Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 9, No. 3, pp. 479–491, July 2013

VALIDATED SOLUTIONS FOR P-MATRIX LINEAR COMPLEMENTARITY PROBLEMS

Takuma Kimura

Abstract: We propose an algorithm for finding a validated solution of the P-matrix linear complementarity problem. The algorithm is based on the error bound proposed by Chen and Xiang [2] and an iteration scheme. The effect of rounding errors has been taken into account to perform the algorithm. Numerical results show that the proposed algorithm is efficient.

Key words: *linear complementarity problems, numerical verification*

Mathematics Subject Classification: *90C33, 65G20, 65G50*

1 Introduction

The linear complementarity problem (LCP),

$$Mx + q \geq 0, \quad x \geq 0, \quad x^T(Mx + q) = 0,$$

can be reformulated as the following system of piecewise linear equations,

$$\min(x,\, Mx + q) = 0, \tag{1.1}$$

where $M \in \mathbb{R}^{n\times n}$, $q \in \mathbb{R}^n$.

LCP has many applications in engineering and economics. And many numerical methods for solving the LCP have been proposed. Therefore, it is very important to verify the accuracy of the solutions obtained by numerical methods.

It is known that (1.1) has a unique solution for any vector q if and only if M is a P-matrix, i.e. M satisfies the condition, $\max\limits_{1\leq i\leq n} v_i(Mv)_i > 0$, for all $v \neq 0$ (Theorem 3.3.7 in [3]). We assume that M is a P-matrix.

Let x^* be an exact solution and x be an approximate solution of (1.1). We define the natural residual function,

$$r(x) = \min(x, Mx + q).$$

In [2], Chen and Xiang proposed an error bound in $\|\cdot\|_p$ ($p \geq 1$ or $p = \infty$) norms. Let $y = Mx + q$ and $y^* = Mx^* + q$. For any $x, x^* \in \mathbb{R}^n$, we have

$$r(x) = (I - D + DM)(x - x^*), \tag{1.2}$$

ISSN 1348-9151

where D is a diagonal matrix whose diagonal elements are d_i,

$$d_i = \begin{cases} 0, & y_i \geq x_i,\ y_i^* \geq x_i^*, \\ 1, & y_i \leq x_i,\ y_i^* \leq x_i^*, \\ \dfrac{\min(x_i, y_i) - \min(x_i^*, y_i^*) - x_i + x_i^*}{y_i - y_i^* - x_i + x_i^*}, & \text{otherwise.} \end{cases} \tag{1.3}$$

Moreover, we have $d_i \in [0, 1]$. It is known that M is a P-matrix if and only if $I - D + DM$ is nonsingular for any diagonal matrix $D = \text{diag}(d)$ with $d \in [0, 1]^n$ [3]. Consequently, Chen and Xiang proposed,

$$\|x - x^*\|_p \leq \beta_p(M)\|r(x)\|_p, \tag{1.4}$$

where

$$\beta_p(M) = \max_{d\in[0,1]^n} \|(I - D + DM)^{-1}\|_p.$$

For some special matrices, we can compute $\beta_p(M)$ easily, for example, if M is an M-matrix then $\beta_p(M) = \|M^{-1}\|_p$. Moreover, Chen and Xiang proved that (1.4) is sharper than the Mathias-Pang error bound [6] when $p = \infty$. However, for general P-matrix, we do not have simple upper bound for $\beta_p(M)$.

In this paper, we propose an algorithm to compute lower and upper bounds for the exact solution of LCP where M is a P-matrix. Specifically, we compute two vectors $\underline{x}^*$ and $\overline{x}^*$ such that

$$\underline{x}^* \leq x^* \leq \overline{x}^*,$$

where x^* is the exact solution of (1.1). This method is based on the equation (1.2), interval arithmetic and rounding mode controlled computations [5, 11].

In section 2, we consider interval arithmetics and rounding mode controlled computations. In section 3, we discuss verification method and iteration scheme. Some numerical results are discussed in section 4.

2 Interval Arithmetics and Rounding Mode Controlled Computations

In this paper, we use the interval arithmetic to control computational errors arising from each calculation.

Let $\mathbb{IR} = \{[a, b] \mid a \leq b,\ a, b \in \mathbb{R}\}$ denote the set of intervals. The set of n-dimensional interval vectors and the set of $n \times n$ interval matrices are denoted by $\mathbb{IR}^n$ and $\mathbb{IR}^{n\times n}$. In this paper, we denote boldface for intervals in which the lower and upper bounds are denoted by underscore and overscore respectively. For example, an interval matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ has interval elements,

$$\mathbf{A}_{ij} \equiv [\underline{A}_{ij}, \overline{A}_{ij}] = \{A_{ij} \in \mathbb{R} \mid \underline{A}_{ij} \leq A_{ij} \leq \overline{A}_{ij}\} \in \mathbb{IR}, \quad i, j = 1, \cdots, n.$$

