Journal of Computational Mathematics, Vol.14, No.1, 1996, 54–63.

A MODIFIED PROJECTION AND CONTRACTION METHOD FOR A CLASS OF LINEAR COMPLEMENTARITY PROBLEMS*1)

B.S. He
(*Department of Mathematics, Nanjing University, Nanjing, China*)

Abstract

Recently, we have proposed an iterative projection and contraction (PC) method for a class of linear complementarity problems (LCP)[4]. The method was showed to be globally convergent, but no statement could be made about the rate of convergence. In this paper, we develop a modified globally linearly convergent PC method for linear complementarity problems. Both the method and the convergence proofs are very simple. The method can also be used to solve some linear variational inequalities. Several computational experiments are presented to indicate that the method is surprising good for solving some known difficult problems.

1. Introduction

Let $L = \{1, \ldots, l\}$, $I \subset L$, M be an $l \times l$ positive semi-definite matrix (but not necessarily symmetric) and $q \in R^l$. For generalized linear complementarity problems

$$\text{(GLCP)} \qquad \begin{cases} u_i \geq 0, \quad (Mu+q)_i \geq 0 \quad u_i(Mu+q)_i = 0, & \text{for} \quad i \in I \\ (Mu+q)_i = 0, \qquad \text{for} \quad i \in L \setminus I, & \end{cases} \tag{1}$$

we have presented a globally convergent projection and contraction method (PC method)[4]. This method is an iterative procedure which requires in each step only two matrix-vector multiplications, and performs no transformation of the matrix elements. The method therefore allows the optimal exploitation of the sparsity of the constraint matrix and may thus be efficient for large sparse problems[4]. However, only for some special GLCP's (GLCP's arising from linear programming with standard form[5,6] and from some least distance problems[8]), the improved PC methods with linear convergence are established.

In this paper, we modify the original PC algorithm in [4]. Using a new step-size rule, without the estimation of the norm of M, we are able to obtain global linear

* Received on June 8, 1994.

1)The project supported by National Natural Science Foundation of China and the Natural Science Foundation of Jiangsu Province.

convergence for problem (1) in general form. Moreover, the convergence proof in this paper is much simpler than the one in [5] and [6].

Our paper is organized as follows. In Section 2, we quote some theoretical background from [4]. Section 3 describes the new algorithm and its relation to the original one. Section 4 proves the convergence properties of our new algorithm. In Section 5, we present some numerical results. Finally, in Section 6, we conclude the paper with some remarks.

We use the following notations. The i-th component of a vector u in the real l-dimensional Euclidean space R^l is denoted by u_i. A superscript such as in u^k refers to specific vectors and k usually denotes an iteration index. $P_\Omega(\cdot)$ denotes the orthogonal projection on the convex closed set Ω. $\|\cdot\|$ and $\|\cdot\|_\infty$ are the Euclidean and the max-norm, respectively. For a positive definite matrix G, the norm $\|u\|_G$ is given by $(u^TGu)^{\frac{1}{2}}$.

2. Theoretical Background

Let

$$\Omega = \{u \mid u_i \geq 0, \text{ for } i \in I\}, \tag{2}$$

$$\Omega^* := \{u \mid u \text{ is a solution of (GLCP)}\}. \tag{3}$$

Throughout the paper we assume that $\Omega^* \neq \emptyset$. The projection $v = P_\Omega(u)$ of u onto Ω is simply given by

$$v_i = \begin{cases} \max\{0, u_i\} & \text{if } i \in I, \\ u_i & \text{if } i \in L \setminus I. \end{cases}$$

It is easy to see that GLCP's can be rewritten in an equivalent way as

$$\text{(PE)} \qquad u = P_\Omega[u - (Mu + q)]. \tag{4}$$

We call it a projection equation. Based on (4) we denote

$$e(u) := u - P_\Omega[u - (Mu + q)], \tag{5}$$

and

$$\varphi(u) := e(u)^T(Mu + q). \tag{6}$$

We have the following basic lemma:

Lemma 1. *Let $u \in \Omega$, then*

$$\varphi(u) \geq \|e(u)\|^2. \tag{7}$$

A simple proof of Lemma 1 can be found in [4]. From this result we obtain immediately the following

Theorem 1. $u \in \Omega$ *and* $\varphi(u) = 0 \iff e(u) = 0 \iff u \in \Omega^*$.

