ITERATIVE ALGORITHMS FOR THE LINEAR COMPLEMENTARITY PROBLEM

Nicholas W. Kappel
Layne T. Watson

CS84002-R

ITERATIVE ALGORITHMS FOR THE LINEAR COMPLEMENTARITY PROBLEM[+]

Nicholas W. KAPPEL
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Layne T. WATSON
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

ABSTRACT. Direct complementary pivot algorithms for the linear complementarity problem with P-matrices are known to have exponential computational complexity. The analog of Gauss-Seidel and SOR iteration for linear complementarity problems with P-matrices has not been extensively developed. This paper extends some work of van Bokhoven to a class of nonsymmetric P-matrices, and develops and compares several new iterative algorithms for the linear complementarity problem. Numerical results for several hundred test problems are presented. Such indirect iterative algorithms may prove useful for large sparse complementarity problems.

Key Words: linear complementarity problem, P-matrix, iterative algorithm, fixed point iteration, principal pivot submatrix, contraction mapping, computational complexity.

[+]This work was supported in part by NSF Grant MCS 8207217.

1. INTRODUCTION.

For a matrix $M \in R^{m \times m}$ and a vector $q \in R^m$, the linear complementarity problem, denoted by (q,M), is to find vectors $x \in R^m$ and $y \in R^m$ such that

$$y = Mx + q$$

$$x \geq 0,\ y \geq 0,\ y^t x = 0.$$

The constraint $y^t x = 0$ is called the complementarity condition since for any i, $1 \leq i \leq m$, $x_i = 0$ if $y_i > 0$, and vice versa. It may be the case that $x_i = y_i = 0$ however. This problem arises in such areas as economic modeling [9,10,29], bimatrix games [18,19], mathematical programming [5,14,21], mechanics [13], lubrication [16], and numerical analysis [3].

Numerous algorithms exist to solve linear complementarity problems. Among the more important are Lemke's complementary pivot algorithm [18], Cottle and Dantzig's principal pivot method [2], Bard-type algorithms [1,pp. 147-149, 24], and the n-cycle algorithms [31,32]. Also of interest are algorithms for solving the nonlinear complementarity problem such as the homotopy methods of Merrill [22], Eaves [6,7], Saigal [27], and Watson [31].

All of the mentioned algorithms for solving the linear complementarity problem (q,M) are based on simplex type

processes. For large scale sparse problems these pivoting methods may destroy sparsity, require too many pivots, and suffer from roundoff error. Just as matrix iterative techniques have a prominent place in the numerical solution of partial differential equations, iterative algorithms might prove important in the practical solution of large scale linear complementarity problems. van Bokhoven [30] proposed an iterative algorithm which he called the modulus algorithm. The complementary pivot and Bard-type algorithms can require 2^m iterations to solve (q,M) [26], and it is probable that the hybrid n-cycle algorithm has exponential computational complexity [32]. van Bokhoven claimed that his algorithm had polynomial complexity, but in fact the complexity depends on the spectral radius of M, and hence is not polynomial in the technical sense. Nevertheless, the modulus algorithm may prove to be very efficient on certain classes of problems.

van Bokhoven proved that the modulus algorithm works correctly when the matrix involved is a symmetric P-matrix. A P-matrix is a matrix for which all principal minors are positive [8]. This paper extends van Bokhoven's results by showing that the modulus algorithm can be applied to a class of non-symmetric P-matrices. In addition, a more efficient block version of the modulus algorithm is presented.

Finally, experimental data for the modulus algorithm, the block algorithm, and a related fixed point iteration algorithm are presented.

2. NOTATION.

Let R^m be m-dimensional real Euclidean space and $R^{m\times m}$ be the set of all real m×m matrices. For $z \in R^m$ and n a nonnegative integer, $z^{(n)}$ refers to the vector obtained after n iterations inside the first cycle of an algorithm (if there is more than 1 cycle). In a similar manner, for $\tilde{z} \in R^{\tilde{m}}$, $1 \le \tilde{m} < m$, $\tilde{z}^{(n)}$ refers to the vector obtained after n iterations inside some subsequent cycle of an algorithm. For $1 \le i \le m$, z_i refers to the i^{th} element of z, and $z_i^{(n)}$ refers to the i^{th} element of the vector obtained after n iterations.

Given $M \in R^{m\times m}$, let K and L be a partition of $\{1,\ldots,m\}$. M_{KL} denotes the submatrix of M with rows indexed by elements of K and columns indexed by elements of L. A principal pivot submatrix $\tilde{M}$ of M is obtained by applying a principal pivot transform [2] to M, and then permuting the rows and columns to produce a matrix of the form

$$\begin{bmatrix} I_{KK} & * \\ 0 & \tilde{M} \end{bmatrix}$$

A P-matrix has 2^m-1 principal pivot submatrices, each of which is also a P-matrix [2].

Given $M \in R^{m\times m}$ and $q \in R^m$, the linear complementarity problem is to find vectors $y \in R^m$ and $x \in R^m$ such that

$$y = Mx + q$$

$$x \geq 0,\ y \geq 0,\ y^t x = 0.$$

Following Murty [25,p. 484], the complementarity problem is denoted (q,M).

3. DERIVATION OF THE FIXED POINT ALGORITHM.

In this section several theorems and lemmas will be given that provide the basis for using the fixed point iteration algorithm to solve the linear complementarity problem (q,M). To accomplish this, (q,M) is transformed into another problem using the transformation

$$y = |z| - z \text{ and } x = |z| + z \qquad (1)$$

where $z \in R^m$ and $|z|$ is defined by

$$|z| = (|z_1|,|z_2|,\ldots,|z_m|)^t.$$

Theorem 1. (van Bokhoven [30])

Let $M \in R^{m\times m}$, $q \in R^m$ and assume that -1 is not an eigenvalue of M (always achievable by scaling M by a positive constant). The linear complementarity problem (q,M) is equivalent to the determination of a vector $z \in R^m$ which satisifies

$$z = -(I+M)^{-1}q + (I+M)^{-1}(I-M)|z|. \qquad (2)$$

The solutions z of this equation and the solutions x and y of (q,M) are related by

$$y = |z| - z,\quad x = |z| + z,\quad z = \tfrac{1}{2}(x-y).$$

Proof:

From transformation (1) it follows that $y \geq 0$, $x \geq 0$ and $y^t x = |z|^t|z| - z^t z = 0$ for any $z \in R^m$, and therefore the substitution guarantees that the complementarity conditions on x and y are satisfied.