The mid point and the radius of an interval $\mathbf{a}$ can be defined as

$$\mathsf{mid}(\mathbf{a}) = (\overline{a} + \underline{a})/2, \quad \mathsf{rad}(\mathbf{a}) = (\overline{a} - \underline{a})/2.$$

The range of minimum and maximum of two intervals $\mathbf{a}$ and $\mathbf{b}$ are defined as

$$\min(\mathbf{a},\ \mathbf{b}) = \{\min(a,\ b) \mid a \in \mathbf{a},\ b \in \mathbf{b}\},$$
$$\max(\mathbf{a},\ \mathbf{b}) = \{\max(a,\ b) \mid a \in \mathbf{a},\ b \in \mathbf{b}\}.$$

An interval matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ is called nonsingular if any matrices included in $\mathbf{A}$ are nonsingular. We denote the solution set of an interval linear systems $\mathbf{Ax} = \mathbf{b}$ by,

$$S(\mathbf{A}, \mathbf{b}) := \{x \mid Ax = b,\ A \in \mathbf{A},\ b \in \mathbf{b}\}.$$

If $\mathbf{A} \in \mathbb{IR}^{n\times n}$ is nonsingular and $\mathbf{b} \in \mathbb{IR}^n$ is bounded, then the interval hull of $S(\mathbf{A}, \mathbf{b})$ exists and is defined as

$$\Sigma(\mathbf{A}, \mathbf{b}) = [\inf S(\mathbf{A}, \mathbf{b}),\ \sup S(\mathbf{A}, \mathbf{b})] \in \mathbb{IR}^n,$$

where "inf" and "sup" are taken componentwise. In this paper, $\mathbf{x} = \Sigma(\mathbf{A}, \mathbf{b})$ is called the solution of $\mathbf{Ax} = \mathbf{b}$.

To compute error bounds, we should take the roundoff errors into account. By utilizing the rounding mode controlled computations, we can compute the interval enclosing the exact result of floating point arithmetic.

In the following, for each calculation, we consider the intervals enclosing the exact result. For example, considering the round off error, we cannot compute exact value of $r(x)$, but we can set $\mathbf{r}(x)$ be an interval vector computed by rounding mode controlled computations such that $r(x) \in \mathbf{r}(x)$.

For more details of interval arithmetic, interval linear system and rounding mode controlled computations, see, e.g., [5, 10, 11, 7].

Here, we consider a numerical verification method based on (1.2).

When we have two vectors $\underline{d}$ and $\overline{d}$ such that $\underline{d} \le d \le \overline{d}$, we can compute two matrices $\underline{K}$ and $\overline{K}$ such that

$$\underline{K} \le I - D + DM \le \overline{K},$$

by using the interval computations.

We can set

$$\mathbf{K} = I + \mathbf{D}(M - I). \tag{2.1}$$

It is easy to show that

$$I + \mathbf{D}(M - I) \subseteq I - \mathbf{D} + \mathbf{D}M,$$

and Since $d_i \in [0, 1]$, we can set $\underline{d}_i = 0$ and $\overline{d}_i = 1$.

If $\mathbf{K}$ is nonsingular and we can find a solution $\mathbf{e}^0 = \Sigma(\mathbf{K}, \mathbf{r}(x))$ then, $\mathbf{e}^0$ is an error bound such that

$$x - x^* \in \mathbf{e}^0.$$

Considering a special case of the verification method for NCP proposed by Alefeld and Wang [1], above mentioned error bound also can be obtained. In [1], the authors remarked that MATLAB functions "intervalhull.m" developed by Rohn and "veryfylss.m" by Rump can compute $\mathbf{e}^0$.

3 Iteration Scheme for Verification

In this section, we consider an algorithm to compute validated solutions for P-matrix LCP.

We assume that an error bound $\mathbf{e}^0$ is given. $\mathbf{e}^0$ satisfies

$$x^* \in x - \mathbf{e}^0, \quad y^* \in \mathbf{y} - M\mathbf{e}^0,$$

where $\mathbf{y}$ is an interval vector such that $\underline{y} \le Mx + q \le \overline{y}$. From $x^* \ge 0$ and $y^* \ge 0$, we have

$$x^* \in \max(0,\, x - \mathbf{e}^0), \quad y^* \in \max(0,\, \mathbf{y} - M\mathbf{e}^0).$$

3.1 Initial Error Bound

Let $\mathbf{d}^0 = [0, 1]^n$, $\mathbf{D}^0 = \text{diag}(\mathbf{d}^0)$ and $\mathbf{K}^0 = I + \mathbf{D}^0(M - I)$. From (1.2), we have

$$r(x) \in \mathbf{K}^0(x - x^*).$$

If we can find a solution,

$$\mathbf{e}^0 = \Sigma(\mathbf{K}^0, \mathbf{r}(x)), \tag{3.1}$$

then we have an initial error bound $\mathbf{e}^0$. If we cannot solve (3.1), we can employ other methods. For example, Cottle, Pang and Stone (Exercises 5.11.20 in [3]) introduced

$$\|x - x^*\|_2 \leq \frac{1 + \|M\|_2}{\lambda_1(\tilde{M})} \|r(x)\|_2 \leq \overline{e}^0, \tag{3.2}$$

where M is a positive definite matrix, $\tilde{M} = (M + M^T)/2$, and $\lambda_1(\tilde{M})$ is the smallest eigenvalue of $\tilde{M}$. (3.2) implies that

$$-\overline{e}^0 \leq x - x^* \leq \overline{e}^0,$$

then we can set $\mathbf{e}^0 = [\underline{e}^0, \overline{e}^0]$, where $\underline{e}^0 = -\overline{e}^0$.

