Linear Algebra and its Applications 434 (2011) 2343–2352

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Solving a class of matrix minimization problems by linear variational inequality approaches

Min Tao [a,c,1], Xiao-ming Yuan [b,2], Bing-sheng He [a,*,3]

[a] Department of Mathematics, Nanjing University, Nanjing 210093, China
[b] Department of Mathematics, Hong Kong Baptist University, Hong Kong, China
[c] School of Science, Nanjing University of Posts and Telecommunications, Nanjing, China

ARTICLE INFO

Article history:
Received 30 September 2008
Accepted 19 November 2010
Available online 19 January 2011

Submitted by V. Mehrmann

Keywords:
Matrix minimization
Projection and contraction method

ABSTRACT

A class of matrix optimization problems can be formulated as a linear variational inequalities with special structures. For solving such problems, the projection and contraction method (PC method) is extended to variational inequalities with matrix variables. Then the main costly computational load in PC method is to make a projection onto the semi-definite cone. Exploiting the special structures of the relevant variational inequalities, the Levenberg–Marquardt type projection and contraction method is advantageous. Preliminary numerical tests up to 1000×1000 matrices indicate that the suggested approach is promising.

1. Introduction

Let S^n be the set of all real symmetric matrices. For $C \in S^n$, we use $C \succeq 0$ to express that C is a positive semi-definite matrix while $C \geq 0$ expresses that each element of C is non-negative. We will use the inner product

$$\langle A, B\rangle := \text{Trace}\left(A^T B\right)$$

* Corresponding author.
E-mail addresses: xmyuan@hkbu.edu.hk (X.-m. Yuan), hebma@nju.edu.cn (B.-s. He).

1 This author was supported by the Scientific Research Foundation of Nanjing University of Posts and Telecommunications (NY210049).
2 This author was supported by the RGC 203009, Hong Kong.
3 This author was supported by the NSFC Grant 10971095 and Jiangsu NSF BK2008255.

0024-3795/$ - see front matter © 2011 Published by Elsevier Inc.
doi:10.1016/j.laa.2010.11.041

defined on the class of real $m \times n$ matrices and which induces the Fröbenis-norm

$$\|A\| = (\text{Trace}(A^T A))^{1/2} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2 \right)^{1/2}.$$

Using vec(A) to denote the mn dimensional vector obtained by stacking the n columns of A on top of one another, we have

$$\text{vec}(A)^T \text{vec}(B) = \text{Trace}\left(A^T B\right).$$

For any given $m \times m$ symmetric positive definite matrix G, we denote

$$\|A\|_G = \sqrt{\langle A, GA \rangle} = \left(\text{Trace}\left(A^T GA\right)\right)^{1/2}.$$

The problem considered in this paper is to find the projection of a given matrix onto the intersection of S^n_Λ and S_B. The mathematical form of the considered problem is

$$\min \left\{ \frac{1}{2} \|X - C\|^2 \mid X \in S^n_\Lambda \cap S_B \right\}, \tag{1.1}$$

where

$$S^n_\Lambda = \left\{ H \in R^{n \times n} \mid H^T = H, \ \lambda_{\min} I \preceq H \preceq \lambda_{\max} I \right\}, \tag{1.2}$$

and

$$S_B = \left\{ H \in R^{n \times n} \mid H^T = H, \ H_L \leq H \leq H_U \right\}, \tag{1.3}$$

C, H_L and H_U are given $n \times n$ symmetric matrices, $\lambda_{\min} \leq \lambda_{\max}$ are given scalars and $S^n_\Lambda \cap S_B$ is nonempty. An important special case of S^n_Λ is that $\lambda_{\min} = 0$ and $\lambda_{\max} = +\infty$ which is denoted by

$$S^n_+ = \left\{ H \in R^{n \times n} \mid H^T = H, \ H \succeq 0 \right\}.$$

In the literature of interior point algorithms, S^n_+ is called semi-definite cone and the related problem belongs to the class of semi-definite programming [10].

Problem (1.1) comes up in several contexts. One is in making adjustments to a symmetric matrix so that it is consistent with prior knowledge or assumptions, and is a valid covariance matrix. The other application is also a special case of the basic problem (1.1), in which each of H_L and H_U diagonal element equals 1. For given matrix C, finding the least square correlation matrix [13] is a problem of form (1.1).