Any complementary feasible pair x and y determines a vector $z = \tfrac{1}{2}(x-y)$, for which $|z| = \tfrac{1}{2}(x+y)$, and (1) holds. Hence the mapping $(x,y) \to z = \tfrac{1}{2}(x-y)$ is a one to one mapping of the set of complementary feasible vectors onto R^m. Then there is a one to one correspondence between the solutions of (q,M) and the solutions of the equation derived from (q,M) by the substitution (1). This substitution yields

$$(I+M)z = -q + (I-M)|z|.$$

By assumption (I+M) is nonsingular, so

$$z = -(I+M)^{-1}q + (I+M)^{-1}(I-M)|z|.$$

Q.E.D.

Equation (2) may be rewritten in the form

$$z = D|z| + b, \tag{3}$$

where $b = -(M+I)^{-1}q \in R^m$ and $D = (I+M)^{-1}(I-M) \in R^{m\times m}$.

In the rest of this paper references will be made to an absolute norm $\|\cdot\|$ over R^m, which has these properties:

a) $\|x\| \geq 0$ for all $x \in R^m$ and $\|x\| = 0$ if and only

if $x = 0$

b) $\|\alpha x\| = |\alpha|\|x\|$ for all $\alpha \in R$, $x \in R^m$

c) $\|a+b\| \leq \|a\| + \|b\|$ for all $a, b \in R^m$

d) $\||a|\| = \|a\|$ for all $a \in R^m$

e) $|a| \leq |b|$ implies $\|a\| \leq \|b\|$ for all $a, b \in R^m$.

Householder [12,p. 160] has demonstrated that properties d) and e) are equivalent. The standard $\|\cdot\|_\infty$ and $\|\cdot\|_2$ norms are absolute norms.

Furthermore the norm $\|A\|$ of a matrix A is defined to be the induced (or operator) norm

$$\|A\| = \sup_{x \neq 0} \|Ax\|/\|x\|$$

such that $\|Ax\| \leq \|A\|\|x\|$.

We will use the trivial fact:

Lemma 1.

$||a|-|b|| \leq |a-b|$ for $a, b \in R^m$.

For completeness a proof of the following lemma, due to van Bokhoven [30], is included.

Lemma 2.

If $\|D\| < 1$ in some absolute norm, then fixed point iteration $z^{(n+1)} = D|z^{(n)}| + b$ converges and the limit z^* of the sequence $\{z^{(n)}\}$ is the unique solution of $z = D|z| + b$.

Proof:

With $f(z) = D|z| + b$, equation (3) can be written as $z = f(z)$. For $z^1, z^2 \in R^m$,

$$\|f(z^1)-f(z^2)\| = \|D(|z^1|-|z^2|)\| \leq \|D\|\||z^1|-|z^2|\|. \quad (4)$$

By Lemma 1 and the absolute norm properties d) and e) it follows that

$$\| |z^1|-|z^2| \| = \| |z^1|-|z^2| \| \leq \| z^1-z^2 \| = \| z^1-z^2 \|.$$

Thus equation (4) yields

$$\| f(z^1)-f(z^2) \| \leq \| D \| \| z^1-z^2 \|$$

and hence f satisfies a Lipschitz condition with Lipschitz constant $\| D \|$.

If $\| D \| < 1$, the iteration $z^{(n+1)} = f(z^{(n)})$ converges by the Contraction Mapping Theorem (Henrici [11]) and

$$z^* = \lim_{n \to \infty} z^{(n)}$$

is the unique solution of $z = f(z)$.

Q.E.D.

The iteration

$$z^{(n+1)} = D|z^{(n)}| + b \tag{5}$$

will be called the fixed point iteration algorithm.

Lemma 3.

Let $\Gamma \in R^{m \times m}$ be a positive definite diagonal matrix and define

$$\| x \| = \| \Gamma x \|_p, \quad 1 \leq p \leq \infty.$$

Then $\| \cdot \|$ is an absolute norm, and

$$(\min_i \Gamma_{ii}) \| x \|_\infty \leq \| x \| \leq \left[\sum_{i=1}^{m} \Gamma_{ii}^{\,p} \right]^{1/p} \| x \|_\infty .$$

Proof:

Since Γ is positive definite and $\|\cdot\|_p$ is an absolute norm, it follows that $\|x\| = \||x|\|$, and therefore $\|\cdot\|$ is an absolute norm.

Now using the definition of the p-norm

$$\|x\| = \|\Gamma x\|_p =$$

$$\left[\sum_{i=1}^{m} |\Gamma_{ii}x_i|^p\right]^{1/p} \leq \left[\sum_{i=1}^{m} |\Gamma_{ii} \max_j |x_j||^p\right]^{1/p} =$$

$$\left[(\max_j|x_j|)^p \sum_{i=1}^{m} |\Gamma_{ii}|^p\right]^{1/p} = \max_j|x_j| \left[\sum_{i=1}^{m} |\Gamma_{ii}|^p\right]^{1/p} =$$

$$\left[\sum_{i=1}^{m} |\Gamma_{ii}|^p\right]^{1/p} \|x\|_\infty.$$

Also

$$\left[\sum_{i=1}^{m} |\Gamma_{ii}x_i|^p\right]^{1/p} \geq \left[\sum_{i=1}^{m} |\min_j \Gamma_{jj}\, x_i|^p\right]^{1/p} =$$

$$\left[(\min_j|\Gamma_{jj}|)^p \sum_{i=1}^{m} |x_i|^p\right]^{1/p} = \left[\min_j|\Gamma_{jj}| \sum_{i=1}^{m} |x_i|^p\right]^{1/p} =$$

$$\min_j \Gamma_{jj}\|x\|_p.$$

Since $\|x\|_p \geq \|x\|_\infty$,

$$\min_j \Gamma_{jj}\|x\|_p \geq \min_j \Gamma_{jj}\|x\|_\infty.$$

Therefore

$$(\min_i \Gamma_{ii})\|x\|_\infty \le \|x\| \le \left[\sum_{i=1}^{m} \Gamma_{ii}^{\,p}\right]^{1/p} \|x\|_\infty .$$

Q.E.D.

Lemma 4.

