Research and Communications in Mathematics and Mathematical Sciences
Vol. 4, Issue 2, 2014, Pages 69-91
ISSN 2319-6939
Published Online on August 5, 2014

http://jyotiacademicpress.net

PROJECTIVE ITERATIVE METHODS FOR SOLVING LINEAR COMPLEMENTARITY PROBLEMS: A SURVEY

M. T. YAHYAPOUR and S. A. EDALATPANAH

Department of Mathematics
Ramsar Branch
Islamic Azad University
Ramsar
Iran
e-mail: saedalatpanah@gmail.com

Abstract

In this paper, we study some important models of projective iterative methods for linear complementarity problems (LCPs). These algorithms have simple and graceful structure and can be applied to other complementarity problems. Asymptotic convergence of the sequence generated by the method to the unique solution of this class of LCPs is established. Finally, numerical results are also given to illustrate the efficiency of these algorithms.

1. Introduction

For a given real vector $q \in R^n$ and a given matrix $A \in R^{n\times n}$, the linear complementarity problem abbreviated as $LCP(A, q)$, consists in finding vectors $z \in R^n$ such that

2010 Mathematics Subject Classification: 65F10, 65H10, 90C33.
Keywords and phrases: linear complementarity problems, iterative methods, GAOR, MAOR, AOR, SOR, SSOR, Gauss-Seidel, H-matrix.
Communicated by S. Ebrahimi Atani.
Received April 17, 2014; Revised July 14, 2014

$$\begin{cases} w = Az + q, \\ z \geq 0,\ w \geq 0, \\ z^T w = 0, \end{cases} \tag{1.1}$$

where z^T denotes the transpose of the vector z. The linear complementarity problems (LCPs) is one of the fundamental problems in optimization and mathematical programming [1].

Many problems in various scientific computing, economics, and engineering areas can lead to the solution of LCP and its generalizations. For example, quadratic programming, Nash equilibrium point of a bimatrix game, nonlinear obstacle problems, invariant capital stock, optimal stopping, contact and structural mechanics, free boundary problem for journal bearings, traffic equilibriums, manufacturing systems, etc.. For more details, see [1-3] and the references therein. So, many direct and iterative methods have been developed for its solution; see [3].

One of the oldest iterative methods related to the linear complementarity problem is due to Hildereth [4], who designed the procedure to solve a strictly convex quadratic program. Hildereth stated its Kuhn-Tucker conditions and used the nonsingularity of the Hessian matrix of the objective function to eliminate the primal variables. What remains after this operation is a linear complementarity problem in which the variables are Lagrange multipliers and the matrix A is symmetric and positive semi-definite. A more general iterative method, attributed to Christopherson [5], has been analyzed and clarified by Cryer [6, 7], and it is often cited as Cryer's method. It is a successive over-relaxation (SOR) method proposed for the solution of the free-boundary problem for journal bearings; see also [8, 9]. Much attention has recently been paid on a class of iterative methods called the matrix-splitting method [10-23]. Matrix splitting method for LCP exploits particular features of matrices such as the sparsity and the block structure and in these methods, most convergence results have been established for the case that the system matrix A is symmetric positive definite, M-matrix, H-matrix or diagonally dominant. The case that A is non-Hermitian is much more difficult and, as we have known, only a few results were reported about this case of matrix for $LCP(A, q)$.

In this paper, we study all of these models of projective iterative methods for linear complementarity problems (LCPs). Finally, numerical results are also given to illustrate the efficiency of these algorithms.

2. Prerequisite

We begin with some basic notation and preliminary results which we refer to later.

Definition 2.1 ([24, 25]).

(a) A matrix $A = [a_{ij}]$ is nonnegative (positive) if $a_{ij} \geq 0\,(a_{ij} > 0)$. In this case, we write $A \geq 0\,(A > 0)$. Similarly, for n-dimensional vectors x, by identifying them with $n \times 1$ matrices, we can also define $x \geq 0\,(x > 0)$.

(b) A matrix $A = (a_{ij})_{n\times n}$ is called a *Z-matrix* if for any $i \neq j$, $a_{ij} \leq 0$.

(c) *Z-matrix* is an *M-matrix*, if A is nonsingular, and $A^{-1} \geq 0$.

(d) A matrix $A = (a_{ij})_{n\times n}$ is called *M-matrix* if $A = \alpha I - B$; $B \geq 0$ and $\alpha > \rho(B)$; (we denote the spectral radius of B by $\rho(B)$).