The main property of problem (1.1) is its large size and special structure. Since the projection and contraction methods are advantageous for large problem with special structure [6], instead of using interior point algorithms [10], we consider to use some extended projection and contraction (PC) methods [2,3].

The paper is organized as follows: In Section 2, we convert the problem to a monotone matrix linear variational inequality (abbreviated as MLVI). In Section 3, we extend the basic projection and contraction method for matrix variational inequalities and a practical algorithm is suggested. Section 4 illustrates the implementations of the MLVI approach. Preliminary numerical results are reported in this section. Finally, we give some concluding remarks.

2. The equivalent MLVI formulation

Let Ω be a nonempty closed convex set of $R^{m \times n}$, for given $v \in R^{m \times n}$, the projection mapping under Fröbenis-norm, denoted by $P_\Omega(v)$, is the unique solution of the following problem

$$\min_u \{ \|u - v\| \mid u \in \Omega \}.$$

A basic property of the projection mapping on a closed convex set is

$$\langle w - P_\Omega(w), \ v - P_\Omega(w) \rangle \leq 0, \quad \forall w \in R^{m \times n}, \ \forall v \in \Omega. \tag{2.1}$$

In the case that $S_B = R^{n\times n}$, the considered problem is reduced to

$$\min\left\{\frac{1}{2}\|X-A\|^2 \mid X \in S^n_\Lambda\right\}. \tag{2.2}$$

The solution of (2.2) is called the projection of A on S^n_Λ and denoted by $P_{S^n_\Lambda}(A)$. Using the fact that matrix Fröbenis-norm is invariant by similarity transformation, it is known (see [12], for instance) that

$$P_{S^n_\Lambda}(A) = Q\tilde{\Lambda}Q^T, \tag{2.3}$$

where

$$Q^TAQ = \Lambda, \qquad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \tag{2.4}$$

is the symmetric Schur decomposition of A ($Q = (q_1, \ldots, q_n)$ is an orthogonal matrix whose column vector $q_i,\ i = 1, \ldots, n$ is the eigenvector of A, and $\lambda_i,\ i = 1, \ldots, n$ is the related eigenvalue),

$$\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n) \quad \text{and} \quad \tilde{\lambda}_i = \min(\max(\lambda_i, \lambda_{\min}), \lambda_{\max}).$$

For a tolerance *tol* greater than the unit roundoff, the Symmetric QR Algorithm (Algorithm 8.3.3 in [7]) computes an approximate symmetric Schur decomposition of $Q^TAQ = \Lambda$ in about $9n^3$ flops. In Matlab, the desired Q and Λ of A (2.4) can be obtained by using

$$[Q, \Lambda] = \mathrm{eig}(A). \tag{2.5}$$

In other words, if $S_B = R^{n\times n}$, the complexity of solving problem (1.1)–(1.3) is about $10n^3$ flops.

For S_B in form (1.3), problem (1.1) can be rewritten as

$$\begin{aligned} \min\ & \left\{\tfrac{1}{2}\|X-C\|^2 \mid X \in S^n_\Lambda\right\} \\ \text{s.t }\ & X \geq H_L \\ & X \leq H_U. \end{aligned} \tag{2.6}$$

By attaching a Lagrangian multiplier matrix Y and $Z \in R^{n\times n}_+$ to the linear constraint $X - H_L \geq 0$ and $X - H_U \leq 0$, respectively, the Lagrangian function of problem (2.6) is

$$L(X, Y, Z) = \frac{1}{2}\|X-C\|^2 - \langle Y,\ X-H_L\rangle + \langle Z,\ X-H_U\rangle, \tag{2.7}$$

which defined on

$$\Omega = S^n_\Lambda \times R^{n\times n}_+ \times R^{n\times n}_+.$$

Let (X^*, Y^*, Z^*) be the KKT point of problem (2.6), we have

$$(X^*, Y^*, Z^*) \in \Omega, \quad \begin{cases} \langle X-X^*,\ (X^*-C) - Y^* + Z^*\rangle \geq 0, \\ \langle Y-Y^*,\ X^*-H_L\rangle \geq 0, \\ \langle Z-Z^*,\ H_U-X^*\rangle \geq 0, \end{cases} \quad \forall\ (X, Y, Z) \in \Omega. \tag{2.8}$$

The compact form of (2.8) is the following linear variational inequality

$$\mathrm{LVI}(\Omega, M, q) \qquad u^* \in \Omega, \quad \langle u-u^*, Mu^*+q\rangle \geq 0, \quad \forall u \in \Omega, \tag{2.9}$$

where

$$u = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \qquad M = \begin{pmatrix} I_n & -I_n & I_n \\ I_n & 0_n & 0_n \\ -I_n & 0_n & 0_n \end{pmatrix}, \qquad q = \begin{pmatrix} -C \\ -H_L \\ H_U \end{pmatrix}. \tag{2.10}$$

We call (2.9) and (2.10) *Matrix linear variational inequality* for problem (1.1), in short, MLVI.