3.2 Reduced Problem

Now, we can check the relations between $x, \mathbf{y}, \mathbf{x}^*, \mathbf{y}^*$. If we find an index i satisfies $\overline{x}_i^* < \underline{y}_i^*$ or $\underline{x}_i^* > \overline{y}_i^*$, we can remove the variable x_i and consider a smaller problem.

Let a (universal) index set $\mathcal{N} = \{1, \cdots, n\}$, support index sets $\mathcal{I} = \{i \mid \overline{x}_i^* \not< \underline{y}_i^*\}$ and $\mathcal{J} = \{i \mid \underline{x}_i^* \not> \overline{y}_i^*\}$. Let $M_{\mathcal{I}\mathcal{J}}$ denote the submatrix of M with rows indexed by $\mathcal{I}$, columns by $\mathcal{J}$. Note that if M is a P-matrix then $M_{\mathcal{I}\mathcal{I}}$ is a P-matrix. The subvector of x and q indexed by $\mathcal{I}$ are denoted by $x_{\mathcal{I}}$ and $q_{\mathcal{I}}$.

Here, we consider the P-matrix LCP (1.1) with $\mathcal{I}^c \neq \emptyset$.

Lemma 3.1. *If $\mathcal{I}^c \neq \emptyset$, then* (1.1) *can be rewritten as P-matrix LCP,*

$$\begin{cases} \min(x_{\mathcal{I}},\, M_{\mathcal{I}\mathcal{I}}x_{\mathcal{I}} + q_{\mathcal{I}}) = 0, & (3.3a) \\ x_{\mathcal{I}^c} = 0. & (3.3b) \end{cases}$$

Proof. Clearly we have $x_i^* = 0$ for $i \in \mathcal{I}^c$. Therefore, (1.1) can be reformulated as following system,

$$\begin{cases} \min(x_{\mathcal{I}},\, M_{\mathcal{I}\mathcal{I}}x_{\mathcal{I}} + q_{\mathcal{I}}) = 0, & (3.4a) \\ M_{\mathcal{I}^c\mathcal{I}}x_{\mathcal{I}} + q_{\mathcal{I}^c} \geq 0, & (3.4b) \\ x_{\mathcal{I}^c} = 0, & (3.4c) \end{cases}$$

Since $M_{\mathcal{I}\mathcal{I}}$ is a P-matrix, (3.4a) has a unique solution $x_{\mathcal{I}}$. Therefore, we can remove the constraint (3.4b). □

Considering the computational cost, the numerical verification for the reduced problem (3.3) is more easily performed than (1.1).

Moreover, if $\mathcal{I}^c \cup \mathcal{J}^c = \mathcal{N}$, then (1.1) can be rewritten as linear systems. Clearly we have $(Mx^* + q)_i = 0$ for $i \in \mathcal{J}^c$. $\mathcal{I}^c \cup \mathcal{J}^c = \mathcal{N}$ implies that $\mathcal{I} \subseteq \mathcal{J}^c$. From Lemma 3.1 and above, we obtain following result.

Corollary 3.2. *If* $\mathcal{I}^c \cup \mathcal{J}^c = \mathcal{N}$ *then* (1.1) *is equivalent to the following linear system,*

$$\begin{cases} M_{\mathcal{I}\mathcal{I}} x_{\mathcal{I}} + q_{\mathcal{I}} = 0, & (3.5a) \\ x_{\mathcal{I}^c} = 0. & (3.5b) \end{cases}$$

There is much discussion on validated solutions of the linear systems (e.g. [5, 8, 7]). In this case, we will easily obtain the solution of (3.5) by using validated solvers for linear systems.

3.3 Updated Error Bounds

Considering the detail of (1.3), we have

$$d_i = \begin{cases} 0, & y_i \geq x_i,\ y_i^* \geq x_i^*, \\ 1, & y_i \leq x_i,\ y_i^* \leq x_i^*, \\ \dfrac{x_i^*}{y_i - x_i + x_i^*}, & y_i \geq x_i,\ y_i^* \leq x_i^*, \\ \dfrac{y_i - x_i}{y_i - y_i^* - x_i}, & y_i \leq x_i,\ y_i^* \geq x_i^*. \end{cases}$$