Let $\Gamma \in R^{m\times m}$ be a positive definite diagonal matrix, $D \in R^{m\times m}$, and $\|\Gamma D\Gamma^{-1}\|_p < 1$ for some p, $1 \le p \le \infty$. Then $\|D\| < 1$ in an absolute norm.

Proof:

Let the vector norm be the absolute norm defined in Lemma 3. Then

$$\|D\| = \sup_{y\neq 0} \|Dy\|/\|y\| =$$

$$\sup_{y\neq 0} \frac{\|\Gamma Dy\|_p}{\|\Gamma y\|_p} = \sup_{y\neq 0} \frac{\|\Gamma D(\Gamma^{-1}\Gamma)y\|_p}{\|\Gamma y\|_p} =$$

$$\sup_{y\neq 0} \frac{\|\Gamma D\Gamma^{-1}(\Gamma y)\|_p}{\|\Gamma y\|_p} = \sup_{w\neq 0} \frac{\|\Gamma D\Gamma^{-1}w\|_p}{\|w\|_p} = \|\Gamma D\Gamma^{-1}\|_p < 1.$$

Q.E.D.

Theorem 2.

Let I+M be invertible, $D = (I+M)^{-1}(I-M)$, Γ a positive definite diagonal matrix, and $\|\Gamma D\Gamma^{-1}\|_p < 1$ for some p, $1 \le p \le \infty$. Then (q,M) is solvable by fixed point iteration (5).

Proof:

Since I+M is invertible and D is defined as in Theorem 1, (q,M) can be transformed into the equivalent problem (3) by Theorem 1. Since Γ is a positive definite matrix and

$\|\Gamma D \Gamma^{-1}\|_p < 1$ for some p, $1 \leq p \leq \infty$, it follows from Lemma 4 that $\|D\| < 1$ in an absolute norm. Finally from Lemma 2 it follows that fixed point iteration will yield the unique solution to (3), and the (unique) solution of (q,M) can be found by using (1).

Q.E.D.

Corollary 1.

Let M satisify:

1) I+M is invertible,

2) There exists a positive definite diagonal matrix Γ and a number p, $1 \leq p \leq \infty$, such that $\|\Gamma(I+M)^{-1}(I-M)\Gamma^{-1}\|_p < 1$.

Then M is a P-matrix.

Proof:

It has been demonstrated in several different ways that M is a P-matrix if and only if the problem (q,M) has a unique solution for all $q \in R^m$. (See Ingleton [13], Samelson, Thrall, and Wesler [28], Murty [23], Watson [31], or Kelly and Watson [15] for example.)

Since the hypotheses satisfy the requirements of Theorem 2, from the proof of Theorem 2 it follows that (q,M) has a unique solution for all $q \in R^m$, and therefore M is a P-matrix.

Q.E.D.

Corollary 2.

Let M satisify the hypotheses of Corollary 1. Then all of the eigenvalues of M lie in the complex (open) right half-plane.

Proof:

The eigenvalues μ_i of $D = (I+M)^{-1}(I-M)$ are given by

$$\mu_i(D) = \frac{1-\lambda_i(M)}{1+\lambda_i(M)}, \quad i = 1,\ldots,m,$$

where $\lambda_i(M)$ are the eigenvalues of M. If an eigenvalue of M lies in the complex closed left half-plane then there exists an eigenvalue μ_i of D such that $|\mu_i| \geq 1$. Therefore the spectral radius $\rho(D) = \max|\mu_i| \geq 1$. But then for any norm, $\|D\| \geq \rho(D) \geq 1$. This contradicts the second hypothesis, and therefore all eigenvalues of M lie in the complex open right half-plane.

Q.E.D.

As examples of P-matrices that have eigenvalues with negative real parts consider the matrices

$$\begin{bmatrix} y & 1 & 0 \\ y^2/2 & y & 1 \\ 1 & 0 & y \end{bmatrix}$$

where $0<y<.1$. This example is from Watson [31].

4. CONVERGENCE OF THE MODULUS AND BLOCK ALGORITHMS.

van Bokhoven [30] derives an algorithm which he calls the modulus algorithm to solve (q,M) when M is symmetric and positive definite. This section will demonstrate that the modulus algorithm can actually be applied to a more general class of matrices M. A more efficient block version is also presented.

If $M \in R^{m\times m}$ the modulus algorithm requires at most m cycles to solve (q,M). Each cycle determines an element of x or y that is positive, and therefore the corresponding element of the other vector is zero by the complementarity condition. The problem is reduced in size by one. During each pass through this loop the current problem $(\tilde{q},\tilde{M})$ is transformed into the similar problem (3). The fixed point iteration (5) is performed a calculated number of times, say N. The magnitude of at least one element of $\tilde{z}^{(N)}$ is known to equal or exceed a test value, and the sign of this element can be used to determine if the corresponding $\tilde{x}_i$ or $\tilde{y}_i$ element is positive. The actual number of cycles needed depends on the number of x_i, y_i pairs which both equal zero. If the number of such pairs equals k then exactly m-k cycles are needed. Finally the actual values of the positive elements of x and y can be determined.

Theorem 3.

Suppose that for every principal pivot submatrix $\tilde{M}$ of $M \in R^{m\times m}$, $I+\tilde{M}$ is invertible and there exists a positive definite diagonal matrix Γ such that $\|\Gamma(I+\tilde{M})^{-1}(I-\tilde{M})\Gamma^{-1}\|_p < 1$ for some p, $1 \le p \le \infty$. Then M is a P-matrix and (q,M) is solvable by the modulus algorithm of van Bokhoven.