3. Projection and contraction method for MLVI

The projection and contraction method can be extended from linear variational inequalities [2,3] to matrix linear variational inequalities straightforwardly. Let Ω be a convex closed subset of $R^{m\times n}$, the mathematical form of matrix linear variational inequality (abbreviated as MLVI) is: Find u^*, such that

$$\text{MLVI}(\Omega, M, q) \qquad u^* \in \Omega, \quad \langle u - u^*, Mu^* + q\rangle \geq 0, \quad \forall u \in \Omega, \tag{3.1}$$

where $M \in R^{m\times m}$ and $q \in R^{m\times n}$ are given matrices. If $n = 1$, the matrix linear variational inequality (3.1) is reduced to the classical one. A matrix linear variational inequality is monotone if and only if the matrix $(M + M^T)$ is positive semi-definite (M is not necessary symmetric). It is clear that the matrix M in (2.10) is positive semi-definite.

Assume that the solution set of MLVI(Ω, M, q), denoted by Ω^*, is nonempty. Similarly as in the classical LVIs [1], the MLVI(Ω, M, q) is equivalent to the following matrix projection equation

$$u = P_\Omega[u - (Mu + q)]. \tag{3.2}$$

In other words, to solve MLVI(Ω, M, q) is equivalent to finding a zero point of the continuous residue function

$$e(u) := u - P_\Omega[u - (Mu + q)]. \tag{3.3}$$

Hence,

$$e(u) = 0 \quad \Leftrightarrow \quad u \in \Omega^*.$$

In the literature for classical variational inequalities, $\|e(u)\|$ is called *error bound* of LVI. It quantitatively measures how much u fails to be in Ω^*.

Let $u^* \in \Omega^*$ be a solution. For any $u \in R^{m\times n}$, because $P_\Omega[u - (Mu + q)] \in \Omega$, it follows from (3.1) that

$$\langle P_\Omega[u - (Mu + q)] - u^*,\ Mu^* + q\rangle \geq 0,\ \forall\, u \in R^{m\times n}.$$

By setting $w = u - (Mu + q)$ and $v = u^*$ in (2.1), we have

$$\langle P_\Omega[u - (Mu + q)] - u^*,\ [u - (Mu + q)] - P_\Omega[u - (Mu + q)]\rangle \geq 0, \quad \forall\, u \in R^{m\times n}.$$

Adding the above two inequalities and using the notation of $e(u)$, we obtain

$$\langle (u - u^*) - e(u),\ e(u) - M(u - u^*)\rangle \geq 0, \quad \forall\, u \in R^{m\times n}. \tag{3.4}$$

For positive semi-definite (not necessary symmetric) matrix M, the following theorem follows from (3.4) directly.

Theorem 3.1 (Theorem 1 in [2]). *For any $u^* \in \Omega^*$, we have*

$$\langle u - u^*,\ d(u)\rangle \geq \|e(u)\|^2, \quad \forall\, u \in R^{m\times n}, \tag{3.5}$$

where

$$d(u) = \left(I + M^T\right) e(u). \tag{3.6}$$

For $u \in \Omega\backslash\Omega^*$, it follows from (3.5) that $-G^{-1}(I + M^T)e(u)$ is a descent direction of the unknown function $\frac{1}{2}\|u - u^*\|_G^2$, where G is a symmetric positive matrix. Based on Theorem 3.1, we state our approach for MLVI as the following framework.

The Framework of Projection and Contraction Method for MLVI

Given $u_0 \in \Omega$ and $\gamma \in (0, 2)$.
While not converged **do**
1. computer $e\left(u^k\right) = u^k - P_\Omega[u^k - (Mu^k + q)]$; where M and q are defined in (2.10).
2. update

$$u^{k+1} = u^k - \gamma\alpha\left(u^k\right) G^{-1} d\left(u^k\right), \tag{3.7}$$

where

$$\alpha\left(u^k\right) = \frac{\|e\left(u^k\right)\|^2}{\|G^{-1}d\left(u^k\right)\|_G^2} \tag{3.8}$$

and $d(u)$ is defined as in (3.6).
Until stopping criterion is satisfied.