(e) For any matrix $A = (a_{ij})_{n\times n}$, the comparison matrix $\langle A\rangle = (m_{ij}) \in R^{n\times n}$ is defined by

$$m_{ii} = |a_{ii}|, \quad m_{ij} = -|a_{ij}|, \quad i \neq j \quad 1 \leq i, j \leq n.$$

(f) $A = (a_{ij})_{n\times n}$ is an *H-matrix* if and only if $\langle A\rangle$ is *M-matrix*.

Definition 2.2 ([10]). For $x \in R^n$, vector x_+ is defined such that $(x_+)_j = \max\{0, x_j\}$, $j = 1, 2, \ldots, n$. Then, for any $x, y \in R^n$, the following facts hold:

(1) $(x + y)_+ \leq x_+ + y_+$;

(2) $x_+ - y_+ \leq (x - y)_+$;

(3) $|x| = x_+ + (-x)_+$;

(4) $x \leq y$ implies $x_+ \leq y_+$.

Definition 2.3 ([24, 25]). Let A be a real matrix. The splitting $A = M - N$ is

(a) convergent if $\rho(M^{-1}N) < 1$;

(b) regular if $M^{-1} \geq 0$ and $N \geq 0$;

(c) weak regular if $M^{-1} \geq 0$ and $M^{-1}N \geq 0$;

(d) *M-splitting* if *M* is *M-matrix* and $N \geq 0$.

Clearly, an *M*-splitting is regular and a regular splitting is weak regular.

Lemma 2.1 ([10]). *Let* $A \in R^{n \times n}$ *be an H-matrix with positive diagonal elements. Then the* $LCP(A, q)$ *has a unique solution* $z^* \in R^n$.

Lemma 2.2 ([10]). $LCP(A, q)$ *can be equivalently transformed to a fixed-point system of equations*

$$z = (z - \alpha E(Az + q))_+,$$

where α *is some positive constant and E is a diagonal matrix with positive diagonal elements.*

3. Projective Iterative Methods

Let us to consider LCP (1.1). We knows that since $LCP(A, q)$ is equivalent to the following zero-finding formulation:

$$\min(z, (Az + q)) = 0.$$

And the zero-finding formulation is equivalent to the following fixed-point formulation:

$$\max(0, z - (Az + q)) = z.$$

Then for any iteration, we have

$$A = M - N,$$

$$\max(0, z^{k+1} - (Mz^{k+1} - Nz^k + q)) = z^{k+1}.$$

We know that, if $(z^{k+1} - (Mz^{k+1} - Nz^k + q))_i < 0$, then $z_i^{k+1} = 0$, otherwise

$$(z^{k+1} - (Mz^{k+1} - Nz^k + q))_i = z_i^{k+1},$$

$$\Rightarrow z_i^{k+1} - (Mz^{k+1} - Nz^k + q)_i = z_i^{k+1},$$

$$\Rightarrow Mz_i^{k+1} = Nz_i^k - (q)_i = -((q)_i + (A - M)z_i^k),$$

$$\Rightarrow Mz_i^{k+1} = (Mz^k - (q + Az^k))_i.$$

Now, for example, if

$$M = D - L,$$

where D, L are diagonal, strictly lower triangular parts of A, in order to solve $LCP(A, q)$, we have

$$(D - L)z_i^{k+1} = ((D - L)z^k - (q + Az^k))_i,$$

$$\Rightarrow Dz_i^{k+1} = (D - L)z_i^k - (q + Az^k)_i + Lz_i^{k+1},$$

$$\Rightarrow z_i^{k+1} = z_i^k - (D)^{-1} Lz_i^k - (D)^{-1}(q + Az^k)_i + (D)^{-1} Lz_i^{k+1}.$$

Therefore, we get following formula of projective iterative method:

$$z^{k+1} = \max(0,\ z^k - (D)^{-1}(q + (A + L)z^k - Lz^{k+1})).$$

Now, we study some important models of projective iterative methods.

