.

7.2. Cryer's examples for PSOR. The iterative method that we compared ours to was Projective Successive Over-Relaxation (PSOR) for solving linear complementarity as used by Cryer in [3, 4]. It is well known, comparable to our in the terms of the number of operations used per iterate, and, like ours, it is robust and easily programmed. For problems where PSOR was initially used, that is, where A is symmetric and positive definite, PSOR tends to work, as well as, or even better than our method. However, the main difference is that our method works when the matrix is a P-matrix (and not necessarily symmetric or positive definite), and in practice works quite well on many that are not P-matrices, regardless of the starting vector. In this section, we present comparisons that will demonstrate this. Note that in each case the matrix A, the right-hand side (RHS) $\mathbf{b}$, and the solution $\mathbf{x}$, are given. In each case, we are solving LCP$(A,\mathbf{b})$.

Note in the following example, that A is not symmetric, but the symmetric part is positive definite.

Example 7.1. $A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -1 & 0 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix}$, a solution of $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ with a starting vector of $\mathbf{0}$.

Note that the rows of A are orthogonal. Our method converges to the solution in 8 cycles with no relaxation, while PSOR does not converge, though it will converge in 13 cycles when we under-relax with $\lambda = 0.65$. We also looked at larger matrices whose structure was the same and we obtained similar results.

Of course, there were matrices where PSOR converges much more quickly than ours, but in each case we find that ours will converge, albeit sometimes more slowly. Even if the matrices involved are not P-matrices, our method never diverges to infinity (in magnitude), whereas PSOR may. This is what occurs in the following example. Our two-step converges in 46 cycles (16 cycles if we over-relax $\lambda = 1.4$) while PSOR seems to diverge to infinity for all values of λ, even values as small as 0.01.

Example 7.2. $A = \begin{bmatrix} 1 & -4 \\ -1 & 1 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} -3 \\ 0 \end{bmatrix}$, and a solution of $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ with a starting vector of $10\,\mathbf{e}$.

In some cases, PSOR may cycle if the matrix is not symmetric, but is positive definite. For instance, in the following example, our two-step converges in 5 cycles, whereas PSOR bounces between several vectors, none of which is a solution. However, if under-relaxation is used, then PSOR converges in as little as 10 cycles.

Example 7.3. $A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, and a solution of $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ with almost any starting vector that we tried.

The two-step method will work on systems upon which PSOR will not work at all. Our two-step is proven to always work in the 2-dimensional case (with A a P-matrix) and in practice has always worked on LCPs with P-matrices and GLCPs with representative matrices for $n > 2$. For instance, take the following matrix.

EXAMPLE 7.4. Consider the following $A = [a_{ij}]$, where A is cyclic,

$$a_{ij} = \begin{cases} 1 & \text{if } i = j, \\ c & \text{if } i = j+1, \\ c & \text{if } i = n,\ j = 1. \end{cases} \tag{7.1}$$

This example is interesting for many reasons. Note that when $c = 4$ and n is even, the matrix A is not a P-matrix (it has a negative determinant or subdeterminant) and may have more than one solution. Whereas when n is odd, the matrix is a P-matrix that is *not* diagonally dominant and whose symmetric part is *not* positive definite. We looked at the case where $n = 4, 5, 50, 51, 100, 101, 500$. In each case the RHS, $\mathbf{b} = 50\mathbf{e}$, the starting vector was chosen to be $\mathbf{x}_0 = \mathbf{0}$ and the solution entered in for comparison was $[50\ 0\ 50 \cdots 50\ 0]$ for n even (which was one of many solutions) and $10\mathbf{e}$ for n odd (the only solution). One interesting point is that, for n even, PSOR would only converge to the solution, $[50\ 0\ 50 \cdots 50\ 0]$, regardless of which starting vector was entered and the two-step would only converge to $10\mathbf{e}$, regardless of both $\mathbf{x}$ and n.