Theorem 3.2. *The method* (3.7) *and* (3.8) *produces a sequence* $\{u^k\}$, *which satisfies*

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - \gamma(2-\gamma)\alpha\left(u^k\right)\|e\left(u^k\right)\|^2. \tag{3.9}$$

Proof. The proof is elementary and similar as Theorem 2 in [2] and omitted.
It follows from (3.6) and (3.8) that

$$\alpha(u) \ge 1/\|(I+M)G^{-1}\left(I+M^T\right)\|,$$

where

$$\|(I+M)G^{-1}\left(I+M^T\right)\| = \lambda_{\max}((I+M)G^{-1}(I+M^T)).$$

Therefore, from the contraction inequality (3.9) we obtain

$$\|u^{k+1} - u^*\|_G^2 \le \|u^k - u^*\|_G^2 - \frac{\gamma(2-\gamma)}{\|(I+M)G^{-1}(I+M^T)\|}\|e\left(u^k\right)\|^2. \tag{3.10}$$

Since the above inequality is true for all $u^* \in \Omega^*$, we have

$$\mathrm{dist}_G^2\left(u^{k+1}, \Omega^*\right) \le \mathrm{dist}_G^2\left(u^k, \Omega^*\right) - \frac{\gamma(2-\gamma)}{\|(I+M)G^{-1}\left(I+M^T\right)\|}\|e\left(u^k\right)\|^2, \tag{3.11}$$

where

$$\mathrm{dist}_G(u, \Omega^*) = \min\{\|u - u^*\|_G \mid u^* \in \Omega^*\}.$$

Such method is called *projection and contraction method*, because it makes projection in each iteration and the generated sequence satisfies (3.11), i.e., it is Fejér monotone with respect to the solution set.

Since problem (1.1) has a unique solution, the solution set of LVI(Ω, M, q) (2.9) and (2.10) is non-empty. It is clear that the matrix M in (2.10) is positive semi-definite and MLVI(Ω, M, q) (2.9) and (2.10) can be solved by PC method (3.7) and (3.8). For such problem, the following algorithms are practical. □

Algorithm 3.1. Setting $G = (I + M^T)(I + M)$ in (3.7) and (3.8), we get a special case of the projection and contraction algorithm. In this case, $\alpha(u) \equiv 1$ (*see* (3.8)). The recursion is simplified to

$$u^{k+1} = u^k - \gamma(I+M)^{-1}e\left(u^k\right). \tag{3.12}$$

Since $G = (I + M^T)(I + M)$, it follows from (3.10) that the sequence $\{u^k\}$ generated by Algorithm 3.1 satisfies

$$\left\|u^{k+1} - u^*\right\|_G \leqslant \left\|u^k - u^*\right\|_G^2 - \gamma(2-\gamma)\left\|e\left(u^k\right)\right\|^2. \tag{3.13}$$

Algorithm 3.1 with $\gamma = 1$ was presented in [3] (see Method 4 in [3]). Note that Algorithm 3.1 with $\gamma \in (0, 2)$ is also the exact method in [5] when the general variational inequality reduced to a linear variational inequality. If M is symmetric and $\Omega = R^{m\times n}$, Algorithm 3.1 is reduced to the Levenberg–Marquardt method for unconstrained optimization. Thus, the method is called *Projection and Contraction Method of Levenberg–Marquardt type* [3].

Remark 3.1. Let $F(u) = Mu + q$, another popular formulation for VIs is the multi-valued equation

$$0 \in F(u) + N_\Omega(u),$$

where $N_\Omega(\cdot)$ is the normal cone operator to Ω, i.e.,

$$N_\Omega(u) := \begin{cases} \{w \mid (v-u)^T w \le 0, \quad \forall v \in \Omega\} & \text{if } u \in \Omega, \\ \emptyset & \text{otherwise.} \end{cases}$$