3.1. GAOR projective methods for $LCP(A, q)$

Let the matrix A be as

$$A = D + L + U, \tag{3.1}$$

where D diagonal, L and U are strictly lower and upper triangular matrices of A, respectively. Then by choice of $\alpha E = D^{-1}$ and Lemma 2.2 and above demonstration, we have

$$z = (z - D^{-1}(Az + q))_+. \tag{3.2}$$

So, in order to solve $LCP(A, q)$, generalized accelerated over-relaxation (GAOR) iterative methods defined in [13] is

$$z^{k+1} = (z^k - D^{-1}[\alpha\Omega Lz^{k+1} + (\Omega A - \alpha\Omega L)z^k + \Omega q])_+, \tag{3.3}$$

where α is a real parameter and $\Omega = (w_1, w_2, \ldots, w_n)$ is a real diagonal relaxation matrix. The operator $f : R^n \longrightarrow R^n$, is defined such that $f(z) = \xi$, where ξ is the fixed point of the system

$$\xi = (z - D^{-1}[\alpha\Omega L\xi + (\Omega A - \alpha\Omega L)z + \Omega q])_+. \tag{3.4}$$

In next theorem, we have the convergence theorem, proposed in [13] for the GAOR methods.

Theorem 1. *Let* $A \in R^{n\times n}$ *be an H-matrix with positive diagonal elements. Moreover, let*

$$G = I - \alpha\Omega D^{-1}|L|, \quad F = \left|I - D^{-1}(\Omega M - \alpha\Omega L)\right|,$$

then, for any initial vector $z^0 \in R^n$, *the iterative sequence* $\{z^k\}$ *generated by the GAOR method converges to the unique solution* z^* *of the* $LCP(A, q)$ *and*

$$\rho(G^{-1}F) \leq \max_{1 \leq i \leq n}\{|1 - w_i| + w_i \rho(|J|)\} < 1,$$

if

$$0 < w_i < \frac{2}{1 + \rho(|J|)}, \quad 0 \leq \alpha \leq 1,$$

where $\rho(|J|)$ *is the spectral radius Jacobi iteration matrix* $(J = D^{-1}(L + U))$.

Corollary 1. *By choosing special parameters in GAOR methods, it can be obtained the similar results for other well-known iterative methods. For example,*

(1) *GSOR (generalized SOR) methods* [13] *for* $\alpha = 1$.

(2) *AOR (accelerated overrelaxation) methods* [29] *for* $\alpha = r / w$ *and* $\Omega = wI$.

(3) *EAOR (extrapolated AOR) methods* [30] *for* $\alpha = r^2/w^2$ *and* $\Omega = (w^2/r)I$.

(4) *SOR methods* [24, 25] *for* $\alpha = 1$ *and* $\Omega = wI$.

(5) *JOR (Jacobi overrelaxation) methods* [31] *for* $\alpha = 0$ *and* $\Omega = wI$.

(6) *Gauss-Seidel method* [24, 25] *for* $\alpha = 1$ *and* $\Omega = I$.

(7) *Jacobi method for* [24, 25] *for* $\alpha = 0$ *and* $\Omega = I$.

3.2. MAOR projective methods for $LCP(A, q)$

Consider Equation (3.1). So, in order to solve $LCP(A, q)$, where A is the following form:

$$A = \begin{bmatrix} D_1 & H \\ K & D_2 \end{bmatrix},$$

where D_1 and D_2 are nonsingular diagonal matrices of orders n_1 and n_2, respectively, and $H \in R^{n_1 \times n_2}$, $K \in R^{n_2 \times n_1}$ and

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 \\ K & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & H \\ 0 & 0 \end{bmatrix}.$$

The modified accelerated over-relaxation (MAOR) iterative methods is defined in [11] as follows:

$$z^{(k+1)} = (z^{(k)} - D^{-1}[\gamma L z^{(k+1)} + (\Omega A - \gamma L) z^{(k)} + \Omega q])_{+}, \tag{3.5}$$

where $\Omega = \operatorname{diag}(w_1 I_1, w_2 I_2)$, $\Gamma = \operatorname{diag}(\gamma_1 I_1, \gamma_2 I_2)$ with $w_1, w_2 \neq 0$ and $I_1 \in R^{n_1 \times n_2}$.

Let

$$Q = I - \gamma D^{-1}|L| \quad \text{and} \quad R = \left|I - D^{-1}(\Omega M - \gamma L)\right|.$$

Then in next theorem, we have the convergence theorem, proposed in [11] for the MAOR method.