PSOR does not converge to the solution when n is odd, whether over- or under-relaxation is used, nor do we expect it to. By this we mean that after 50,000 cycles, the method had not reached the solution and was not getting closer to the (unique) known solution. In the case where n is even, since A is not a P-matrix, there are multiple solutions, and PSOR seems to bounce back and forth between several vectors (none of which is a solution to the LCP$(A, \mathbf{b})$) when no relaxation is used. In the case where n is odd, there is only one solution since A is a P-matrix, and PSOR still bounces between several vectors.

However, for the two-step, when $n = 4$, over- and under-relaxation seems to speed the convergence of the two-step. When n is even and larger, optimal convergence seems to occur when no relaxation is used in the two-step. When n is odd, no relaxation seems best. Results for both methods are provided in Table 7.1.

Note that if the solution is known ahead of time and is used for comparison, the number of steps to convergence decrease (typically) by 2 iterations.

A more practical example is the food chain matrix. These can appear in Lotka-Volterra systems (see [9, 10, 19, 22]). A typical food chain matrix is a tridiagonal matrix with negative entries on either the sub- or super-diagonal and at least one 0 on the diagonal and positive entries elsewhere on the diagonal. These matrices are P_0 matrices and the issue of solving them with the two-step approach is addressed in [12]. Instead, in Example 7.5, we looked at the following food chain matrix A (which happens to be a P-matrix whose symmetric part is positive definite) found in Zhien et al. [22].

EXAMPLE 7.5. Solve LCP$(A, \mathbf{b})$ when $A = [a_{ij}]$ and

$$a_{ij} = \begin{cases} 1 & \text{if } i = j-1, \\ d & \text{if } i = j, \\ -1 & \text{if } i = j+1. \end{cases} \tag{7.2}$$

We previously considered the case where $n = 4$ and $d = 1$ in Example 7.1. In that example we noted that PSOR does not converge. Generally we restricted $d \geq 1$. In

TABLE 7.1

n	Solution(s)	PSOR		Two-step	
Odd Even	$10\mathbf{e}$ $[50\ 0 \cdots 50\ 0]$, $10\mathbf{e}$, etc.	$\lambda = 1$	# cycles/optimal λ solution	$\lambda = 1$	# cycles/optimal λ solution
4	Many	DNC	32/0.45 [50 0 50 0]	12 $\mathbf{x} = 10\mathbf{e}$	10/1.05 $\mathbf{x} = 10\mathbf{e}$
5	$10\mathbf{e}$	DNC	DNC	10 $\mathbf{x} = 10\mathbf{e}$	10 $\mathbf{x} = 10\mathbf{e}$
50	Many	DNC	89/0.38 $[50\ 0 \cdots 50\ 0]$	13 $\mathbf{x} = 10\mathbf{e}$	13/1 $\mathbf{x} = 10\mathbf{e}$
51	$10\mathbf{e}$	DNC	DNC	11 $\mathbf{x} = 10\mathbf{e}$	11 $\mathbf{x} = 10\mathbf{e}$
100	Many	DNC	145/0.39 $[50\ 0 \cdots 50\ 0]$	13 $\mathbf{x} = 10\mathbf{e}$	13/1 $\mathbf{x} = 10\mathbf{e}$
101	$10\mathbf{e}$	DNC	DNC	11 $\mathbf{x} = 10\mathbf{e}$	11 $\mathbf{x} = 10\mathbf{e}$
500	Many	DNC	> 500/0.38 $[50\ 0 \cdots 50\ 0]$	14 $\mathbf{x} = 10\mathbf{e}$	14/1 $\mathbf{x} = 10\mathbf{e}$
501	$10\mathbf{e}$	DNC	DNC	11 $\mathbf{x} = 10\mathbf{e}$	11 $\mathbf{x} = 10\mathbf{e}$

Table 7.2, we summarize results for $d = 2$, when RHS is $\mathbf{b} = A\mathbf{x}$, where $\mathbf{x} = \mathbf{e}$ is the solution and the starting vector is $\mathbf{0}$. The optimal relaxation parameter in each case for PSOR seems to be $\lambda \approx 0.8$, while no relaxation seems to work best for the two-step.

