Ann. Funct. Anal. 9 (2018), no. 1, 8–16
https://doi.org/10.1215/20088752-2017-0019
ISSN: 2008-8752 (electronic)
http://projecteuclid.org/afa

A NEW ALGORITHM FOR THE SYMMETRIC SOLUTION OF THE MATRIX EQUATIONS $AXB = E$ AND $CXD = F$

CHUNMEI LI,[1*] XUEFENG DUAN,[2] JUAN LI,[3] and SITTING YU[4]

Communicated by Q.-W. Wang

Abstract. We propose a new iterative algorithm to compute the symmetric solution of the matrix equations $AXB = E$ and $CXD = F$. The greatest advantage of this new algorithm is higher speed and lower computational cost at each step compared with existing numerical algorithms. We state the solutions of these matrix equations as the intersection point of some closed convex sets, and then we use the alternating projection method to solve them. Finally, we use some numerical examples to show that the new algorithm is feasible and effective.

1. Introduction

Denote by $R^{m\times n}$ the set of $m \times n$ real matrices, and let A^T and A^+ be the transpose and the Moore–Penrose generalized inverse of the matrix A, respectively. We use I_n and O_n to stand for the $n \times n$ identity matrix and the zero matrix, respectively. The symbol $V_1 \oplus V_2$ stands for the direct sum of two subspaces V_1 and V_2. For $A, B \in R^{m\times n}$, $\langle A, B\rangle = \operatorname{trace}(B^T A)$ denotes the inner product of the matrices A and B. The induced norm is the so-called *Frobenius norm*; that is, if $\|A\| = \langle A, A\rangle^{1/2}$, then $R^{m\times n}$ is a real Hilbert space. As a first step in the development of this article, we need to give the following definition.

Definition 1.1 ([1, p. 420]). Let M be a closed convex subset in a real Hilbert space H, and let u be a point in H. Then the point in M nearest to u is called

Received Dec. 3, 2016; Accepted Jan. 11, 2017.
First published online Jul. 12, 2017.
*Corresponding author.
2010 *Mathematics Subject Classification.* Primary 39B82; Secondary 44B20, 46C05.
Keywords. matrix equation, new algorithm, symmetric solution, alternating projection method.

the *projection* of u onto M and denoted by $P_M(u)$; that is, $P_M(u)$ is the solution of the following minimization problem

$$\min_{x\in M} \|x-u\|, \quad u \in H, \tag{1.1}$$

that is,

$$\big\|P_M(u)-u\big\| = \min_{x\in M} \|x-u\|. \tag{1.2}$$

In this article, we mainly consider the following problem.

Problem I. Given matrices $A \in R^{p\times n}$, $B \in R^{n\times q}$, $C \in R^{s\times n}$, $D \in R^{n\times t}$, $E \in R^{p\times q}$, and $F \in R^{s\times t}$, find $X \in \pounds$ such that

$$AXB = E, \qquad CXD = F, \tag{1.3}$$

where $\pounds$ is the set of $n\times n$ symmetric matrices. Obviously, the set $\pounds$ is a closed convex set.

The matrix equations (1.3) have been extensively studied over the past 40 years or so. Wang [13] gave some conditions for the existence of a solution and some representations of the common solution to (1.3). Based on the projection theorem and matrix decompositions, Liao, Lei, and Yuan [9] gave an analytical expression of the optimal approximate least square symmetric solution of (1.3). However, these direct methods may be less efficient for the large coefficient matrices due to limited computer processing power and storage capacity. Therefore, iterative methods for solving the matrix equations (1.3) have recently attracted much interest. By making use of the idea behind the conjugate gradients method, Sheng and Chen [12] proposed an efficient iterative method to solve (1.3). Recently, Ding and Chen [7] presented the gradient-based iterative algorithms and least-squares-based iterative algorithm for solving (coupled) matrix equations. These methods represent an innovative, computationally efficient numerical algorithm. Dehghan and Hajarian [6] (see also [8]) proposed some efficient methods to solve the coupled matrix equations.

Based on the alternating projection method (see Section 2), we propose a new algorithm to solve Problem I in Section 3. The new algorithm has the following advantages: (1) global convergence; (2) faster and with lower computational cost at each step than the algorithm proposed in [3] and [11]; and (3) involves only matrix-matrix multiplication at each iteration step, making it suitable for parallel computation. In Section 4, we use some numerical examples to show that the new algorithm is feasible and effective.