Theorem 2. *Let $A \in R^{n \times n}$ be an H-matrix with positive diagonal elements. Then, for any initial vector $z^0 \in R^n$, the iterative sequence $\{z^k\}$ generated by the MAOR method converges to the unique solution z^* of the $LCP(A, q)$ and*

$$\rho(Q^{-1}R) \leq \operatorname*{Max}_{I=1,2}\{|1 - w_i| + w_i \rho(|J|)\} < 1,$$

whenever

$$0 < w_i < \frac{2}{1 + \rho(|J|)}, \quad 0 \leq \gamma \leq w_2,$$

where

$$J = D^{-1}(L + U).$$

3.3. Symmetric projective methods for $LCP(A, q)$

In order to solve $LCP(A, q)$, symmetric successive over-relaxation (SSOR) iterative methods is defined in [15] as follows:

$$z^{k+1} = (z^k - D^{-1}[-wLz^{k+1} + (w(2-w)A - wL)z^k + w(2-w)q])_+. \quad (3.6)$$

Also they proposed the following model:

$$z^{k+1} = (z^k - D^{-1}[-wUz^{k+1} + (w(2-w)A - wU)z^k + w(2-w)q])_+, \quad (3.7)$$

where $0 < w < 2$.

Let

$$Q = I - wD^{-1}|L| \quad \text{and} \quad R = \left|I - D^{-1}(w(2-w)M - wL)\right|.$$

In next theorem, we have the convergence theorem, proposed in [15] for the SSOR method.

Theorem 3. *Let $A \in R^{n\times n}$ be an H-matrix with positive diagonal elements and $0 < w < 2$. Then, for any initial vector $z^0 \in R^n$, the iterative sequence $\{z^k\}$ generated by the SSOR method converges to the unique solution z^* of the $LCP(A, q)$ and $\rho(Q^{-1}R) < 1$.*

3.4. SOR-Like method for non-Hermitian positive definite $LCP(A, q)$

Let the matrix A is split as Equation (3.1).Then the iterative method successive over-relaxation like method (SOR-like method) for $LCP(A, q)$ by the following [22]:

Algorithm 1: SOR-Like method for LCP

Step 1. Choose an initial vector $z^0 \in R^n$ parameter w and set $k = 0$.

Step 2. For $k = 0, 1, 2, \ldots$ do

$$z^{k+1} = (z^k - wD^{-1}[(L - U^*)z^{k+1} + (D + U + U^*)z^k + q])_+.$$

Step 3. If $z^{k+1} = z^k$, then stop; otherwise, set $k = k + 1$ and go to Step 2.

Note. U^* denotes the conjugate transpose of the matrix U.

In next theorem, we have the existence and uniqueness of the solution of SOR-like method proposed in [22], when the coefficient matrix is non-Hermitian positive definite.

Theorem 4. *Let $A \in C^{n \times n}$ be non-Hermitian positive definite with $H = (A + A^*)/2$ its Hermitian part, $\eta = \lambda_{\min}(B)$ be the smallest eigenvalue of $B = H - 2D^{-1}(U + U^*)$ and*

$$\begin{cases} w \in (0, 1], & \text{if } \eta \geq 0, \\ w \in (0, 1), & \text{if } \eta = 0, \\ w \in (0, \frac{2}{2 - \eta}), & \text{if } \eta < 0. \end{cases}$$

Then for any initial vector z^0, Algorithm 1 convergence to the unique solution of $LCP(A, q)$.

3.5. Krylov subspace methods for $LCP(A, q)$

Here, we survey the Krylov subspace methods for LCP [23]. We know these methods are based on refinement residuals algorithms. Then if we consider the MAOR for LCP based on refinement methods, we have the following algorithm:

Algorithm 2: MAOR (A, q)

Step 1. Choose an initial vector $z^{(0)} \in R^n$, parameter w.

Step 2. For $k = 0, 1, 2, \ldots$ do

$$r^{(k+1)} = -q - [\gamma\Omega^{-1}Lz^{(k+1)} + (A - \gamma\Omega^{-1}L)z^{(k)}],$$

$$z_i^{(k+1)} = z_i^{(k+1)} + \Omega D^{-1}r_i^{(k+1)},$$

$$z_i^{(k+1)} = \max\{0, z_i^{(k+1)}\}.$$

Step 3. If $z^{(k+1)} = z^{(k)}$, then stop; otherwise, set $k = k+1$ and go to Step 2.