Now, we can check the relations between $x, \mathbf{y}, \mathbf{x}^*, \mathbf{y}^*$. Therefore, we can set

$$\mathbf{d}_i = \begin{cases} 0, & \underline{y}_i \geq x_i,\ \underline{y}_i^* \geq \overline{x}_i^*, \\ 1, & \overline{y}_i \leq x_i,\ \overline{y}_i^* \leq \underline{x}_i^*, \\ \dfrac{\mathbf{x}_i^*}{\mathbf{y}_i - x_i + \mathbf{x}_i^*}, & \underline{y}_i \geq x_i,\ \overline{y}_i^* \leq \underline{x}_i^*, \\ \dfrac{\mathbf{y}_i - x_i}{\mathbf{y}_i - \mathbf{y}_i^* - x_i}, & \overline{y}_i \leq x_i,\ \underline{y}_i^* \geq \overline{x}_i^*, \\ \dfrac{\mathbf{y}_i - x_i}{\mathbf{y}_i - \mathbf{y}_i^* - x_i} \cup 1, & \overline{y}_i \leq x_i,\ 0 \notin \mathbf{y}_i - \mathbf{y}_i^* - x_i, \\ \dfrac{\mathbf{x}_i^*}{\mathbf{y}_i - x_i + \mathbf{x}_i^*} \cup 0, & \underline{y}_i \geq x_i,\ 0 \notin \mathbf{y}_i - x_i + \mathbf{x}_i^*, \\ \dfrac{\mathbf{x}_i^*}{\mathbf{y}_i - x_i + \mathbf{x}_i^*} \cup 1, & \overline{y}_i^* \leq \underline{x}_i^*,\ 0 \notin \mathbf{y}_i - x_i + \mathbf{x}_i^*, \\ \dfrac{\mathbf{y}_i - x_i}{\mathbf{y}_i - \mathbf{y}_i^* - x_i} \cup 0, & \underline{y}_i^* \geq \overline{x}_i^*,\ 0 \notin \mathbf{y}_i - \mathbf{y}_i^* - x_i, \\ [0,\ 1], & \text{otherwise}, \end{cases} \tag{3.6}$$

such that $d \in \mathbf{d}$. Then we can set new $\mathbf{K}$ by (2.1) and compute new error bound $\mathbf{e}$ by solving $\mathbf{K}\mathbf{e} = \mathbf{r}(x)$.

3.4 Iterative Method

Since $\mathbf{d}^0$ and new $\mathbf{d}$ satisfies that

$$d \in \mathbf{d}^0, \quad d \in \mathbf{d},$$

we have

$$d \in \mathbf{d}^1 \equiv \mathbf{d} \cap \mathbf{d}^0.$$

Moreover, we have

$$x - x^* \in \mathbf{e}^1 \equiv \mathbf{e} \cap \mathbf{e}^0.$$

Therefore, we can compute interval vectors $\mathbf{d}^k$ and $\mathbf{e}^k$ such that

$$d \in \mathbf{d}^k \subseteqq \mathbf{d}^{k-1} \subseteqq \cdots \subseteqq \mathbf{d}^1 \subseteqq \mathbf{d}^0 = [0,1]^n,$$

$$x - x^* \in \mathbf{e}^k \subseteqq \mathbf{e}^{k-1} \subseteqq \cdots \subseteqq \mathbf{e}^1 \subseteqq \mathbf{e}^0.$$

Based on the above-mentioned, we propose the following algorithm to verify the solutions of P-matrix LCP.

Algorithm 1.

Step 0

Input M, q, approximate solution x and $maxiter$.

Compute an initial error bound $\mathbf{e}^0$.

Set $\mathbf{d}^0 = [0,1]^n$, $\mathbf{y} = Mx + q$, $\mathbf{x}^* = \max(0,\, x - \mathbf{e}^0)$, $\mathbf{y}^* = \max(0,\, \mathbf{y} - M\mathbf{e}^0)$, $\mathbf{r} = \min(x, Mx + q)$, $\mathcal{N} = \{1, \cdots, n\}$, $\mathcal{I}_0 = \{i \mid \overline{x}_i^* \not< \underline{y}_i^*\}$, $\mathcal{J}_0 = \{i \mid \underline{x}_i^* \not> \overline{y}_i^*\}$ and an iteration counter $k = 0$.

If ${\mathcal{I}_0}^c \neq \emptyset$ then goto Step 1, else Step 2.

Step 1

If ${\mathcal{I}_k}^c \cup {\mathcal{J}_k}^c = \mathcal{N}$ then return $\mathbf{x}^*$ by solving (3.5).

Set $\mathbf{x}_i^* = 0$, $\mathbf{e}_i^k = x_i$ and $\mathbf{d}_i^k = 0$ for $i \in {\mathcal{I}_k}^c$. Compute $\mathbf{y}^* = \mathbf{y}^* \cap (\mathbf{y} - M\mathbf{e}^k)$ and then goto Step 2.

Step 2

If $k = maxiter$, return $\mathbf{x}^*$ and warning message.

Set $k = k + 1$.

Set $\mathbf{d}$ by (3.6).

Set $\mathbf{d}^k = \mathbf{d} \cap \mathbf{d}^{k-1}$. If $\mathbf{d}^k = \mathbf{d}^{k-1}$, then return $\mathbf{x}^*$.