For $u \in \Omega$, the function $\|e(u)\|$ and $\varphi(u)$ are continuous and can be viewed as measures for the distance of u from the solution set Ω^*. In our algorithm below we take the vector

$$g(u) := M^T e(u) + (Mu + q) \tag{8}$$

as search direction.

Theorem 2. *Let $u \in \Omega$, $u^* \in \Omega^*$ and $g(u)$ be defined as above. Then we have*

$$(u - u^*)^T g(u) \geq \varphi(u) + (u - u^*)^T M(u - u^*) \geq \varphi(u). \tag{9}$$

The proof of Theorem 2 can also be found in [4]. For $u \in \Omega$, the direction $-g(u)$ is a profitable direction, i.e., a descent direction of $\|u - u^*\|^2$. As a consequence of Theorem 2, one can build a first PC algorithm[1,4], which is already globally convergent whenever $\Omega^* \neq \emptyset$.

3. The New Algorithm and its Relation to the Original One

First, we state our new algorithm.

PC Algorithm (new)

Step 0. Assume Δ_1 and Δ_2 are fixed constants satisfying $0 < \Delta_1 \leq \Delta_2 < 2$. Let $\varepsilon > 0$ and $u^0 \in \Omega$. Set $k := 0$.

Step 1. Calculate $e(u^k)$. If $\dfrac{\|e(u^k)\|_\infty}{\|q\|_\infty} \leq \varepsilon$, stop.

Step 2. Calculate $g(u^k)$, set

$$\rho_k = \frac{\|e(u^k)\|^2}{\|(I + M^T)e(u^k)\|^2}, \tag{10}$$

and choose any

$$\gamma_k \in [\Delta_1, \Delta_2]. \tag{11}$$

Step 3. Set

$$\bar{u}^k = u^k - \gamma_k \rho_k g(u^k), \tag{12}$$

$$u^{k+1} = P_\Omega[\bar{u}^k]. \tag{13}$$

Set $k := k + 1$ and go to Step 1.

The main difference of our new algorithm with the original PC algorithm in [4] is that they take different step-size rules (choice of ρ). For convenience, we fix $\gamma_k \equiv 1$. In the original algorithm [3], for $u \in \Omega$, one puts

$$N(u) := \{i \in I \mid u_i = 0 \text{ and } g_i \geq 0\},$$
$$B(u) := \{1, \ldots, l\} \setminus N(u),$$

and denotes correspondingly,

$$u = \begin{pmatrix} u_B \\ u_N \end{pmatrix} \qquad g(u) = \begin{pmatrix} g_{u_B} \\ g_{u_N} \end{pmatrix}, \qquad g_B(u) = \begin{pmatrix} g_{u_B} \\ 0 \end{pmatrix}, \qquad g_N(u) = \begin{pmatrix} 0 \\ g_{u_N} \end{pmatrix}.$$

It is easy to see that for $u \in \Omega$ and $u^* \in \Omega^*$

$$(u - u^*)^T g_B(u) \geq (u - u^*)^T g(u) \geq \varphi(u). \tag{14}$$

The choice of the step-size in the original algorithm [4] is

$$\rho_{\text{prime}}(u^k) = \frac{\varphi(u^k)}{\|g_B(u^k)\|^2}. \tag{15}$$

This choice guarantees that the generated sequence $\{u^k\}$ satisfies

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \rho_{\text{prime}}(u^k)\|e(u^k)\|^2 \tag{16}$$

and converges globally to the solution set Ω^* (see [4] for a simple proof). However, ρ_{prime} may be very small and an iteration may get a petty profit.