Proof:

First transform (q,M) into $z = D|z| + b$ as described in Theorem 1. Now let $f(z) = D|z| + b$. Then

$$\|f(z^1)-f(z^2)\| = \|D(|z^1|-|z^2|)\| \le \|D\|\|z^1-z^2\|.$$

Since $\|\Gamma(I+M)^{-1}(I-M)\Gamma^{-1}\|_p < 1$ for some p, $1 \le p \le \infty$, it follows from Lemma 4 that $\|D\| < 1$ for an absolute norm. Therefore by the Contraction Mapping Theorem (Henrici [11,pp. 99-100]) the fixed point iterates $z^{(n)} = f(z^{(n-1)})$ satisify

$$\|z^*-z^{(n)}\| \le \frac{\|D\|^n}{1-\|D\|}\|z^{(1)}-z^{(0)}\| \qquad (6)$$

with z^* being the solution of the equivalent problem $z^* = D|z^*| + b$. By Lemma 2 z^* is unique. Now

$$\|z^*\| = \|b+D|z^*|\| \ge \|b\|-\|D|z^*|\| \ge \|b\|-\|D\|\|z^*\|$$

and therefore

$$\|z^*\| \ge \frac{\|b\|}{1+\|D\|}.$$

By letting $z^{(0)} = 0$, $z^{(1)} = b$. If $b = 0$ then $x = y = 0$ and the solution has been found. Otherwise from (6)

$$\|z^*-z^{(n)}\| \le \frac{\|D\|^n}{1-\|D\|}\|b\|.$$

Then

$$\|z^*\| - \|z^{(n)}\| \le \|z^* - z^{(n)}\| \le \frac{\|D\|^n}{1-\|D\|}\|b\|$$

or

$$\|z^{(n)}\| \ge \|z^*\| - \frac{\|D\|^n}{1-\|D\|}\|b\| \ge$$

$$\frac{\|b\|}{1+\|D\|} - \frac{\|D\|^n}{1-\|D\|}\|b\| = \left(\frac{1}{1+\|D\|} - \frac{\|D\|^n}{1-\|D\|}\right)\|b\|.$$

From

$$\|z^{(n)}\| \ge \left(\frac{1}{1+\|D\|} - \frac{\|D\|^n}{1-\|D\|}\right)\|b\|$$

there exists an i such that

$$\|z^{(n)}\|_\infty = |z_i^{(n)}| \ge c_1\left(\frac{1}{1+\|D\|} - \frac{\|D\|^n}{1-\|D\|}\right)\|b\|$$

where c_1 is a constant dependent on the norm $\|\cdot\|$ used. Also

$$|z_i^* - z_i^{(n)}| \le \|z^* - z^{(n)}\|_\infty \le c_2\left(\frac{\|D\|^n}{1-\|D\|}\right)\|b\|$$

with c_2 another constant dependent on the norm $\|\cdot\|$. From Lemma 4 it follows that $c_1 = 1/(\Sigma\Gamma_{ii}{}^p)^{1/p}$ and $c_2 = 1/(\min \Gamma_{ii})$. As soon as

$$c_1\left(\frac{1}{1+\|D\|} - \frac{\|D\|^n}{1-\|D\|}\right) > c_2\frac{\|D\|^n}{1-\|D\|}$$

it follows from the above that the component z_i^* must have the same sign as $z_i^{(n)}$. This occurs as soon as

$$\frac{1}{1+\|D\|} > \left(1 + \frac{c_2}{c_1}\right)\frac{\|D\|^n}{1-\|D\|}$$

or

$$n \geq N = \left\lceil \frac{\ell n \frac{1-\|D\|}{1+\|D\|} - \ell n\left(1 + \frac{c_2}{c_1}\right)}{\ell n \|D\|} \right\rceil$$

Hence after N iterations, the sign of at least one component of z^*, say z_i^*, is known. The index i is determined by

$$|z_i^{(n)}| = \max_k |z_k^{(n)}|.$$

From Theorem 1 it is now known whether x_i or y_i in the complementary feasible solution has to be zero.

In the case that $x_i = 0$, the i^{th} equation of (q,M) is deleted as well as the i^{th} column of M, and a similar problem of dimension m-1 with matrix $\tilde{M} \in R^{m-1 \times m-1}$ is left. Should $y_i = 0$, exchange x_i and y_i in (q,M) by a principal pivot transform [2] and reduce the problem in the same manner.

It was assumed that all principal pivot submatrices $\tilde{M}$ of M have the property that there exists a positive definite diagonal matrix Γ such that $\|\Gamma(I+\tilde{M})^{-1}(I-\tilde{M})\Gamma^{-1}\|_p < 1$ for some p, $1 \leq p \leq \infty$. Therefore the algorithm can be correctly applied to all possible reduced problems that arise.

The algorithm results in the signs of all components of z^* after running through at most m such cycles. In the case that $x_i = y_i = 0$ for some i, a reduced system is encountered sooner or later for which the $\tilde{b}$ vector is zero leading to $\tilde{z} = 0$, and hence the remaining components of x and y are zero.

Q.E.D.

If M is symmetric and positive definite, Theorem 3 reduces to a result of van Bokhoven [30], with $\Gamma = I$ and $p = 2$. For this case the constant $c_1 = 1/\sqrt{m}$ and the constant $c_2 = 1$.

The block algorithm is nearly identical to the modulus algorithm except that all elements of $\tilde{z}^{(N)}$ that equal or exceed the calculated test value are used to determine whether the corresponding x_i or y_i elements are positive. Therefore the number of cycles needed to solve (q,M) is significantly less than m.

Theorem 4.

With the hypotheses of Theorem 3, (q,M) is solvable by the block modulus algorithm.

Proof:

In the proof of Theorem 3 it was demonstrated that after N iterations there exists an i such that

$$|z_i^{(N)}| \geq c_1\left(\frac{1}{1+\|D\|} - \frac{\|D\|^n}{1-\|D\|}\right)\|b\|.$$

It is possible that more than one element of $z^{(N)}$ satifies this condition. Since any of these elements $z_i^{(N)}$ may be chosen as the indicator as to whether y_i or x_i in the complementary feasible solution has to be zero, and the unique solutions y and x will correctly be determined, it follows that all elements of $z^{(N)}$ that satisfy this

condition can be used to determine if the corresponding elements of y or x are zero, and the solution vectors y and x will still be correctly determined. Therefore it is possible to reduce the problem by more than one equation at a time.

Q.E.D.

5. IMPLEMENTATION DETAILS OF THE ALGORITHMS.

Four different algorithms to solve the linear complementarity problem were implemented and tested, and several more variations were considered but not fully explored. The first algorithm tested was a straightforward implementation of the modulus algorithm that was derived in the previous section. This algorithm reduces the size of the problem by one for each outer iteration. The algorithm therefore requires m outer iterations if $M \in R^{m \times m}$. The second algorithm is a variation of the first, differing in the determination for which element of z should be chosen as nonzero. The third algorithm, referred to as the block algorithm, is an enhanced version of the modulus algorithm. The problem is reduced as much as possible after each outer iteration. The minimum reduction is one dimension per outer iteration, so this algorithm requires at most m iterations, but in almost all observed cases the number of outer iterations was far less than m.