For solving VI(Ω, F), there are operator splitting algorithms

$$u^{k+1} = (I+F)^{-1}(I - N_\Omega)(I + N_\Omega)^{-1}(I - F)u^k, \tag{3.14}$$

which was introduced by Peaceman–Rachford [9], and

$$u^{k+1} = (I+F)^{-1}[(I+N_\Omega)^{-1}(I-F) + F]u^k, \tag{3.15}$$

which was introduced by Douglas–Rachford [8]. Since N_Ω denote the normal cone operator to Ω, so $(I + N_\Omega)^{-1}$ is just the orthogonal projection operator onto Ω, i.e., $(I + N_\Omega)^{-1} \equiv P_\Omega$. Therefore, using the notation of $e(u)$ (see (3.3)), the Peaceman-Rachford and the Douglas–Rachford recursions can be rewritten as

$$u^{k+1} = (I+F)^{-1}\left[u^k + F\left(u^k\right) - 2e\left(u^k\right)\right], \tag{3.16}$$

and

$$u^{k+1} = (I+F)^{-1}\left[u^k + F\left(u^k\right) - e\left(u^k\right)\right], \tag{3.17}$$

respectively. In [11] (p. 240), Varga suggested to combine the Peaceman–Rachford and the Douglas–Rachford algorithms into a single algorithm depending on a parameter γ, which gives Peaceman–Rachford for $\gamma = 2$ and Douglas–Rachford for $\gamma = 1$. Hence, the related recursions can be written as

$$u^{k+1} = (I+F)^{-1}\left[u^k + F\left(u^k\right) - \gamma e\left(u^k\right)\right], \tag{3.18}$$

and it is called the Douglas-Peaceman–Rachford–Varga operator splitting method (in short DPRV method). Since $F(u) = Mu + q$, it follows from (3.18) that

$$(I+M)u^{k+1} = (I+M)u^k - \gamma e\left(u^k\right). \tag{3.19}$$

This is just the recursion form of Algorithm 3.1. Therefore, Algorithm 3.1 is a special form of the DPRV operator splitting method for linear variational inequalities.

Theorem 3.3. *Let $\{u^k\}$ be the sequence generated by Algorithm 3.1. Then the sequence of the residue function $\{\|e\left(u^k\right)\|\}$ is monotonically decreasing. In detail, it satisfies*

$$\left\|e\left(u^{k+1}\right)\right\|^2 \le \left\|e\left(u^k\right)\right\|^2 - \frac{2-\gamma}{\gamma}\left\|e\left(u^k\right) - e\left(u^{k+1}\right)\right\|^2. \tag{3.20}$$

Proof. For convenience, instead of u^k and u^{k+1}, we write u and $\tilde{u}$ in the proof. By a manipulation, we have

$$\|e(u)\|^2 = \|e(\tilde{u})\|^2 - \|e(u) - e(\tilde{u})\|^2 + 2\langle e(u),\ e(u) - e(\tilde{u})\rangle. \tag{3.21}$$

Using (2.1) (setting $w = \tilde{u} - (M\tilde{u} + q)$ and $v = P_\Omega[u - (Mu + q)]$ in (2.1)), we have

$$\langle \tilde{u} - (M\tilde{u} + q) - P_\Omega[\tilde{u} - (M\tilde{u} + q)],\ P_\Omega[\tilde{u} - (M\tilde{u} + q)] - P_\Omega[u - (Mu + q)]\rangle \geq 0. \tag{3.22}$$

Exchanging u and $\tilde{u}$ in (3.22), we get

$$\langle P_\Omega[u - (Mu + q)] - u + (Mu + q),\ P_\Omega[\tilde{u} - (M\tilde{u} + q)] - P_\Omega[u - (Mu + q)]\rangle \geq 0. \tag{3.23}$$

Adding (3.22) and (3.23) we obtain

$$\{(e(\tilde{u}) - e(u)) + M(u - \tilde{u})\}^T\{(e(u) - e(\tilde{u})) + (\tilde{u} - u)\} \geq 0,$$

and consequently using the semi-definiteness of M,

$$\langle (u - \tilde{u}) + M(u - \tilde{u}),\ e(u) - e(\tilde{u})\rangle \geq \|e(u) - e(\tilde{u})\|^2. \tag{3.24}$$

Note that from (3.12) we have

$$(I + M)(u - \tilde{u}) = \gamma e(u).$$

Hence, it follows from (3.24) that

$$\langle \gamma e(u),\ e(u) - e(\tilde{u})\rangle \geq \|e(u) - e(\tilde{u})\|^2. \tag{3.25}$$

Substituting (3.25) into (3.21) we get

$$\|e(u)\|^2 \geq \|e(\tilde{u})\|^2 - \|e(u) - e(\tilde{u})\|^2 + \frac{2}{\gamma}\|e(u) - e(\tilde{u})\|^2. \tag{3.26}$$

The assertion of this theorem follows from (3.26) immediately. □

4. Implementations and numerical experiments

This section illustrates the implementation of the suggested algorithms for the matrix linear variational inequality approach. We will see that the implementations are easy to be carried out.