Therefore, based on the concept of refinement methods and Algorithm 2, we can explain the Krylov subspace methods for LCP. Now, we shortly describe a Krylov subspace method called conjugate gradient squared method (CGS) for $LCP(A, q)$.

The conjugate gradient squared method is a related algorithm that attempts to improvement the some problems of bi-conjugate gradient method (BiCG). Additionally, often one observes a speed of convergence for CGS that is about twice as fast as for the biconjugate gradient method; see [32].

Algorithm 3: CGS for $LCP(A, q)$

1. Compute $r^{(0)} = -q - Az^{(0)}$.

2. Set $\tilde{r} = r^{(0)}$.

3. For $k = 1, \ldots,$ until convergence do

$$\rho_{(k-1)} = \tilde{r}^T r^{(k-1)},$$

if $\rho_{k-1} = 0$, then method fails,

if $k = 1$

$$u^{(1)} = r^{(1)},$$

$$l^{(1)} = u^{(1)},$$

else

$$\beta_{k-1} = \rho_{k-1} / \rho_{k-2}$$

$$u^{(k)} = r^{(k-1)} + \beta_{k-1} y_{k-1},$$

$$l^{(k)} = u^{(k)} + \beta_{k-1}(y_{k-1} + \beta_{k-1} l^{(k-1)}),$$

end if

Solve $\hat{l} = l^{(k)}$,

$$\hat{v} = A\hat{l},$$

$$\alpha_k = \rho_{k-1} / \tilde{r}^T \hat{v},$$

$$y_k = u_k - \alpha_k \hat{v},$$

Solve $\hat{u} = u^{(k)} + y_k$,

$$\hat{l} = A\hat{u},$$

$$z^{(k)} = z^{(k-1)} + \alpha_k \hat{u},$$

$$z^{(k)} = \max\{0, z^{(k)}\},$$

$$r^{(k)} = -q - Az^{(k)},$$

4. end.

Remark 1. All of these techniques and their results are also applicable for parallel computing such as *multisplitting* methods [10, 28, 34-36], SIMD and MIMD systems [26, 27].

4. Numerical Examples

Here we give some examples, to illustrate the results obtained in previous section. The initial approximation of z^0 is $z^0 = (1, 1, \ldots, 1)^T$ and as a stopping criterion we choose

$$\begin{cases} \left\| \min(Mz^k + q, z^k) \right\|_\infty \leq 10^{-6}, \\ \left\| \min(\overline{M}z^k + \overline{q}, z^k) \right\|_\infty \leq 10^{-6}. \end{cases}$$

Furthermore, we report the CPU time and number of iterations for the corresponding iterative methods by CPU and Iter, respectively. All the numerical experiments presented in this section were computed with MATLAB 7 on a PC with a 1.86GHz 32-bit processor and 1GB memory.

Example 4.1. Consider $LCP(A, q)$ as

$$\begin{cases} A = I \otimes B + R \otimes I \in R^{N \times N}, \\ q = (-1, 1, \ldots, (-1)^{n^2})^T \in R^N, \end{cases}$$

where $I \in R^{N \times N}$ and $\otimes$ denotes the Kronecker product. Furthermore, B and R are $n \times n$ tridiagonal matrices given by

$$\begin{cases} B = \text{tridiagonal} \left[-\left(\frac{2+h}{8} \right), 1, -\left(\frac{2-h}{8} \right) \right], \\ R = \text{tridiagonal} \left[-\left(\frac{1+h}{4} \right), 0, -\left(\frac{1-h}{4} \right) \right], \\ \& \ h = 1/n; \ N = n^2. \end{cases}$$

Evidently, A is an *H-matrix* with positive diagonal elements so, $LCP(A, q)$ has a unique solution. Then, we solved the $n^2 \times n^2$ *H-matrix* yielded by the iterative methods.

In Table 1, we report the CPU time and the number of iterations for the corresponding GAOR methods. Also, the N parameters w_i are taken from the N equal-partitioned points of the interval [0.9, 1.1] and alpha is one.

Table 1. It shows the results of Example 4.1 for GAOR

Method	GAOR	
n	**Iter**	**CPU**
7	53	0.070
25	315	19.520

In Table 2, we report the CPU time and the number of iterations by different n for the corresponding AOR methods with $(w = 1, r = 0.8)$.