2. Alternating projection method

Alternating projection is a very simple algorithm for computing a point in the intersection of some convex sets using a sequence of projections onto the sets. For example, suppose that C and D are closed convex sets in R^n, and let P_C and P_D denote the projection on C and D, respectively. The alternating projection method starts with any $x_0 \in R^n$ and then projects into C and D

$$y_k = P_D(x_k), \qquad x_{k+1} = P_C(y_k), \quad k = 0,1,2,\ldots.$$

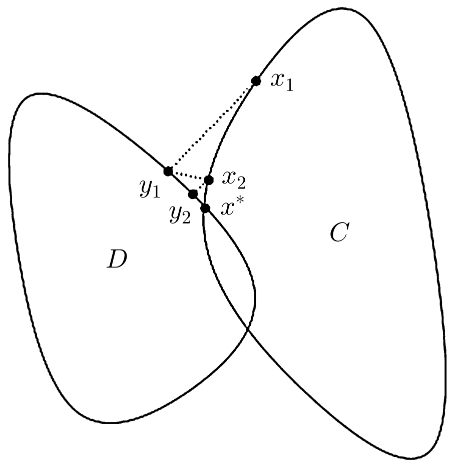

FIGURE 1. First few iterations of the alternating projection method. The sequences $\{x_k\}$ and $\{y_k\}$ both converge to the point $x^* \in C \cap D$.

This generates two sequences: $\{x_k\} \subset C$ and $\{y_k\} \subset D$. According to Theorem 2 in Cheney and Goldstein [4], we know that if $C \cap D \neq \phi$, then the sequences $\{x_k\}$ and $\{y_k\}$ both converge to a point $x^* \in C \cap D$. That is, alternating projections find a point in the intersection of the sets, provided they intersect. This can be illustrated in Figure 1. Later, Brègman [2] extended the alternating projection method to the case of several closed convex sets, and this process can also be found in Orsi [10].

Lemma 2.1 ([10, Theorem 2.2]). *Let $C_1, C_2, \ldots, C_n$ be a family of closed convex sets in a finite-dimensional Hilbert space H. If $C_1 \cap C_2 \cap \cdots \cap C_n \neq \emptyset$, then the sequence $\{x_k\}$ generated by the alternating projection method*

$$\forall x_0 \in H, \quad x_{i+1} = P_{C_{\phi(i)}} , \quad \text{where } \phi(i) \ = (i \bmod n) + 1, i = 0, 1, 2, \ldots, \tag{2.1}$$

converges to the point $x^* \in C_1 \cap C_2 \cap \cdots \cap C_n$.

Remark 2.2. By taking three closed convex sets, for example, the alternating projection method (2.1) can be equivalently written as

$$\begin{aligned} \forall x_0 \in H, \quad y_k &= P_{C_1}(x_k), \\ z_k &= P_{C_2}(y_k), \\ x_{k+1} &= Pc_3(z_k), \quad k = 0, 1, 2, \ldots. \end{aligned} \tag{2.2}$$

3. A NEW ALGORITHM FOR SOLVING PROBLEM I

Based on the alternating projection method, we propose a new algorithm to solve Problem I in this section. We begin with a lemma.

Lemma 3.1 ([5, Theorem 9.3.2]). *Given $Z \in R^{n\times n}$, set*

$$\Re = \{X \in R^{n\times n} \mid AXB = E, A \in R^{p\times n}, B \in R^{n\times q}, E \in R^{p\times q}\}.$$

Then the solution $\widehat{X}$ *of the following problem*

$$\min_{x\in\Re} \|X - Z\|$$

is

$$\widehat{X} = Z + A^{+}(E - AZB)B^{+};$$

that is,

$$\|\widehat{X} - Z\| = \min_{x\in\Re} \|X - Z\|.$$

Now, to begin our consideration of Problem I, we first define two sets

$$\Omega_1 = \{X \in R^{n\times n} \mid AXB = E\},$$
$$\Omega_2 = \{X \in R^{n\times n} \mid CXD = F\}.$$

Obviously, Ω_1 and Ω_2 are closed convex sets. If Problem I is consistent, then $\Omega_1 \cap \Omega_2 \cap \pounds \neq \emptyset$, and the intersection point $X^* \in \Omega_1 \cap \Omega_2 \cap \pounds$ is the solution of Problem I. Hence, we can solve Problem I by finding the intersection point X^* of the sets Ω_1, Ω_2, and $\pounds$. It is here that we begin to use the alternating projection method (2.1) to find the intersection point X^*. As a consequence, we can derive the solution of Problem I.

By Lemma 2.1 and Remark 2.2, we can see that the key problem in implementing the alternating projection method (2.2) is how to compute the projections $P_{\Omega_1}(Z)$, $P_{\Omega_2}(Z)$, and $P_{\pounds}(Z)$ of matrix Z onto Ω_1, Ω_2, and $\pounds$, respectively. This problem may be perfectly solved as follows.