Set $\mathbf{K} = I + \mathrm{diag}(\mathbf{d}^k)(M - I)$.

Solve $\mathbf{Ke} = \mathbf{r}$. If failed to solve it, return $\mathbf{x}^*$.

Set $\mathbf{e}^k = \mathbf{e} \cap \mathbf{e}^{k-1}$. If $\mathbf{e}^k = \mathbf{e}^{k-1}$, then return $\mathbf{x}^*$.

Set $\mathbf{x}^* = \mathbf{x}^* \cap (x - \mathbf{e}^k)$, $\mathbf{y}^* = \mathbf{y}^* \cap (\mathbf{y} - M\mathbf{e}^k)$.

Set $\mathcal{I}_k = \{i \mid \overline{x}_i^* \not< \underline{y}_i^*\}$ and $\mathcal{J}_k = \{i \mid \underline{x}_i^* \not> \overline{y}_i^*\}$.

If $\mathcal{I}_k \neq \mathcal{I}_{k-1}$ or $\mathcal{J}_k \neq \mathcal{J}_{k-1}$ then goto Step 1, else repeat Step 2.

The computational cost of Algorithm 1 is less than,

$$\mathcal{O}_1 + \left(\mathcal{O}_2 + \mathcal{O}(n^2)\right) \times maxiter,$$

where $\mathcal{O}_1$ is the cost to find an approximate solution x and $\mathcal{O}_2$ the cost to solve $\mathbf{Ke} = \mathbf{r}$.

Usually, iterative methods like Newton-Raphson method or SQP method will be used to find an approximate solution.

If we employ fast verification methods for linear systems, then $\mathcal{O}_2 = \mathcal{O}(n^3)$. However, accurate methods will require larger cost.

If approximate solution x is close to x^*, Algorithm 1 will return a shape $\mathbf{x}^*$. In addition, $\mathsf{mid}(\mathbf{x}^*)$ can be more close to x^* than x. We consider a following easy algorithm.

Algorithm 2.

Step 0

Input M, q, approximate solution x and find $\mathbf{x}^*$.

Step 1

Set $x = \mathsf{mid}(\mathbf{x}^*)$.
Find new $\mathbf{x}^*$ by Algorithm 1.
If shaper $\mathbf{x}^*$ is obtained, then repeat Step 1, else return $\mathbf{x}^*$.

4 Numerical Experiments

The numerical testing was carried out on a PC (Intel Core i7 860 processor, 16GB of memory) with the use of MATLAB R2010a and INTLAB 6 [10, 11].

In INTLAB, the function setround allows to change the rounding mode of the processor and the function intval allows interval arithmetics automatically. Moreover, the function verifylss returns an interval vector including solution set of an interval linear system. For example,

$$\mathbf{e} = \mathsf{verifylss}(\mathbf{K}, \mathbf{r}),$$

returns the solution of $\mathbf{Ke} = \mathbf{r}$. If $\mathbf{K}$ includes singular matrices, verifylss will returns NaN or Inf.

The exact solution satisfies $r(x^*) = 0$. $0 \in \mathbf{r}(\mathbf{x}^*)$ was checked by using the INTLAB function in. If $\mathsf{in}(0, \mathbf{r})$ returns an all-ones vector, then $0 \in \mathbf{r}$.

In tables, CPU times are measured with MATLAB function cputime. In Tables 3-5, the CPU time (sec.) is listed in [ ].

For Example 4 and 5, we compared the error bounds of proposed algorithm with other methods. In these examples, matrices M are H-matrix with positive diagonal elements. Then, following error bounds can be applied (Corollary 4.4 and Corollary 4.7 in [1]):

$$\|x - x^*\|_\infty \le \|\Sigma(\mathbf{J}_{\mathbb{R}^n\Delta}, h_\Delta(x))\|_\infty =: E_\Delta^{\mathrm{lis}}(x), \tag{4.1}$$

$$\|x - x^*\|_p \le \|\langle M\rangle^{-1} \max\{\Lambda,\ \Delta^{-1}\}\|_p \|h_\Delta(x)\|_p =: E_{\Delta,p}^{\mathrm{bnd}}(x), \tag{4.2}$$

where,

$$h_\Delta(x) = \min\{x,\ \Delta(Mx + q)\},$$

$$\mathbf{J}_{\mathbb{R}^n\Delta} = \begin{cases} [\delta_i \min\{0,\ M_{ij}\},\ \delta_i \max\{0,\ M_{ij}\}], & j \ne i, \\ [\delta_i\ M_{ii},\ 1], & j = i, \end{cases}$$

$\langle M\rangle$ is the comparison matrix of M, Λ is diagonalpart of M, $\Delta = \mathrm{diag}(\delta)$, $\delta_i = 1/M_{ii}$. We set

$$E^{AW} := \min\{E_\Delta^{\mathrm{lis}}(x),\ E_{\Delta,\infty}^{\mathrm{bnd}}(x)\},$$

as the error bound proposed by Alefeld and Wang [1].