Table 2. It shows the results of Example 4.1 for AOR

Method	GAOR	
n	**Iter**	**CPU**
9	94	0.130
18	248	3.800
25	375	27.689

In Table 3, we report the CPU time and the number of iterations by different n for the corresponding SOR methods with $(w = 0.9)$.

Table 3. It shows the results of Example 4.1 for SOR

Method	GAOR	
n	**Iter**	**CPU**
10	111	0.270
20	288	5.538
30	460	72.554

Example 4.2 (Application to the obstacle problems).

The test problem comes from the finite difference discretization of the one side obstacle problem [33],

$$\langle -\Delta u - b, \nu - u \rangle \geq 0, \quad \forall \nu \in K,$$

where $K = \{\nu \in H_0^1(\Omega) : \nu \geq 0\}$, $b = 4\sin(4xy)$, $\Omega = (0, 1) \times (0, 1)$. By discretization, we obtain the problem as $LCP(M, q)$, where

$$h = 1/m, \quad n = m^2, \quad q = (4h^2 \sin(4ij/m^2))_{ij}, \quad i, j = 1, \ldots, m,$$

$$M = \begin{bmatrix} A & -I & & & \\ -I & A & & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & A \end{bmatrix} \in R^{n \times n},$$

where I is the identity matrix of m-dimension, and

$$A = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 4 \end{bmatrix} \in R^{m \times m}.$$

In Table 4, we report the CPU time and the number of iterations for the corresponding GAOR methods. Furthermore, the N parameters w_i, are taken from the N equal-partitioned points of the interval [1, 1.02] and alpha is one.

Table 4. It shows the results of Example 4.2

Method	GAOR	
n	**Iter**	**CPU**
100	102	0.004054
225	198	0.064214
400	345	0.289446
625	472	0.839852
900	608	1.960319
1225	744	3.688457
1600	907	8.138793
2025	1063	25.073047
2500	1249	27.860959

Example 4.3. Consider $LCP(A, q)$ with following system:

$$A = G \otimes I \otimes I + I \otimes F \otimes I + I \otimes I \otimes F \in R^{N \times N},$$

$$q = (-1, 1, \dots, (-1)^{n^3})^T \in R^N,$$

where $I \in R^{N \times N}$. Also G and F are $n \times n$ tridiagonal matrices given by;

$$G = \text{tridiagonal}\left[-\left(\frac{2+2h}{12}\right), 1, -\left(\frac{2-2h}{12}\right)\right],$$

$$F = \text{tridiagonal}\left[-\left(\frac{2+h}{12}\right), 0, -\left(\frac{2-h}{12}\right)\right],$$

and $h = 1/n$; $N = n^3$.

Then, we solved the $n^3 \times n^3$ *H-matrix* yielded by the iterative methods. In Table 5, with several values, we report the CPU time (CPU) and the number of iterations (Iter) for the corresponding SSOR methods (when $N = 1000$).

Table 5. The results of Example 4.3 for SSOR

Method	SSOR	
w	**Iter**	**CPU**
0.02	615	110.788
0.1	142	24.391
0.2	74	12.523
0.4	37	6.112
0.7	20	3.390
0.9	15	2.607
1.2	13	1.863

Example 4.4. Consider some randomly generated LCP with non-Hermitian positive definite A, where the data (A, q) are generated by the Matlab scripts:

```
function random for LCP(n)
rand ('state', 0);
R = rand (n, n);
A = R + n* eye (n);
q = rand (n, 1).
```

In Table 6, we report the CPU time (CPU) and the number of iterations (Iter) for the corresponding SOR-like method for different values of n and w.

Table 6. It shows the results of Example 4.4 for SOR-like

Method		SOR-Like		
n	the sign of η	w	**Iter**	**CPU**
100	+	0.1	135	0.005278
		0.5	24	0.001008
		1.0	19	0.000720
1000	+	0.1	145	0.399574
		0.5	26	0.070849
		1.0	19	0.051053
1500	+	0.1	147	0.901282
		0.5	26	0.158273
		1.0	19	0.116798