The fourth algorithm tested was a straightforward implementation of fixed point iteration. Techniques were considered to speed up fixed point iteration using the diagonal Aitken procedure or Steffensen's iteration [11,pp. 90-92]. The details of the four algorithms and the problems encountered with Aitken acceleration will now be discussed.

As shown in the preceding section the linear complementarity problem (q,M) can be transformed into the equivalent fixed point problem

$$z = D|z| + b,\ D \in R^{m\times m},\ b,\ z \in R^{m}.$$

The modulus algorithm as implemented works as follows. A loop based on the size of the problem, m, is set up. Each pass through this loop allows the size of the problem to be reduced by 1.

During each pass through the outer loop the transformation of the current problem

$$\tilde{y} = \tilde{M}\tilde{x} + \tilde{q},\ \tilde{x} \geq 0,\ \tilde{y} \geq 0,\ \tilde{y}^{t}\tilde{x} = 0$$

$$\tilde{M} \in R^{\tilde{m}\times\tilde{m}},\ \tilde{x},\ \tilde{y} \in R^{\tilde{m}},\ 1 \leq \tilde{m} \leq m$$

to

$$\tilde{z} = \tilde{D}|z| + \tilde{b}$$

$$\tilde{D} \in R^{\tilde{m}\times\tilde{m}},\ \tilde{z},\ \tilde{b} \in R^{\tilde{m}},\ 1 \leq \tilde{m} \leq m$$

is required.

First $\tilde{b}$ is determined using the LINPACK routines DGEFA and DGESL [4] to solve $-\tilde{q} = (I+\tilde{M})\tilde{b}$. If at some point $\tilde{b} = 0$ then it is known that $\tilde{x} = \tilde{y} = 0$, and the outer loop may be exited early.

Assuming $\tilde{b} \neq 0$, the spectral radius $\theta(\tilde{M})$ of $\tilde{D}$ is determined by finding the eigenvalues $\lambda_i(\tilde{M})$ of $\tilde{M}$ and letting

$$\theta(\tilde{M}) = \max |(1-\lambda_i(\tilde{M}))/(1+\lambda_i(\tilde{M}))|.$$

Next a value

$$\tilde{N} = \left\lceil \frac{\ell n \frac{1-\theta(\tilde{M})}{1+\theta(\tilde{M})} - \ell n (1+\sqrt{\tilde{m}})}{\ell n\, \theta(\tilde{M})} \right\rceil$$

is calculated. Letting $\tilde{z}^{(0)} = 0$, the iteration

$$\tilde{z}^{(n+1)} = \tilde{D}|\tilde{z}^{(n)}| + \tilde{b}$$

is performed $\tilde{N}$ times. A test value

$$\tilde{T} = \frac{\left(\frac{1}{1+\theta(\tilde{M})} - \frac{\theta(\tilde{M})^{\tilde{N}}}{1-\theta(\tilde{M})}\right) \|\tilde{b}\|_2}{\sqrt{\tilde{m}}}$$

can be calculated. If the original matrix M in (q,M) is symmetric and positive definite, then there exists an i such that

$$|z_i^{(\tilde{N})}| \geq \tilde{T}.$$

Furthermore the i^{th} component of the solution z^* (to the reduced problem) has the same sign as the i^{th} component of

$\tilde{z}^{(\tilde{N})}$. Therefore it is known whether the i^{th} component of $\tilde{y}$ or $\tilde{x}$ in the complementary feasible solution is zero. In the first modulus algorithm $\tilde{T}$ is not explicitly calculated. It is assumed that the element of $\tilde{z}^{(\tilde{N})}$ with the largest magnitude would exceed $\tilde{T}$. The index of this element is noted in a pivot vector. If the current iteration of the outer loop is the j^{th} iteration for example, the j^{th} element of the pivot vector is assigned the index of the element chosen. This algorithm will be referred to as the modulus algorithm (max). In the second modulus algorithm $\tilde{T}$ is calculated. The first element of $\tilde{z}^{(\tilde{N})}$ which equals or exceeds $\tilde{T}$ is used or, if no such element is found, the element with the largest magnitude is used. Again the index of the element is noted in the pivot vector. This second modulus algorithm will be referred to as the modulus algorithm (test).

If $\tilde{x}_i = 0$, the i^{th} equation in $\tilde{y} = \tilde{M}\tilde{x} + \tilde{q}$ is deleted as well as the i^{th} column of $\tilde{M}$. This is implemented by interchanging the i^{th} and $\tilde{m}^{th}$ rows and columns of a matrix, and the i^{th} and $\tilde{m}^{th}$ components of a vector. The new reduced problem is thus left in the first $\tilde{m} - 1$ rows and columns of a matrix, and the first $\tilde{m} - 1$ components of a vector. The corresponding elements of the pivot vector are also exchanged, so all row and column exchanges can later be deduced.

If $\tilde{y}_i = 0$, the variables $\tilde{x}_i$ and $\tilde{y}_i$ are exchanged in $\tilde{y} = \tilde{M}\tilde{x} + \tilde{q}$ and the problem is reduced in the same manner as above. One constant is added to the appropriate element of the pivot vector to signify that $\tilde{x}_i = 0$, or a different constant is added if $\tilde{y}_i = 0$. This concludes one iteration of the outer cycle.

After this outer loop terminates a modified system which can be represented by

$$M'x' = q', \; M' \in R^{m \times m}, \; x', \; q' \in R^m$$

is left. M' is lower triangular with minus ones on the diagonal, all of the elements of q' are known, and $x'_1 = -q'_1$. Therefore x'_{i+1} equals the inner product of the first i components of the $(i+1)^{st}$ row of M' with the first i components of x', minus q'_{i+1}, where $1 \leq i \leq m-1$. It is therefore straightforward to calculate x and y since it is known from the pivot vector which elements of x and y are represented by each element of x'. This concludes the modulus algorithm.

The modulus algorithm as implemented is guaranteed to work only when the original matrix M is symmetric and positive definite. This is because the 2-norm and $1/\sqrt{m}$ (the constant relating the 2-norm and the max norm) have been used to calculate the number of iterations of $\tilde{z}^{(n+1)} = \tilde{D}|\tilde{z}^{(n)}| + \tilde{b}$. As shown in the proof of Theorem 3, the

norm to be used and the constants c_1 and c_2 depend on the current problem. The appropriate norm and constants for nonsymmetric P-matrices cannot be easily determined, and therefore the 2-norm, $c_1 = 1/\sqrt{\tilde{m}}$, and $c_2 = 1$ have been arbitrarily used. As shown in the next section the use of these choices seems reasonable, since the algorithm has a good success rate on nonsymmetric problems.