Moreover, the function verlcpall can be applied to estimate the error bound of LCP [9].

4.1 Example 1

Consider the following P-matrix (Example 5.10.4 in [3]),

$$M = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}.$$

Here, we set $q = (0, -1)^T$ and consider that

$$\min\left(\begin{pmatrix} x_1 + tx_2 \\ x_2 - 1 \end{pmatrix}, \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = 0.$$

When $t \geq 0$, then we have $x^* = (0, 1)^T$ and $\beta_\infty(M) = 1 + t$.

Setting $x = (4, 3)^T$ and $t = 1$, we have

$$x - x^* = (4, 2)^T, \quad \|x - x^*\|_\infty = 4, \quad \beta_\infty(M) = 2, \quad \|r(x)\|_\infty = 4.$$

Employing error bound (1.4), we get

$$\|x - x^*\|_\infty \leq \beta_\infty(M)\|r(x)\|_\infty = 8.$$

At $k = 2$, Algorithm 1 finds,

$$\mathbf{x}^* = \begin{pmatrix} [0.0000, & 0.8889] \\ [1.0000, & 1.0000] \end{pmatrix},$$

and

$$\mathbf{y}^* = \begin{pmatrix} [1.0000, & 1.8889] \\ [0.0000, & 0.0000] \end{pmatrix}.$$

Since ${\mathcal{I}_k}^c \cup {\mathcal{J}_k}^c = \mathcal{N}$, it returned exact solution $\mathbf{x}^* = ([0,\ 0],\ [1,\ 1])^T$ by solving (3.5).

4.2 Example 2

Consider the following symmetric positive definite matrix (Remarks 3.12 in [1]),

$$M = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

$\mathbf{K}$ given in (2.1) with $\mathbf{d} = [0, 1]^n$ contains a singular matrix

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

then we couldn't solve $\mathbf{Ke} = \mathbf{r}$. However, $\mathbf{d} \subseteq (0, 1]^n$ then $\mathbf{K}$ is nonsingular.

Employing error bound (3.2), we have

$$\|x - x^*\|_2 \leq 4\|r(x)\|_2 \equiv \overline{e}.$$

If x is sufficiently close to x^*, then (3.2) provides $\overline{e}$ such that $\mathbf{K} = I + \mathrm{diag}(\mathbf{d}^1)(M - I)$ is nonsingular.

Setting $q = (-1, -1)^T$, then we have a unique solution $x^* = (1, 1)^T$.

Let $x = (0.8, 1.2)^T$. Since $x > y = (-0.6, 0.6)^T$, Algorithm 1 provides,

$$\mathbf{d}^1 = \begin{pmatrix} [0.1593, & 1.0000] \\ [0.0682, & 1.0000] \end{pmatrix},$$

and

$$\mathbf{K} = \begin{pmatrix} [1.1593, \quad 2.0000] & [-1.0000, \quad -0.1593] \\ [-1.0000, \quad -0.0682] & [1.0682, \quad 2.0000] \end{pmatrix},$$

which is nonsingular. At $k = 12$, Algorithm 1 finds,

$$\mathbf{x}^* = \begin{pmatrix} [0.8065, \quad 1.2361] \\ [0.7032, \quad 1.1862] \end{pmatrix},$$

and

$$\mathbf{y}^* = \begin{pmatrix} [0.0000, \quad 0.7688] \\ [0.0000, \quad 0.5657] \end{pmatrix}.$$

Since ${\mathcal{I}_k}^c \cup {\mathcal{J}_k}^c = \mathcal{N}$, it returned exact solution $\mathbf{x}^* = ([1,\ 1],\ [1,\ 1])^T$ by solving (3.5).

4.3 Example 3. Convex hulls in the plane

Let $(u_i, v_i) \in \mathbb{R}^2$ be the measurement points $(i = 0, 1, \cdots, n+1)$. We assume that $u_i < u_{i+1}$ for $i = 0, 1, \cdots, n$. We consider the convex hull of the points [3]. Let $f(u)$ be a ceil of convex hull and we denote $x_i = f(u_i) - v_i$. The edge of convex hull contains some points. $x_i = 0$ implies that a point (u_i, v_i) is on the ceil, whereas $x_i > 0$ implies that the point is under the ceil. Considering the slope of the ceil,

$$\frac{f(u_i) - f(u_{i-1})}{u_i - u_{i-1}} > \frac{f(u_{i+1}) - f(u_i)}{u_{i+1} - u_i},$$

implies that (u_i, v_i) is on the ceil, whereas

$$\frac{f(u_i) - f(u_{i-1})}{u_i - u_{i-1}} = \frac{f(u_{i+1}) - f(u_i)}{u_{i+1} - u_i},$$

implies that (u_i, v_i) is under the ceil. Moreover, (u_0, v_0) and (u_{n+1}, v_{n+1}) are on the ceil. From above relations, we obtain M and q such that

$$M_{ij} = \begin{cases} 1/(u_i - u_{i-1}) + 1/(u_{i+1} - u_i), & i = 1, \cdots, n,\ j = i, \\ -1/(u_i - u_{i-1}), & i = 2, \cdots, n,\ j = i-1, \\ -1/(u_{i+1} - u_i), & i = 1, \cdots, n-1,\ j = i+1, \\ 0, & \text{otherwise}, \end{cases}$$

$$q_i = \frac{v_i - v_{i-1}}{u_i - u_{i-1}} - \frac{v_{i+1} - v_i}{u_{i+1} - u_i}, \quad i = 1, \cdots, n.$$