The conversion leads the problem to an MLVI Approach. In matrix linear variational inequality (2.9) and (2.10), for given $u = (X, Y, Z)$, we have

$$Mu + q = \begin{pmatrix} X - Y + Z - C \\ X - H_L \\ H_U - X \end{pmatrix}. \tag{4.1}$$

In each iteration of Algorithm 3.1, we have to calculate $e(u)$ which is given by

$$e(u) = \begin{pmatrix} e_X(u) \\ e_Y(u) \\ e_Z(u) \end{pmatrix} = \begin{pmatrix} X - P_{S^n_\Lambda}[Y - Z + C] \\ Y - P_{R^{n\times n}_+}[Y - X + H_L] \\ Z - P_{R^{n\times n}_+}[Z + X - H_U] \end{pmatrix}. \tag{4.2}$$

Therein the most time consuming operation is to calculate $P_{S^n_\Lambda}[Y - Z + C]$ in $e_X(u)$, it is about $10n^3$ flops. In comparison with the calculation of $e(u)$, the remanent computational load in each iteration of Algorithm 3.1 is insignificant.

Using Algorithm 3.1 to solve the linear variational inequality (2.9) and (2.10) the recursion is

$$u^{k+1} = u^k - \gamma (I+M)^{-1} e\left(u^k\right). \tag{4.3}$$

For matrix M in (2.10),

$$(I+M)^{-1} = \frac{1}{4}\begin{pmatrix} I_n & I_n & -I_n \\ -I_n & 3I_n & I_n \\ I_n & I_n & 3I_n \end{pmatrix},$$

and thus

$$(I+M)^{-1}e(u) = \begin{pmatrix} 0.25e_X(u) + 0.25e_Y(u) - 0.25e_Z(u) \\ -0.25e_X(u) + 0.75e_Y(u) + 0.25e_Z(u) \\ 0.25e_X(u) + 0.25e_Y(u) + 0.75e_Z(u) \end{pmatrix}. \tag{4.4}$$

After getting $e(u)$, the implementation of (4.3) is almost cost-free.

To construct the test examples of problems (1.1), we need only to give matrix C, the sets S_B and S^n_Λ.

- **The matrix** C. The diagonal elements of matrix C are generated from a uniform distribution in the interval $(0, 2)$. The off-diagonal elements of matrix C are generated from a uniform distribution in the interval $(-1, 1)$.
- **The set** S_B. Each diagonal element of both matrix H_L and H_U is equal 1. All of the off-diagonal elements of matrix H_L and H_U are equal -0.1 and 0.1, respectively.
- **The set** S^n_Λ. In the first set of test examples, $S^n_\Lambda = S^n_+$. In the second set of test examples

  $$S^n_\Lambda = \{X \in R^{n\times n} \mid 0 \preceq X \preceq \bar{\lambda} I\}.$$

  where $\bar{\lambda} < \lambda_{\max}$ and $\lambda_{\max}$ is the maximal eigenvalue of X^* which is the solution of the related problem in the first set.

The problem can be converted to MLVI(Ω, M, q) (2.9)–(2.10). We use the starting point $u^0 = (X^0, Y^0, Z^0) = (I_n, 0_n, 0_n)$ and the relaxation factor $\gamma = 1.9$. The problem is tested for $n = 100, 200, \ldots, 1000$.

Since the optimal solution of MLVI(Ω, M, q) is satisfied with $e(u^*) = 0$, we stop the iteration as soon as

$$\frac{\max(\text{abs}(e\left(u^k\right)))}{\max(\text{abs}(e(u^0)))} \le \varepsilon.$$

For $\varepsilon = 10^{-4}$, Tables 4.1 and 4.2 report the iteration numbers and the CPU times for the first set of test examples and the second set of test examples, respectively. In Table 4.1, the last column gives the max eigenvalues of the last iteration, denoted as X^{stop}, when the stop criterion is met. The scalar $\bar{\lambda}$ in the second set of test examples are given in the last column of Table 4.2.