The step-size $\rho_{\text{new}}(u^k)$ in our new algorithm is chosen by (10) and therefore

$$\rho_{\text{new}} \geq \frac{1}{\|I + M^T\|^2} := \text{const} > 0 \tag{17}$$

is bounded below. In the next section we will prove that the sequence $\{u^k\}$ generated by this new algorithm (in the case $\gamma_k \equiv 1$) satisfies

$$\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \rho_{\text{new}}(u^k)\|e(u^k)\|^2. \tag{18}$$

Therefore theoretically the new choice of the step length guarantees to get an "enough" great profit in each iteration. In practical computation, we suggest to choose

$$\rho = \max\{\rho_{\text{prime}}, \rho_{\text{new}}\} \tag{19}$$

and call the method a modified PC method. So that ρ_{new} serves as a safeguard against a "too short" step-length in the original method and the sequence $\{u^k\}$ generated by the modified algorithm still satisfies (18). We note that the projection on the general orthant Ω is trivial and for both PC algorithms the main work in each iteration is the computation of Mu and $M^T e(u)$.

Remark. In [6], for linear complementarity problems arising from standard linear programming, the step size ρ_{new} is defined as

$$\rho_{\text{new}} = \frac{\|e(u)\|^2}{\|e(u)\|^2 + \|M^T e(u)\|^2}.$$

Because in this case the matrix M is skew symmetric, the ρ_{new} in [6] can be rewritten as

$$\rho_{\text{new}} = \frac{\|e(u)\|^2}{\|e(u) + M^T e(u)\|^2}.$$

Therefore, for linear programming, the method in this paper is the same as in [6].

4. Convergence Results

We begin this section by stating the main result.

Theorem 3. *Let $u^* \in \Omega^*$. Then the sequence $\{u^k\}$ generated by the new PC Algorithm for linear complementarity problems of the form* (1) *satisfies*

$$\|u^{k+1}-u^*\|^2 \le \|u^k-u^*\|^2 - \gamma_k(2-\gamma_k)\rho_k\|e(u^k)\|^2. \tag{20}$$

Proof. We let

$$\Theta_k = \|u^k-u^*\|^2 - \|u^{k+1}-u^*\|^2 - \gamma_k(2-\gamma_k)\rho_k\|e(u^k)\|^2 \tag{21}$$

and show the equivalent assertion $\Theta_k \ge 0$. Since $u^{k+1} = P_\Omega(\bar{u}^k)$ and $u^* \in \Omega$, using the well-known property of a projection on a convex set

$$(\bar{u}^k - u^{k+1})^T(u^* - u^{k+1}) \le 0,$$

it follows that

$$\|u^{k+1}-u^*\|^2 \le \|\bar{u}^k-u^*\|^2 - \|\bar{u}^k-u^{k+1}\|^2. \tag{22}$$

Then from (21) and (22)

$$\begin{aligned}\Theta_k &\ge \|u^k-u^*\|^2 - \|\bar{u}^k-u^*\|^2 + \|\bar{u}^k-u^{k+1}\|^2 - \gamma_k(2-\gamma_k)\rho_k\|e(u^k)\|^2\\ &= \|u^k-u^*\|^2 - \|u^k-u^*-\gamma_k\rho_k g(u^k)\|^2 + \|u^k-u^{k+1}-\gamma_k\rho_k g(u^k)\|^2\\ &\quad - \gamma_k(2-\gamma_k)\rho_k\|e(u^k)\|^2 \qquad \text{(use (13))}\\ &= \|u^k-u^{k+1}\|^2 + 2\gamma_k\rho_k(u^k-u^*)^T g(u^k) + 2\gamma_k\rho_k(u^{k+1}-u^k)^T g(u^k)\\ &\quad - \gamma_k(2-\gamma_k)\rho_k\|e(u^k)\|^2.\end{aligned} \tag{23}$$

Note that from (9) and (6)

$$\begin{aligned}&(u^k-u^*)^T g(u^k) + (u^{k+1}-u^k)^T g(u^k)\\ &\quad\ge e(u^k)^T(Mu^k+q) + (u^{k+1}-u^k)^T[M^T e(u^k) + (Mu^k+q)]\\ &\quad= \{u^{k+1} - P_\Omega[u^k-(Mu^k+q)]\}^T(Mu^k+q) + (u^{k+1}-u^k)^T M^T e(u^k).\end{aligned} \tag{24}$$