TABLE 7.2

n (size of matrix) $d = 2$	PSOR w/ no relaxation	PSOR w/optimal relaxation ($\lambda = 0.8$)	Two-step w/ no (optimal) relaxation
4	27	9	5
10	116	12	7
50	> 1000	16	9
100	> 1000	17	9
500	> 1000	18	10

EXAMPLE 7.6. In this last example, we consider food chain matrices with a similar structure except that the off-diagonal elements are larger in magnitude than those on the diagonal. We present the results for the case where $A = [a_{ij}]$ and

$$a_{ij} = \begin{cases} -c & \text{if } i = j-1, \\ 1 & \text{if } i = j, \\ c & \text{if } i = j+1. \end{cases} \tag{7.3}$$

where $c = 4$, $\mathbf{x} = \mathbf{e}$, $\mathbf{b} = A\mathbf{x}$, and the starting vector is $\mathbf{0}$. Results are provided in Table 7.3.

TABLE 7.3

n (size of matrix) $c = 4$	PSOR w/ no relaxation	PSOR w/optimal relaxation	Two-step w/no relaxation	Two-step w/optimal relaxation
4	DNC	50, $\lambda = .21$	16	10, $\lambda = 1.25$
10	DNC	52, $\lambda = .21$	74	18, $\lambda = 1.45$
50	DNC	68, $\lambda = .21$	199	36, $\lambda = 1.65$
100	DNC	91, $\lambda = .21$	219	48, $\lambda = 1.62$
500	DNC	91, $\lambda = .21$	240	60, $\lambda = 1.6$

7.3. Murty's example for Murty's and Lemke's methods. Consider the $n \times n$ upper triangular matrix U with entries u_{ij} given by

$$u_{ij} = \begin{cases} 0 & \text{if } i > j, \\ 1 & \text{if } i = j, \\ 2 & \text{if } i < j. \end{cases} \tag{7.4}$$

This matrix is a P-matrix and has a unique solution for every $\mathbf{b} \in \mathbb{R}^n$. For $n \geq 2$, $2^n - 1$ pivots are required to solve LCP$(U, \mathbf{e})$ using Murty's algorithm [16], a classic pivoting method used to solve LCPs. When $n = 100$, we find that roughly 10^{31} pivots are required using Murty, whereas 1530 cycles are needed if the two-step is used. Solving LCP$(U^T, \mathbf{e})$ for $n \geq 2$ with another well-known pivoting scheme, Lemke's method [14] also requires $2^n - 1$, while the two-step converges in exactly one cycle with no relaxation! Of course, these examples are contrived and are meant only as pathologies.

8. Conclusion and possible future directions. Several algorithms have been presented. Firstly, they provide yet another means for which one might solve both linear and generalized linear complementarity problems. Secondly, the class of such problems, for which the matrices involved are P-matrices, also represents a class of problems where the solution is the intersection of a number of closed nonconvex sets and alternating projections onto each set leads to a solution. It is our belief that this is the first such classification. Since these are merely extensions of intersecting lines of linear systems (which happen to be convex), some properties of linear systems appear to apply to the intersections of piecewise linear systems. Unfortunately, one of these properties is that convergence can be extremely slow if opposing hyperplanes are "close" in the sense that the angle between their normals is small. This may be compounded in the case of bent hyperplanes and worsens as the number of segments in each hyperplane increase (as in the GLCP). In order for the representative matrices to all be P in the GLCP, many of the angles between bends of consecutive hyperplanes must be small. Fortunately, over- and under-relaxation appears to speed up convergence in many cases. One property of linear systems that is lost is convexity, and hence the nonexapansivity of the projection operators.

Currently we are looking into other forms of acceleration and possible preconditioning of the system. Attempts to pre-condition so far have not been successful due, in part, to the fact that orthogonality (of the axes, in particular) is lost for various

parts of the problem, while it may or may not be introduced to other parts. If one knew approximately where the solution lay, then one could possibly take advantage of this by, say, pre-multiplying by a matrix that enlarges the acute angles between the hyperplanes, thereby increasing the rate of convergence.

Acknowledgement. This work is a portion of the Ph.D. dissertation done by the first author while at Rensselaer Polytechnic Institute, while under the advisement of the second author.