Example 4.5. Consider $LCP(A, q)$ with $A \in R^{(n_1+n_2)\times(n_1+n_2)}$ and $q \in R^{(n_1+n_2)}$

$$A = \begin{bmatrix} D_1 & H \\ K & D_2 \end{bmatrix};$$

$D_1 = \alpha_1 \times I_{n_1 \times n_1}$; $D_2 = \alpha_2 \times I_{n_2 \times n_2}$;

$$H = \begin{bmatrix} \frac{-1}{4} & 0 & \frac{-1}{4} & \cdots & \cdots & 0 \\ 0 & \frac{-1}{4} & 0 & \frac{-1}{4} & \cdots & 0 \\ \frac{-1}{6} & 0 & \frac{-1}{4} & \ddots & \ddots & \vdots \\ \frac{-1}{6} & \frac{-1}{6} & 0 & \frac{-1}{4} & \ddots & \frac{-1}{4} \\ 0 & \frac{-1}{6} & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \frac{-1}{4} \\ 0 & \cdots & 0 & \frac{-1}{6} & \frac{-1}{6} & 0 \end{bmatrix}_{n_1 \times n_2} \quad \text{and } K = H^T.$$

And

$$q = (-1, 1, \ldots, (-1)^{(n_1+n_2)})^T \in R^{(n_1+n_2)}.$$

Then, we solved the $(n_1 + n_2) \times (n_1 + n_2)$ *H-matrix* yielded by the MAOR iterative methods.

In Table 7, we report the CPU time (CPU) and the number of iterations (Iter) for the corresponding MAOR methods for different values of α_1, α_2, n_1, n_2 and ($w_1 = 0.89$, $w_2 = 0.92$, and $\gamma_2 = 0.88$).

Table 7. It shows the results of Example 4.5 for MAOR

Method				MAOR	
α_1	α_2	n_1	n_2	**Iter**	**CPU**
0.5	1.5	5	9	21	0.000314
0.5	1.5	9	9	40	0.000471
0.6	1.6	10	15	18	0.000292
0.6	1.6	15	10	32	0.000465
0.6	1.6	15	15	50	0.000935
0.6	1.7	900	600	99	0.619097
0.6	1.7	600	900	99	0.621756
0.6	1.7	900	900	99	0.887001

In Table 8, we report the CPU time (CPU) and the number of iterations (Iter) for the corresponding CGS method for different values of α_1, α_2, n_1, n_2.

Table 8. It shows the results of Example 4.5 for CGS

Method				CGS	
α_1	α_2	n_1	n_2	**Iter**	**CPU**
0.5	1.5	5	9	10	0.003633
0.5	1.5	9	5	11	0.003954
0.5	1.5	9	9	14	0.005302
0.6	1.6	10	15	13	0.003690
0.6	1.6	15	10	13	0.004340
0.6	1.6	15	15	15	0.006117
0.6	1.7	900	600	71	3.530221
0.6	1.7	600	900	71	3.966532
0.6	1.7	900	900	71	6.127211

5. Conclusion

In this paper, we have studied the projection iterative methods for linear complementarity problem and established the convergence for these methods under certain conditions. Furthermore, we have used the Krylov subspace methods such as conjugate gradient squared method for LCP. The results show that these methods are efficient.

Acknowledgement

The authors would like to thank Ramsar Branch of Islamic Azad University for the financial support of this research, which is based on a research project contract.

References

[1] K. G. Murty, Linear Complementarity, Linear and Nonlinear Programming, Heldermann Verlag, Berlin, 1988.

[2] M. S. Bazaraa, H. D. Sheral and C. M. Shetty, Nonlinear Programming, Theory and Algorithms, Third Edition, Wiley-Interscience, Hoboken, NJ, 2006.

[3] R. W. Cottle, J. S. Pang and R. E. Stone, The Linear Complementarity Problem, Academic Press, London, 1992.

[4] C. Hildereth, Point estimates of ordinates of concave function, Journal of the American Statistical Association 49 (1954), 598-619.

[5] D. G. Christopherson, A new mathematical method for the solution of film lubrication problems, Institute of Mechanical Engineers, Proceedings 146 (1941), 126-135.

[6] C. W. Cryer, The method of Christopherson for solving free boundary problems for infinite journal bearings by means of finite differences, Mathematics of Computation 25 (1971), 435-443.

[7] C. W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, SIAM J. Control 9 (1971), 385-392.

[8] A. A. Rainondi and J. Boyd, A solution for the finite journal bearing and its application to analysis and design, III, Transactions of the American Society of Lubrication Engineers 1 (1958), 194-209.

[9] V. M. Friedman and V. S. Chernina, An iteration process for the solution of the finite dimensional contact problem, USSR Computational Mathematics and Mathematical Physics 8 (1967), 210-214.