Again since $u^{k+1} \in \Omega$, using the well-known projection property

$$\{[u^k-(Mu^k+q)] - P_\Omega[u^k-(Mu^k+q)]\}^T\{u^{k+1} - P_\Omega[u^k-(Mu^k+q)]\} \le 0,$$

it follows that

$$\begin{aligned}&\{u^{k+1} - P_\Omega[u^k-(Mu^k+q)]\}^T(Mu^k+q)\\ &\qquad\ge \{u^k - P_\Omega[u^k-(Mu^k+q)]\}^T\{u^{k+1} - P_\Omega[u^k-(Mu^k+q)]\}\\ &\qquad= \|e(u^k)\|^2 + (u^{k+1}-u^k)^T e(u^k).\end{aligned} \tag{25}$$

We use (24) and (25) in the basic inequality (23)

$$\begin{aligned}\Theta_k &\geq \|u^k - u^{k+1}\|^2 + 2\gamma_k\rho_k\{\|e(u^k)\|^2 + (u^{k+1} - u^k)^T[(I + M^T)e(u^k)]\}\\ &\quad - \gamma_k(2 - \gamma_k)\rho_k\|e(u^k)\|^2\\ &\geq \|u^k - u^{k+1}\|^2 + 2\gamma_k\rho_k(u^{k+1} - u^k)^T[(I + M^T)e(u^k)] + \gamma_k^2\rho_k\|e(u^k)\|^2\\ &\geq \|u^k - u^{k+1}\|^2 - \{\|u^{k+1} - u^k\|^2 + \gamma_k^2\rho_k^2\|(I + M^T)e(u^k)\|^2\} + \gamma_k^2\rho_k\|e(u^k)\|^2\\ &= \gamma_k^2\rho_k\{\|e(u^k)\|^2 - \rho_k\|(I + M^T)e(u^k)\|^2\}\\ &= 0. \qquad\qquad \text{(use (10))}\end{aligned}$$

The convergence theorem is the same as in [6]. However, here we have proved that the result is true for general positive semidefinite matrix M and general closed convex set Ω. The above proof is different and simpler in contrast with the one in [6].

In fact, because Theorem 3 is true for any $u^* \in \Omega^*$ and $\rho_k \geq \dfrac{1}{\|I + M^T\|^2}$, we have proved

$$\operatorname{dist}^2(u^{k+1}, \Omega^*) \leq \operatorname{dist}^2(u^k, \Omega^*) - \frac{\Delta_1(2 - \Delta_2)}{\|I + M^T\|^2}\|e(u^k)\|^2. \tag{26}$$

where

$$\operatorname{dist}(u, \Omega^*) = \inf\{\|u - u^*\| \mid u^* \in \Omega^*\}.$$

Especially, if $\Delta_1 = \Delta_2 = 1$, we have

$$\operatorname{dist}^2(u^{k+1}, \Omega^*) \leq \operatorname{dist}^2(u^k, \Omega^*) - \frac{1}{\|I + M^T\|^2}\|e(u^k)\|^2. \tag{27}$$

The function $\|e(u)\|$ measures how much u fails to be in Ω^*. (27) states that, we get a 'big' profit from an iteration, if $\|e(u^k)\|$ is not too small; conversely, if we get a very small profit from an iteration, then $\|e(u^k)\|$ is already very small and u^k is a 'sufficiently good' approximation of a $u^* \in \Omega^*$. As in [7], one can prove the global linear convergence from (27).