If the calculated number of iterations, $\tilde{N}$, is not large enough it is possible that the algorithm as implemented will return one or more negative x_i or y_i values even though a feasible solution does exist. This may be due to the inapplicability of Theorem 3, the use of the 2-norm instead of an appropriate absolute norm, or roundoff error.

Theorem 3 also required that all principal pivot submatrices satisfy a certain condition. However, the modulus algorithm as implemented may work even if some of the principal pivot submatrices violate this condition, because such a submatrix might not actually be created during the execution of the algorithm.

The block algorithm as implemented is identical to the modulus algorithm (test) with the single exception that all elements of $z^{(\tilde{N})}$ are tested against $\tilde{T}$. The problem is reduced when the first element of $z^{(\tilde{N})}$ is found that equals or exceeds $\tilde{T}$, but instead of returning to the top of the

outer loop to begin again, the testing of elements of $z^{(\tilde{N})}$ continues until all possible reductions have been made. If no elements of $z^{(\tilde{N})}$ equal or exceed $\tilde{T}$ then the element with the largest magnitude is used as in the modulus algorithm (test).

This enhancement greatly reduces the overall time to solve the problem if more than one reduction can actually be made for each pass through the outer loop. As mentioned, it is almost always the case that the number of outer iterations is far less than m, at least in the observed cases. This is clear from the timing results presented in the next section.

The fourth algorithm implemented and tested was fixed point iteration. As with the preceeding algorithms the problem (q,M) is transformed to $z = D|z| + b$. The vector b is again determined by using LINPACK routines DGEFA and DGESL to solve $-q = (I+M)b$. However, this calculation is performed only once, as compared to once for each pass through the outer loop in the previous algorithms.

Next the fixed point iteration $z^{(n+1)} = D|z^{(n)}| + b$ starting with $z^{(0)} = 0$ is performed until convergence is detected. For the tests reported here, convergence was defined to occur when all elements of $z^{(n+1)}$ differed from the corresponding elements of $z^{(n)}$ by less than 10^{-12}.

Since it is possible for some of the elements of $z^{(n)}$ to oscillate or for convergence to be extremely slow, an arbitrary limit of 10,000 iterations was enforced.

Assuming convergence is detected, the x and y solution vectors are determined using

$$y = |z| - z \text{ and } x = |z| + z.$$

Another algorithm considered was the fixed point iteration (5) coupled with the diagonal Aitken procedure to accelerate convergence. This algorithm is also referred to as Steffensen's iteration as it was first proposed by Steffensen according to Henrici [11,p. 91]. This algorithm would solve the equivalent fixed point problem $z = D|z| + b$ as the previous algorithms do, again with $z^{(0)} = 0$.

The vector $z^{(n+1)}$ would be determined as follows: Two iterations are performed, say

$$z^{(n1)} = D|z^{(n)}| + b \text{ and } z^{(n2)} = D|z^{(n1)}| + b.$$

Each element of $z^{(n+1)}$ is determined using the formula

$$z_i^{(n+1)} = z_i^{(n)} - \frac{(z_i^{(n1)} - z_i^{(n)})^2}{(z_i^{(n2)} - 2z_i^{(n1)} + z_i^{(n)})}.$$

If the denominator is equal to zero, $z_i^{(n+1)}$ is set equal to $z_i^{(n)}$. The elements of $z^{(n+1)}$ are then compared with the elements of $z^{(n)}$ to determine if convergence has occured.

However, it is possible for there to be various combinations of positive and negative values in the set

$\{z_i^{(n)}, z_i^{(n1)}, z_i^{(n2)}\}$. This leads to the possibility that some elements of z diverge rather than converge. This was frequently the case during the testing of this algorithm. Various modifications involving testing the signs of $z_i^{(n)}$, $z_i^{(n1)}$, and $z_i^{(n2)}$, and handling each case differently were tried, but none proved successful. It is possible that a method to accelerate fixed point iteration using methods along the lines of Mangasarian [20] can be found, however.

6. EXPERIMENTAL RESULTS.

The four algorithms discussed in the previous section were coded in Fortran 77, and were compiled and run on a VAX-11/780 computer. All real variables, vectors and matrices were declared to be double precision. Execution times for the different algorithms were measured by accessing system tables. This section will discuss the timing data and success rates observed during the testing of these algorithms.

Two generating algorithms were used to create, in a somewhat random fashion, the problems that were used to test the algorithms.

The first algorithm generated P-matrices in such a manner that the elements could be positive or negative with a slightly higher probability that any given element was

positive. The P-matrices generally did not have the property that all principal pivot submatrices had eigenvalues with positive real parts. However, each matrix generated was tested to insure that the original matrix had all of its eigenvalues in the complex right half-plane. (This is tantamount to $\rho(D) < 1$.) Any that did not were rejected. The absolute values of the eigenvalues generated by this method were found to be fairly randomly distributed between 0 and 10 with a scattering of values larger than 10. The maximum eigenvalues in absolute value for each matrix fell fairly randomly in the range 10 to 50, with a few observed above or below this range. This generating system will be referred to as Method 1.

The second algorithm generated 2×2 block diagonal matrices in such a manner that the principal 2×2 submatrices (and therefore each principal submatrix) were P-matrices with their eigenvalues in the complex right half-plane. The absolute values of the eigenvalues for these matrices were exponentially distributed from 0 to 142. The maximum eigenvalue in absolute value for each matrix fell in the approximate range 15 to 140 with the majority in the range 30 to 50. This generating system will be referred to as Method 2.

Both generators created q vectors with an even probability that any given element would be positive or negative, and each component of q uniformly distributed in the interval [-10.0,10.0].

As discussed earlier the modulus algorithm (max) did not calculate the test value $\tilde{T}$. Instead the element of $z^{(N)}$ with the largest magnitude was used to indicate which element of $\tilde{x}$ or $\tilde{y}$ was zero. The modulus algorithm (test) actually did calculate each $\tilde{T}$. If none of the elements of $z^{(\tilde{N})}$ equaled or exceeded $\tilde{T}$, the element with the largest magnitude was used. Tables 1 and 2 show that the modulus algorithm (max) is about equal to the modulus algorithm (test) in terms of execution time.