For a reasonable accuracy, the algorithms obtain the solutions in a moderate iteration number. Since the complexity of each iteration is $O(n^3)$ (about $10n^3$), the CPU time is proportional to the

Table 4.1
Test results with $\lambda_{\max} = +\infty$.

n	No. It	CPU Sec.	$\lambda_{\max}(X^{stop})$
100	89	1.9	2.9261
200	99	9.4	3.8855
500	143	154.4	5.7623
800	165	704.0	7.2872
1000	186	1536.5	8.1573

Table 4.2
Test results with bounded $\bar{\lambda}$.

n	No. It	CPU Sec.	$\bar{\lambda}$
100	159	2.5	2.0
200	147	13.5	3.0
500	210	229.1	4.0
800	216	930.9	5.0
1000	234	1956.6	6.0

Table 4.3
Iteration number for different stop accuracies ε $S = S_+^n$.

Matrix$_{n\times n}$	Using MLVI Approach and Algorithm 3.1				
n	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\lambda_{\max}(X^{stop})$
100	15	46	89	157	2.9261
200	12	38	99	171	3.8855
500	21	69	143	233	5.7623
800	23	78	165	269	7.2872
1000	26	87	186	290	8.1573

Table 4.4
Iteration number for different stop accuracies ε $S = S_\Lambda^n$.

Matrix$_{n\times n}$	Using MLVI Approach and Algorithm 3.1				$X \in S_\Lambda^n$
n	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\bar{\lambda}$
100	43	94	159	226	2.0
200	42	89	147	224	3.0
500	55	125	210	296	4.0
800	59	133	216	310	5.0
1000	57	138	234	332	6.0

product of the iteration number by n^3. For solving the test examples with 1000×1000 matrices (500,500 variables), using MLVI Approach and Algorithm 3.1 we need about 20–30 min with a desktop computer. In addition, it seems that all suggested algorithms are linear convergent. Tables 4.3 and 4.4 give the iteration numbers for different ε by using MLVI Approach and Algorithm 3.1.

5. Conclusions remarks

This paper mainly studies the application of the Levenberg–Marquardt type projection and contraction method for solving a class of matrix optimization problems. Preliminary numerical results clarify that our algorithm performs robustly and is easy to handle large scale problems with low complexity. The per-iteration computation of this algorithm is dominated by making projection of a real symmetric matrix onto the semi-definite cone. It should be mentioned that the efficiency of this approach may be improved by some scaling technique, for example, by changing the objective $\frac{1}{2}\|X - C\|^2$ to $\frac{\tau}{2}\|X - C\|^2$ with some suitable $\tau > 0$. In addition, our approach can also be extended to solve (1.1) with different weighted Fröbenis-norms without any effort.

References

[1] B.C. Eaves, On the basic theorem of complementarity, Math. Program. 1 (1971) 68–75.
[2] B.S. He, A new method for a class of linear variational inequalities, Math. Program. 66 (1994) 137–144.
[3] B.S. He, Solving a class of linear projection equations, Numer. Math. 68 (1994) 71–80.
[4] B.S. He, A modified projection and contraction method for a class of linear complementarity problems, J. Comput. Math. 14 (1996) 54–63.
[5] B.S. He, Inexact implicit methods for monotone general variational inequalities, Math. Program. 86 (1999) 199–217.
[6] B.S. He, Solving trust region problem in large scale optimization, J. Comput. Math. 18 (2000) 1–12.
[7] G.H. Golub, C.F. van Loan, Matrix Computation, third ed., The Johns Hopkins University Press, 1996.

[8] J. Douglas, H.H. Rachford, On the numerical solution of the heat conduction problem in 2 and 3 space variables, Trans. Amer. Math. Soc. 82 (1956) 421–439.
[9] D.H. Peaceman, H.H. Rachford, The numerical solution of parabolic elliptic differential equations, J. SIAM 3 (1955) 28–41.
[10] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[11] R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1966.
[12] W. Glunt, An alternating projections method for certain linear problems in a Hilbert space, IMA J. Numer. Anal. 15 (1995) 291–305.
[13] Y. Gao, D.F. Sun, Calibrating least squares semidefinite programming with equality and inequality constraints. SIAM J. Matrix Anal. Appl. 31 (2009) 1432–1457.