5. Numerical Experiments

First, the modified PC method was applied to two linear complementarity problems for which Lemke's algorithm is known to run in exponential time (see Chap. 6 in [9]). The first problem consist of an n-vector q which has each component equal -1 and the following matrix M:

$$M = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2\\ 0 & 1 & 2 & \cdots & 2\\ 0 & 0 & 1 & \cdots & 2\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Because $\frac{1}{2}(M+M^T)=ee^T$, M is positive semidefinite. Another 'exponential time' problem is defined by the same q and a matrix M given by:

$$M=\begin{pmatrix}1 & 2 & 2 & \cdots & 2\\ 2 & 5 & 6 & \cdots & 6\\ 2 & 6 & 9 & \cdots & 10\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 2 & 6 & 10 & \cdots & 4(n-1)+1\end{pmatrix}$$

It is possible to prove that the above matrix is positive definite. Harker and Pang[3] solved the both problems with the damped-Newton algorithm up to $n=128$. Table 1 reports their results.

Table 1. Number of Iteration with Damped–Newton Method

n=	8	16	32	64	128
Example 1	9	20	72	208	>300
Example 2	8	16	32	65	63

Both problems were solved by the modified PC iterative scheme

$$u^{k+1}=P_\Omega[u^k-\gamma\rho_k g(u^k)]$$

with $\gamma=1.8$. All the codes were written in FORTRAN. The calculations have been performed in single precision on a 486 Personal Computer. The stopping test was $\|e(u)\|_\infty \le 10^{-6}$. Tables 2–4 report the iteration number for different start points.

Table 2. Number of Iteration with PC Method

$n=$	8	16	32	64	128	256	512	1024	2048
Example 1	10	11	10	12	11	12	14	12	13
Example 2	25	28	44	56	54	93	65	134	69

The start vector u^0 has each component equal to 0.

Table 3. Number of Iteration with PC Method

$n=$	8	16	32	64	128	256	512	1024	2048
Example 1	9	12	12	12	12	13	13	16	16
Example 2	14	15	18	25	21	24	25	41	41

The start vector u^0 has each component equal to 1.

Table 4. Number of Iteration with PC Method

$n =$	8	16	32	64	128	256	512	1024	2048
Example 1	11	10	13	13	14	13	13	18	15
Example 2	22	32	34	30	40	82	117	48	42

The start vector u^0 was randomly generated with uniformly distributed entries in the interval (0,1)

Our test results show, that the modified PC-method for the first special problem up to $n = 2048$ requires only no more than 20 iterations and for the second no more than 140. For large problems the iteration number by PC method is fewer than by the damped-Newton method. Moreover, the damped-Newton method has to solve a Newton equation at each iteration and therefore needs $O(n^3)$ operations, while our method needs only $O(n^2)$ operations. In other words, when $n = 2048$, even if the damped-Newton method for example 1 requires only 300 iterations, in view of $\frac{300 \cdot O(2048^3)}{16 \cdot O(2048^2)}$, we believe that our method converges at least 2000 times faster than the damped-Newton method. For the same reason, in example 2, our method is at least 200 times faster than the damped-Newton method.

The second set of test examples consists of GLCP's resulting from linear programming. We compare the efficiency of the new method with the original one. As test problem we consider the transportation problem:

$$\begin{aligned} \min\ & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t}\ & \sum_{j=1}^{n} x_{ij} = s_i, \quad i = 1,\cdots,m, \\ & \sum_{i=1}^{m} x_{ij} = d_j, \quad j = 1,\cdots,n, \\ & x_{ij} \geq 0,\ i = 1,\cdots,m,\ j = 1,\cdots,n. \end{aligned}$$

where

m= number of sources,

n= number of destinations,

s_i= supply at source i,

d_j= demand at destination j.

It is convenient to form such large problem randomly and its constraint-matrix is very sparse. For feasibility, it is necessary that we have

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j.$$

We generate random test problems as follows: we take

$$s_i = 80 \times ran(*) + 20 \quad \text{for } i = 1, \cdots, m,$$
$$\tilde{d_j} = 80 \times ran(*) + 20 \quad \text{for } j = 1, \cdots, n,$$

and set

$$d_j = t\tilde{d_j}, \qquad j = 1, \cdots, n.$$

where $ran(*)$ denotes a random variable in $(0,1)$ and

$$t := \sum_{i=1}^{m} s_i / \sum_{j=1}^{n} \tilde{d_j}.$$

Then the system is balanced in the sense that the total supply equals the total demand. Further we choose the components c_{ij} of the vector c, randomly in $(0,100)$.