For both of the generators the following was done: One hundred problems of size 5×5, 10×10, 15×15 and 20×20 each were generated. Each of these 400 problems was passed to each of the four algorithms for solution. Table 1 summarizes the results of the testing using problems generated by Method 1. For each size the decimal number represents the average time in seconds for each problem to be solved. Only those problems successfully solved by all four algorithms are included in the average. The number in parentheses following this decimal number is the number of problems out of 100 that the particular algorithm did not

successfully solve. The last row shows the number of problems out of 100 that all four methods correctly solved. Table 2 shows the results for the same number of problems generated using Method 2.

A second batch of tests was run with enough problems generated so that each of the four algorithms tested ran successfully on 100 problems of each size. Table 3 shows the average time in seconds for 100 problems generated by Method 1 of each size that each algorithm solved successfully. Table 4 shows the results when 100 successful solutions of each size problem generated by Method 2 are compared.

An interesting question is the significance of the number used for $\tilde{\theta}(M)$. As an experiment, the incorrect formula

$$\theta'(\tilde{M}) = |(1-\max|\lambda_i(\tilde{M})|)/(1+\max|\lambda_i(\tilde{M})|)|$$

was used instead of the correct formula for $\tilde{\theta}(M)$ shown in the previous section. Clearly $\theta'(\tilde{M}) \leq \tilde{\theta}(M)$, so the number of iterations $\tilde{N}'$ calculated using $\theta'(\tilde{M})$ is less than or equal to the number of iterations $\tilde{N}$ calculated using the corrrect value. It was found that the modulus (test), modulus (max), and block algorithms all ran faster when $\theta'(\tilde{M})$ was used, and that the success rates were only slightly reduced when compared with the results from the correct versions of these algorithms.

TABLE 1

Timing and Failure Rates- Generating Method 1

size	5	10	15	20
modulus(test)	.59(15)	4.93(14)	14.39(19)	23.15(21)
modulus(max)	.72(16)	6.03(20)	14.51(19)	20.92(21)
block	.39(11)	1.92(5)	4.41(5)	6.90(8)
fixed point	.36(4)	1.80(4)	2.97(6)	3.94(7)
sample size	77	71	70	69

TABLE 2

Timing and Failure Rates- Generating Method 2

size	5	10	15	20
modulus(test)	.59(0)	4.62(0)	14.45(0)	37.41(0)
modulus(max)	.58(0)	4.56(0)	14.17(0)	37.52(0)
block	.38(0)	2.55(0)	7.14(0)	18.08(0)
fixed point	1.27(0)	5.03(1)	10.66(0)	20.19(1)
sample size	100	99	100	99

TABLE 3

Timing - Generating Method 1

size	5	10	15	20
modulus(test)	.96	3.16	11.17	27.00
modulus(max)	1.01	2.72	9.54	16.20
block	.61	1.25	2.58	4.13
fixed point	.55	.96	1.99	3.44

TABLE 4

Timing - Generating Method 2

size	5	10	15	20
modulus(test)	.68	4.82	14.35	37.21
modulus(max)	.64	4.67	14.10	36.57
block	.41	2.55	6.93	16.17
fixed point	1.35	4.99	9.87	19.48

Table 5 shows the timing and failure rates for the modified versions of these algorithms. The problems used were exactly the same 400 problems used to produce the results shown in Table 1. Table 6 corresponds to Table 2 in a similar manner. Timing data obtained on a time-shared system always contains some variability, which explains the discrepancy between the fixed point iteration times in Tables 2 and 6.

It is reasonable to assume that the modulus (test) algorithm and the modulus (max) algorithm would have similar success rates. This was found to be the case, with the single exception of the problems of size 10 generated by Method 1 (Table 1). Since there seems to be no logical explanation for the modulus (max) algorithm to have a higher failure rate than the modulus (test) algorithm, it can only be assumed that the difference in the failure rate is a random fluctuation, and would diminish with a much larger sample size.

The block algorithm is clearly superior to either modulus algorithm, both in terms of execution time and success rate. The block algorithm is considerably faster because the problem can frequently be reduced by more than one equation for each pass through the outer loop. By reducing the problem by more than one equation at a time, there are fewer

TABLE 5

Results using $\tilde{\theta}'(M)$- Generating Method 1

size	5	10	15	20
modulus(test)	.47(15)	3.26(18)	10.64(19)	16.63(22)
modulus(max)	.54(15)	3.07(19)	9.14(18)	13.57(21)
block	.29(10)	.90(7)	2.61(5)	3.08(8)
fixed point	.35(4)	1.80(4)	2.90(6)	3.39(7)
sample size	78	71	71	68

TABLE 6

Results using $\tilde{\theta}'(M)$- Generating Method 2

size	5	10	15	20
modulus(test)	.53(2)	3.89(0)	11.87(0)	31.17(0)
modulus(max)	.52(2)	3.84(0)	11.61(0)	30.74(0)
block	.34(2)	2.16(0)	5.75(0)	14.87(0)
fixed point	1.24(0)	5.01(1)	10.50(0)	20.12(1)
sample size	98	99	100	99

subproblems to solve, and therefore the chances of encountering a principal pivot submatrix with eigenvalues in the complex left half-plane are reduced. This could account for the lower failure rate for the block algorithm that is apparent in Table 1.

For problems generated by Method 1, the fixed point iteration algorithm is faster than the block algorithm, and has a better rate of success (Tables 1 and 3). However, for problems generated by Method 2, the block algorithm is faster than the fixed point iteration algorithm (Tables 2 and 4). This suggests that for problems with matrices with fairly small eigenvalues (e.g., those created by Method 1), the overhead of calculating the number of iterations $\tilde{N}$ and the test value $\tilde{T}$ does not offset the gain from having to perform fixed point iteration a fewer number of times to determine a solution. For problems with matrices with larger eigenvalues (e.g., those created by Method 2), the overhead of calculating these values does pay for itself, however.

With generating Method 2, $\rho(D) < 1$ is guaranteed. However the fixed point iteration algorithm failed to solve two problems generated by Method 2 (Table 2). It was found that neither of these two problems was solved when the number of iterations allowed was increased to 100,000. It

is possible that for these two problems, the rate of convergence is extremely slow. However, it is also possible that, even though $\rho(D) < 1$ for both of these problems, $\|D\| \geq 1$ for all absolute norms, and convergence could therefore not be guaranteed.