[10] Z. Z. Bai and D. J. Evans, Matrix multisplitting relaxation methods for linear complementarity problems, Int. J. Comput. Math. 63 (1997), 309-326.

[11] D. Yuan and Y. Z. Song, Modified AOR methods for linear complementarity problem, Appl. Math. Comput. 140 (2003), 53-67.

[12] Lj. Cvetković and S. Rapajić, How to improve MAOR method convergence area for linear complementarity problems, Appl. Math. Comput. 162 (2005), 577-584.

[13] Y. Li and P. Dai, Generalized AOR methods for linear complementarity problem, Appl. Math. Comput. 188 (2007), 7-18.

[14] M. H. Xu and G. F. Luan, A rapid algorithm for a class of linear complementarity problems, Appl. Math. Comput. 188 (2007), 1647-1655.

[15] M. Dehghan and M. Hajarian, Convergence of SSOR methods for linear complementarity problems, Operations Research Letters 37 (2009), 219-223.

[16] Z. Z. Bai, Modulus-based matrix splitting iteration methods for linear complementarity problems, Numer. Linear Algebra Appl. 17 (2010), 917-933.

[17] X. Han, D. Yuan and D. S. Jiang, Two SAOR iterative formats for solving linear complementarity problems, Int. J. Information Technology and Computer Science 2 (2011), 38-45.

[18] H. Saberi Najafi and S. A. Edalatpanah, On the two SAOR iterative formats for solving linear complementarity problems, Int. J. Information Technology and Computer Science 3(5) (2011), 19-24.

[19] H. Saberi Najafi and S. A. Edalatpanah, A kind of symmetrical iterative methods to solve special class of LCP (M, q), International Journal of Applied Mathematics and Applications 4(2) (2012), 183-189.

[20] H. Saberi Najafi and S. A. Edalatpanah, On the convergence regions of generalized AOR methods for linear complementarity problems, Journal of Optimization Theory and Applications 156 (2013), 859-866.

[21] H. Saberi Najafi and S. A. Edalatpanah, Iterative methods with analytical preconditioning technique to linear complementarity problems: Application to obstacle problems, RAIRO-Operations Research 47 (2013), 59-71. doi:10.1051/ro/2013027.

[22] H. Saberi Najafi and S. A. Edalatpanah, SOR-like methods for non-Hermitian positive definite linear complementarity problems, Advanced Modeling and Optimization 15 (2013), 697-704.

[23] H. Saberi Najafi and S. A. Edalatpanah, Modification of iterative methods for solving linear complementarity problems, Engineering Computations 30(7) (2013), 910-923.

[24] R. S. Varga, Matrix Iterative Analysis, Second Edition, Springer, Berlin, 2000.

[25] A. Berman and R. J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York, 1979.

[26] O. L. Mangasarian and R. De Leone, Parallel successive overrelaxation methods for symmetric linear complementarity problems and linear programs, J. Optim. Theory Appl. 54 (1987), 437-446.

[27] R. De Leone and O. L. Mangasarian, Asynchronous parallel successive overrelaxation for the symmetric linear complementarity problem, Math. Programming 42 (1988), 347-361.

[28] Z. Z. Bai and D. J. Evans, Matrix multisplitting methods with applications to linear complementarity problems: Parallel asynchronous methods, Int. J. Comput. Math. 79 (2002), 205-232.

[29] A. Hadjidimos, Accelerated overrelaxation method, Math. Comput. 32 (1978), 149-157.

[30] D. J. Evans and M. M. Martins, On the convergence of the extrapolated AOR method, Int. J. Comput. Math. 43 (1992), 161-171.

[31] B. Truyen and J. Cornelis, Adiabatic layering: A new concept of hierarchical multi-scale optimization, Neural Networks 8 (1995), 1373-1378.

[32] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, PA, 1994.

[33] P. G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland Publishing Company, Amsterdam, 1978.

[34] N. Machida, M. Fukushima and T. Ibaraki, A multisplitting method for symmetric linear complementarity problems, J. Comput. Appl. Math. 62 (1995), 217-227.

[35] Z. Z. Bai, The convergence of parallel iteration algorithms for linear complementarity problems, Comput. Math. Appl. 32 (1996), 1-17.

[36] Z. Z. Bai, On the convergence of the multisplitting methods for the linear complementarity problem, SIAM J. Matrix Anal. Appl. 21 (1999), 67-78.

■