References

[1] R. W. Cottle and G. B. Dantzig, *A generalization of the linear complementarity problem*, J. Combinatorial Theory **8** (1970), 79–90.

[2] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Computer Science and Scientific Computing, Academic Press, Massachusetts, 1992.

[3] C. W. Cryer, *The method of Christopherson for solving free boundary problems for infinite journal bearings by means of finite differences*, Math. Comp. **25** (1971), 435–443.

[4] ———, *The solution of a quadratic programming problem using systematic overrelaxation*, SIAM J. Control Optim. **9** (1971), 385–392.

[5] S. P. Dirkse and M. C. Ferris, *MCPLIB: a collection of nonlinear mixed complementarity problems*, Optim. Methods Softw. **5** (1995), 319–345.

[6] A. A. Ebiefung and M. M. Kostreva, *The generalized Leontief input-output model and its application to the choice of new technology*, Ann. Oper. Res. **44** (1993), no. 1-4, 161–172.

[7] M. C. Ferris and J. S. Pang, *Engineering and economic applications of complementarity problems*, SIAM Rev. **39** (1997), no. 4, 669–713.

[8] L. G. Gubin, B. T. Polyak, and E. V. Raik, *The method of projections for finding the common point of convex sets*, U.S.S.R. Comput. Math. and Math. Phys. **7** (1967), no. 6, 1–24.

[9] G. J. Habetler and C. N. Haddad, *Global stability of a two-species piecewise linear Volterra ecosystem*, Appl. Math. Lett. **5** (1992), no. 6, 25–28.

[10] ———, *Stability of piecewise linear generalized Volterra discrete diffusion systems with patches*, Appl. Math. Lett. **12** (1999), no. 2, 95–99.

[11] G. J. Habetler and B. P. Szanc, *Existence and uniqueness of solutions for the generalized linear complementarity problem*, J. Optim. Theory Appl. **84** (1995), no. 1, 103–116.

[12] C. N. Haddad, *Algorithms involving alternating projections onto the nonconvex Sets occurring in linear complementarity problems*, Ph.D. thesis, Rensselaer Polytechnic Institute, 1993.

[13] M. M. Kostreva, *Cycling in linear complementarity problems*, Math. Programming **16** (1979), no. 1, 127–130.

[14] C. E. Lemke, *Some pivot schemes for the linear complementarity problem*, Math. Programming Stud. (1978), no. 7, 15–35.

[15] L. Mathiesen, *An algorithm based on a sequence of linear complementarity problems applied to a Walrasian equilibrium model: an example*, Math. Programming **37** (1987), no. 1, 1–18.

[16] K. G. Murty, *Computational complexity of complementary pivot methods*, Math. Programming Stud. (1978), no. 7, 61–73.

[17] K. P. Oh and P. K. Goenka, *The elastohydrodynamic solution of journal bearings under dynamic loading*, Journal of Tribology **107** (1985), 389–395.

[18] B. P. Szanc, *The generalized complementarity problem*, Ph.D. thesis, Rensselaer Polytechnic Institute, New York, 1989.

[19] Y. Takeuchi, *Global stability in generalized Lotka-Volterra diffusion systems*, J. Math. Anal. Appl. **116** (1986), no. 1, 209–221.

[20] R. L. Tobin, *A variable dimension solution approach for the general spatial price equilibrium problem*, Math. Programming **40** (1988), no. 1, (Ser. A), 33–51.

[21] L. Vandenberghe, B. L. De Moor, and J. Vandewalle, *The generalized linear complementarity problem applied to the complete analysis of resistive piecewise-linear circuits*, IEEE Trans. Circuits and Systems **36** (1989), no. 11, 1382–1391.

[22] M. Zhien, Z. Wengang, and L. Zhixue, *The thresholds of survival for an n-dimensional food chain model in a polluted environment*, J. Math. Anal. Appl. **210** (1997), no. 2, 440–458.

Caroline N. Haddad: Department of Mathematics, State University of New York at Geneseo, Geneseo, NY 14454, USA

E-mail address: haddad@geneseo.edu

George J. Habetler: Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

E-mail address: habetg@rpi.edu