In comparing the data in Tables 1 and 2 with the data in Tables 5 and 6, it seems that the algorithms using the incorrect value $\theta'(\tilde{M})$ are faster than the corresponding algorithms which use the correct value $\theta(\tilde{M})$. The failure rates are, on average, only slightly better when using the correct value. This suggests that the use of the 2-norm and setting $c_1 = 1/\sqrt{\tilde{m}}$ are reasonable choices, or even in some way "conservative" choices. It seems that the number of iterations $\tilde{N}$ calculated using these values is perhaps larger than necessary.

7. CONCLUSIONS.

The number of iterations $\tilde{N}$ calculated for each subproblem during the execution of the modulus and block algorithms is somewhat arbitrary, as shown by the testing involving $\theta'(\tilde{M})$. The tradeoff between $\tilde{N}$ and the failure rate of the block algorithm warrants further study.

Of the algorithms tested here, the fixed point algorithm seems to be the best. Depending on the nature of the

problem, the block algorithm can be faster. However, the fixed point iteration algorithm has a better success rate. The success rate could possibly be even better if the arbitrary limit on the number of iterations were increased. Further research into a method to accelerate the fixed point algorithm needs to be done. An accelerated fixed point algorithm would clearly be better than the block algorithm.

REFERENCES

1. Y. Bard, Nonlinear Parameter Estimation, (Academic Press, New York, 1974).

2. R. W. Cottle and G. B. Danzig, "Complementary Pivot Theory of Mathematical Programming," Linear Algebra and its Applications 1 (1968) 103-125.

3. C. Cryer, "The Solution of a Quadratic Programming Problem using Systematic Overrelaxation," SIAM Journal of Control 9 (1971) 385-392.

4. J. J. Dongerra, C. B. Moler, J. R. Bunch, G. W. Stewart, LINPACK User's Guide, (SIAM 1979).

5. B. C. Eaves, "The Linear Complementarity Problem in Mathematical Programming," Management Science 17 (1971) 612-634.

6. B. C. Eaves, "Homotopies for the Computation of Fixed Points," Mathematical Programming 3 (1972) 1-22.

7. B. C. Eaves and R. Saigal, "Homotopies for Computation of Fixed Points on Unbounded Regions," Mathematical Programming 3 (1972) 225-237.

8. M. Fiedler and V. Ptak, "On Matrices with Non-positive Off-diagonal Elements and Positive Principal Minors," Czechoslovak Mathematical Journal 12 (1962) 382-400.

9. P. Gale, The Theory of Linear Economic Models, (McGraw Hill Book Co., New York, 1960).

10. T. Hansen and H. Scarf, "On the Applications of a Recent Combinatorial Algorithm," Cowles Commission Discussion Paper No. 272, Yale University, 1969.

11. P. Henrici, Elements of Numerical Analysis, (John Wiley & Sons, Inc., New York, 1964).

12. A. S. Householder, Lectures on Numerical Algebra, (Mathematical Association of America, 1972).

13. A. W. Ingleton, "A Problem in Linear Inequalities," Proceedings of the London Mathematical Society 3rd Series 16 (1966) 519-536.

14. S. Karamardian, "The Complementarity Problem," Mathematical Programming 2 (1972) 107-129.

15. L. M. Kelly and L. T. Watson, "Q-Matrices and Spherical Geometry," Linear Geometry and its Applications 25 (1979) 175-189.

16. M. M. Kostreva, "Elasto-Hydrodynamic Lubrication: A Nonlinear Complementarity Problem," Tech. Report GMR-4113, Mathematics Dept., GM Research Laboratories, Warren, MI (1982).

17. C. E. Lemke, "Recent Results on Complementarity Problems," in : Nonlinear Programming, Eds. J. B. Rosen, O. L. Mangasarian and K. Ritter, (Academic Press, New York, 1970) pp. 349-384.

18. C. E. Lemke, "Bimatrix Equilibrium Points and Mathematical Programming," Management Science 11 (1965) 681-689.

19. C. E. Lemke and J. T. Howson, "Equilibrium Points of Bimatrix Games," SIAM Journal of Applied Mathematics 12 (1964) 413-423.

20. O. L. Mangasarian, "Equivalence of the Complimentarity Problem to a System of Nonlinear Equations," SIAM Journal of Applied Mathematics 31 (1976) 89-92.

21. O. L. Mangasarian, "Linear Complementarity Problems Solvable by a Single Linear Program," Mathematical Programming 10 (1976) 263-270.

22. O. H. Merrill, "Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Non-Empty, Convex, Upper Semi-Continuous Point to Set Mappings," Dept. of Industrial Engineering, Univ. of Michigan Technical Report No. 71-7, 1971.

23. K. G. Murty, "On the Number of Solutions to the Complementarity Problem and Spanning Properties of Complemenetarity Cones," Linear Algebra and its Applications 5 (1972) 65-108.

24. K. G. Murty, "Note on a Bard-Type Scheme for Solving the Complementarity Problem," Opsearch 11 (1974) 123-130.

25. K. G. Murty, Linear and Combinatorial Programming, (John Wiley & Sons, New York, 1976).

26. K. G. Murty, "Computational Complexity of Complementary Pivot Methods," Mathematical Programming Study 7 (1978) 61-73.

27. R. Saigal, "On the Convergence Rate of Algorithms for Solving Equations that are Based on Methods of Complementary Pivoting," Mathematics of Operations Research 2,2 (1977) 108-124.

28. H. Samelson, R. M. Thrall and O. Wesler, "A Partitioning Theorem for Euclidean N-Space," Proceedings of the American Mathematical Society 9 (1958) 805-807.

29. H. Scarf, The Computation of Economic Equilibria, (Yale University Press, New Haven, CT, 1973).

30. W. M. G. van Bokhoven, "A Class of Linear Complementarity Problems is Solvable in Polynomial Time," unpublished paper, Dept. of Electrical Engineering, University of Technology, The Netherlands, 1980.

31. L. T. Watson, "A Variational Approach to the Linear Complementarity Problem," Doctoral Dissertation, Dept. of Mathematics, University of Michigan, Ann Arbor, MI, 1974.

32. L. T. Watson, "An Algorithm for the Linear Complementarity Problem," International Journal of Computer Mathematics 6 (1978) 319-325.

33. L. T. Watson, "Solving the Nonlinear Complementarity Problem by a Homotopy Method," SIAM Journal on Control and Optimization 17,1 (1979) 36-46.