This M is an $n \times n$ symmetric positive definite matrix. By solving a LCP (1.1), we obtain the ceil of convex hull. The floor can be obtained in a similar manner. Thus, by solving two LCP, we can obtain the convex hull of the points.

In this example, we use random points by utilizing MATLAB function `rand`. To find the approximate solutions x, we use Lemke's method [3]. Numerical results with various n are given in Tables 1 and 2. In Table 1, T_1 and T_2 means CPU times to compute approximate solution and validated solution, respectively.

Table 1: Example 3. Algorithm 1.

n	k	$\|\mathbf{r}(x)\|_\infty$	$\|\mathbf{e}^0\|_\infty$	$\|\mathbf{e}^k\|_\infty$	$n(I_k{}^c)$	$\|\mathsf{rad}(\mathbf{x}^*)\|_\infty$	T_1	T_2
100	2	8.7e-15	3.7e-12	3.2e-14	4	2.1e-14	1.99681	0.0936
500	2	4.4e-14	3.9e-10	7.9e-12	6	5.3e-13	222.597	0.8581
1000	2	1.2e-13	3.9e-09	1.2e-11	7	1.5e-12	1752.86	14.025
1500	2	1.7e-13	1.2e-08	6.2e-10	7	7.0e-12	6149.12	37.299
2000	2	3.4e-15	3.7e-10	3.1e-11	11	7.2e-12	13830.0	75.115

Remark 4.1. In Table 1, $\|\mathbf{e}^0\|_\infty$ means the error bound (3.2). Algorithm 1 gives sharper error bounds than (3.2).

$n(I_k{}^c)$ is the number of $x_i^* = 0$ found by Algorithm 1. Lemke's method sets $x_i = 0$ for some i. In this example, $\mathsf{rad}(\mathbf{e})_i$ can be quite small when $x_i = 0$ and $\underline{y}_i > 0$.

T_1 is always larger than T_2. It implies the proposed algorithm for validation can be faster than Lemke's method for finding the approximate solution.

Table 2: Example 3. Algorithm 2

n	iter	$\|\mathbf{r}(\mathsf{mid}(\mathbf{x}^*))\|_\infty$	$\|\mathbf{e}^k\|_\infty$	$\|\mathsf{rad}(\mathbf{x}^*)\|_\infty$	CPU time
100	4	1.12e-16	1.66e-14	1.62e-14	0.3901
500	11	2.22e-16	3.59e-13	3.35e-13	9.0481
1000	10	2.22e-16	1.37e-12	1.36e-12	130.0112
1500	6	2.57e-16	6.33e-12	6.12e-12	224.5791
2000	6	2.50e-16	6.23e-12	6.14e-12	450.2345

Remark 4.2. Validated solutions obtained by Algorithm 2 are more accurate than Algorithm 1. Moreover, residuals $\|\mathbf{r}(\mathsf{mid}(\mathbf{x}^*))\|_\infty$ are quite close to the machine epsilon.

4.4 Example 4.

Let $M \in \mathbb{R}^{n\times n}$ with,

$$M_{ij} = \begin{cases} c, & j = i+1, \\ b + \mu \sin(i/n), & j = i, \\ a, & j = i-1, \\ 0, & \text{otherwise}, \end{cases}$$

set the exact solutions x^* and the vector q as

$$x_i^* = \max\{0,\ v_i - 0.5\} \times 10^{10(w_i - 0.5)}$$

$$q_i = \begin{cases} -(Mx^*)_i, & x_i^* > 0, \\ -(Mx^*)_i + \max\{0,\ \tilde{v}_i - 0.5\} \times 10^{10(\tilde{w}_i - 0.5)}, & x_i^* = 0. \end{cases}$$

We consider four types of choose the prameters $\pi = (\mu, a, b, c)$:

$$\pi_1 = (0,\ -1,\ 2,\ -1), \qquad \pi_2 = (n^{-2},\ -1.5,\ 2,\ -0.5),$$
$$\pi_3 = (1,\ -1.5,\ 3,\ -1.5), \quad \pi_4 = (n^{-2},\ -1.5,\ 2.2,\ -0.5).$$

This example was studied in [1] and [2].

Numerical results with $n = 20, 500$ are given in Tables 3 and 4. In these tables, we show the ∞-norm of exact error and the upper bounds of $\|x - x^*\|_\infty$ obtained by three algorithms, Algorithm 2, E^{AW} and verlcpall.

Table 3: Error bounds for Example 4. $n = 20$.