It is well known that a linear programming problem is equivalent to a generalized linear complementarity problem of the form (1) with

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad M = \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix}, \qquad q = \begin{pmatrix} c \\ -b \end{pmatrix}$$

and $\Omega = \{u = (x, y) \mid x \geq 0\}$. Because in this case

$$e(u) = \begin{pmatrix} x - [x + (A^T y - c)]_+ \\ Ax - b \end{pmatrix},$$

the iteration stopped as soon as

$$\max \left\{ \frac{\|x^k - [x^k + (A^T y^k - c)]_+\|_\infty}{\|c\|_\infty}, \frac{\|Ax^k - b\|_\infty}{\|b\|_\infty} \right\} \leq \varepsilon$$

for some prescribed $\varepsilon > 0$. The iteration number for $\varepsilon = 10^{-3}$ are given in Table 5.

Table 5. Linear Programming, Transportation Problem

# orig.	# dest.	# var.	# it.	original PC	# it.	new PC
m	n	mn	$\gamma = 1$	γ=1.5	γ=1.5	γ=1.95
40	50	2000	685	777	376	335
50	100	5000	719	906	542	495
80	125	10000	817	1031	601	564

The start vector u^0 has each component equal to 0.

The numerical results show that for the original PC method $\gamma = 1$ is better than $\gamma > 1$. For the new PC method, the best choice of γ should be close to 2 (because in the proof of Theorem 3 many inequalities were used). The new PC method require 30-40 percent fewer iterations than the original one.

6. Conclusion

In this paper, we have modified the PC method in [4] and demonstrated its effectiveness for some examples. The method can be implemented to the linear complementarity problem which might otherwise be excluded by some other algorithms (such as when M is skew symmetric). Moreover, since a linear variational inequality

$$\text{(LVI)} \qquad u \in \Omega, \qquad (v-u)^T(Mu+q) \geq 0, \qquad \forall v \in \Omega$$

is equivalent to the following projection equation [2]

$$u = P_\Omega[u-(Mu+q)],$$

the new method (with $\rho = \rho_{\text{new}}$) can be used to solve linear variational inequality, when Ω is convex and the projection on Ω is simple to carry out.

In general, as an iterative method, the PC-method is considerably simpler and advantageous for large sparse problems. However, we want to point out, although the PC-method performs here well, as most iterative methods, it is easy to construct a small example for which the PC methods run very poorly. To overcome this disadvantage is a topic of our present research.

References

[1] E. Blum and W. Oettli, Mathematische Optimierung. Econometrics and Operations Research XX, Springer-Verlag Berlin Heidelberg New York, 1975.

[2] P.G. Ciarlet, Introduction to matrix numerical analysis and optimization. Collection of Applied Mathematics for the Master's Degree, Masson, Paris, 1982.

[3] P.T. Harker and J.S. Pang, A damped-Newton method for the linear complementarity problem. *Lectures in Applied Mathematics*, 26(1990), 265–284.

[4] B.S. He, A projection and contraction method for a class of linear complementarity problem and its application in convex quadratic programming, *Applied Mathematics and Optimization*, 25(1992), 247–262.

[5] B.S. He, On a class of iterative projection and contraction methods for linear programming, *Journal of Optimization Theory and Applications*, 78(1993), 247–266.

[6] B.S. He, Further developments in an iterative projection and contraction method for linear programming, *Journal of Computational Mathematics*, 11(1993), 350–364.

[7] B.S. He, A new method for a class of linear variational inequalities, *Mathematical Programming*, 66(1994), 137–144.

[8] B.S. He and J. Stoer, Solution of projection problems over polytopes. *Numerische Mathematik*, 61(1992), 73–90.

[9] K.G. Murty, Linear Complementarity, Linear and Nolinear Programming. Helderman Verlag, Berlin, 1988.