	Exact	Algorithm 2	E^{AW}	verlcpall
π_1	2.84e-14	2.85e-14 [0.2340]	2.23e-13 [0.2184]	2.85e-14 [3683.3]
π_2	1.50e-12	6.54e-12 [0.2185]	7.40e-12 [0.2652]	6.05e-11 [1704.8]
π_3	9.43e-14	2.66e-13 [0.2964]	4.24e-13 [0.2344]	1.42e-13 [3553.0]
π_4	9.84e-13	2.30e-12 [0.3120]	2.62e-12 [0.2496]	4.13e-11 [1994.8]

Table 4: Error bounds for Example 4. $n = 500$.

	Exact	Algorithm 2	E^{AW}	verlcpall
π_1	8.44e-10	1.62e-09 [244.21]	7.75e-07 [134.25]	—
π_2	5.92e-10	1.59e-09 [173.78]	8.18e-09 [136.45]	—
π_3	5.09e-10	1.37e-09 [176.74]	5.48e-09 [136.34]	—
π_4	1.01e-09	1.39e-09 [182.36]	1.63e-09 [136.67]	—

4.5 Example 5. Journal Bearing Problem.

Let $M \in \mathbb{R}^{n\times n}$ with,

$$M_{ij} = \begin{cases} -h_{i+1}^3, & j = i+1, \\ h_i^3 + h_{i+1}^3, & j = i, \\ -h_i^3, & j = i-1, \\ 0, & \text{otherwise,} \end{cases}$$

and let q with,

$$q_i = \mu(h_{i+1} - h_i),$$

where

$$h_i = (1 + \varepsilon \cos((i - 0.5)\mu\pi))/\sqrt{\pi}.$$

This example arises from a finite different discretization of a free boundary problem [4]. We choose the prameters $\varepsilon = 0.8$ and $\mu = 20/n$. The numerical results are given in Table 5.

Remark 4.3. In Tables 3-5, results of Algorithm 2 are more accurate than E^{AW}. For smaller problems, sometimes verlcpall returned accurate results, however it required huge computational labor.

Table 5: Error bounds for Example 5.

n	Algorithm 2	E^{AW}	verlcpall
10	3.79e–14 [0.2652]	3.74e–13 [0.1092]	3.85e–13 [6.95533]
25	9.11e–11 [0.3120]	6.55e–10 [0.4836]	7.96e–12 [35371.6]
100	1.72e–10 [1.7940]	2.21e–10 [5.4288]	—
500	5.87e–09 [188.33]	1.38e–08 [135.42]	—
1000	2.30e–08 [994.25]	1.12e–07 [554.69]	—
1500	4.99e–08 [2588.7]	4.20e–07 [1252.7]	—
2000	9.21e–08 [4414.5]	1.05e–06 [2249.7]	—

Acknowledgements

I am very grateful to the two referees for their helpful comments and suggestions. I am grateful to Professor Xiaojun Chen. She was the first person who introduced me to LCP, and helped me to learn more on this subject which is the core of this research. I would like to thank Professor Mitsuhiro T. Nakao and Professor Hiroshi Nakazato for giving me helpful comments.

References

[1] G.E. Alefeld and Z. Wang, Error estimation for nonlinear complementarity problems via linear systems with interval data, *Numer. Funct. Anal. Optim.* 29 (2008) 243–267.

[2] X. Chen and S. Xiang, Computation of error bounds for P-matrix linear complementarity problem, *Math. Program.* 106 (2006) 513–525.

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

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

[5] N.J. Higham, *Accuracy and Stability of Numerical Algorithms, 2nd Ed.*, SIAM, Philadelphia, 2002.

[6] R. Mathias and J.S. Pang, Error bounds for the linear complementarity problem with a P-matrix, *Linear Algebra Appl.* 132 (1990) 123–136.

[7] G. Mayer and J. Rohn, On the Applicability of the Interval Gaussian Algorithm, *Reliable Computing* 4 (1998) 205–222.

[8] S. Oishi and S.M. Rump, Fast verification of solutions of matrix equations, *Numer. Math.* 90 (2002) 755–773.

[9] J. Rohn, verlcpall, `http://uivtx.cs.cas.cz/~rohn/matlab/verlcpall.html`, 2008.

[10] S.M. Rump, Interval computations with INTLAB, *Brazilian Electronic Journal on Mathematics of Computation*, 1 (1999).

[11] S.M. Rump, Computer-assisted proofs and Self-Validating Methods, in *Handbook on Accuracy and Reliability in Scientific Computation*, B. Einarsson (ed.), SIAM, 2005, pp. 195–240.

[12] S.M. Rump, Verified bounds for singular values, in particular for the spectral norm of a matrix and its inverse, *BIT.*, Online First (2010).

Manuscript received 18 April 2011
revised 9 August 2011, 9 April 2012
accepted for publication 17 July 2012

Takuma Kimura
JST CREST/ Faculty of Science and Engineering, Waseda University
Tokyo, 169-8555, Japan
E-mail address: `tkimura@aoni.waseda.jp`