Theorem 3.2. *Suppose that the set* Ω_1 *is nonempty. For a given* $n\times n$ *matrix* Z*, we have*

$$P_{\Omega_1}(Z) = Z + A^{+}(E - AZB)B^{+}. \tag{3.1}$$

Proof. By Definition 1.1, we know that the projection $P_{\Omega_1}(Z)$ is the solution of the following minimization problem

$$\min_{X\in\Omega_1} \|X - Z\|,$$

and according to Lemma 3.1, we get that the solution of this problem is $Z + A^{+}(E - AZB)B^{+}$. Hence,

$$P_{\Omega_1}(Z) = Z + A^{+}(E - AZb)B^{+}.$$ □

Theorem 3.3. *Suppose that the set* Ω_2 *is nonempty. For a given* $n\times n$ *matrix* Z*, we have*

$$P_{\Omega_2}(Z) = Z + C^{+}(E - CZD)D^{+}. \tag{3.2}$$

Proof. The proof is similar to that of Theorem 3.2 and is omitted here. □

Theorem 3.4. *For given* $n\times n$ *matrix* Z*, we have*

$$P_{\pounds}(Z) = \frac{Z + Z^T}{2}. \tag{3.3}$$

Proof. By Definition 1.1, we know that the projection $P_{\pounds}(Z)$ is the solution of the following minimization problem

$$\min_{X\in\pounds} \|X - Z\|.$$

Since

$$\begin{aligned}\|X - Z\|^2 &= \left\|\left(X - \frac{Z+Z^T}{2}\right) + \frac{Z+Z^T}{2}\right\|^2 \\ &= \left\|X - \frac{Z+Z^T}{2}\right\|^2 + \left\|\frac{Z+Z^T}{2}\right\|^2,\end{aligned}$$

$\min_{X\in\pounds} \|X - Z\|$ is equivalent to $\min_{X\in\pounds} \|X - \frac{Z+Z^T}{2}\|$, and so the solution of the minimization problem $\min_{X\in\pounds} \|X - Z\|$ is also $\frac{Z+Z^T}{2}$, that is,

$$P_{\pounds}(Z) = \frac{Z+Z^T}{2}.$$

□

By the alternation projection method (2.2), and noting the projections $P_{\Omega_1}(Z)$, $P_{\Omega_2}(Z)$, and $P_{\pounds}(Z)$ defined by (3.1)–(3.3), we get a new algorithm to solve Problem I, which can be stated as follows.

Algorithm 3.5.

Input: The initial matrix $X_0 \in R^{n\times n}$.

Output: The solution X of Problem I.

Step 1. Set $\widetilde{A} = A^+, \widetilde{B} = B^+, \widetilde{C} = C^+, \widetilde{D} = D^+$;

Step 2. **for** $k = 0, 1, 2, 3, \ldots$:

Step 2.1. $Y_k = P_{\Omega_1}(X_k) = X_k + \widetilde{A}(E - AX_kB)\widetilde{B}$,

Step 2.2. $Z_k = P_{\Omega_2}(X_k) = Y_k + \widetilde{C}(F - CY_kD)\widetilde{D}$,

Step 2.3. $X_{k+1} = P_{\pounds}(Z_k) = \frac{Z_k+Z_k^T}{2}$;

and for

Step 3. $X = X_{k+1}$.

Remark 3.6. It is easy to obtain the following: (1) Algorithm 3.5 has global convergence; (2) compared with the least squares with QR (orthogonal triangular)-decomposition (LSQR) algorithm in [3] and the quasi conjugate gradient (QCG) algorithm in [11], Algorithm 3.5 has lower computational requirements at each step; and (3) Algorithm 3.5 involves only matrix-matrix multiplication at each iteration step, making it suitable for parallel computation.

By Lemma 2.1, we get the convergence theorem for Algorithm 3.5.

Theorem 3.7. *If Problem I is consistent, then the matrix sequence $\{X_k\}$ generated by Algorithm 3.5 converges to the solution X^* of Problem I, that is, $X_k \to X^*$, $k \to +\infty$.*

Proof. If Problem I is consistent, then $\Omega_1 \cap \Omega_2 \cap \pounds \neq \emptyset$. By Lemma 2.1, we know that if the set $\Omega_1 \cap \Omega_2 \cap \pounds \neq \emptyset$, then the matrix sequence $\{X_k\}$ generated by Algorithm 3.5 converges to the intersection point $X^* \in \Omega_1\cap\Omega_2\cap\pounds$, noting that the intersection point X^* is the solution of Problem I. Therefore, the matrix sequence $\{X_k\}$ generated by Algorithm 3.5 converges to the solution X^* of Problem I. □

4. Numerical experiments

In this section, we give some numerical examples to illustrate that the new algorithm is feasible and effective for solving Problem I. (All tests were performed using MATLAB R2013a on a PC with a Pentium(R) Dual-Core CPU at 2.8 GHz.) We denote

$$E(k) = \|E - AX_kB\| + \|F - CX_kD\|,$$

and we use the practical stopping criterion Error $\leq 1.0 \times 10^{-9}$.

Example 4.1. Consider Problem I with

$$A = \begin{pmatrix} 1 & 3 & -1 & 3 & -2 & -3 & 1 \\ 4 & -2 & -1 & -2 & 1 & 0 & -5 \\ -1 & -3 & 1 & -3 & 2 & 3 & -1 \\ 5 & 1 & -2 & 1 & -1 & -3 & -4 \\ 3 & -5 & 0 & -5 & 3 & 3 & 6 \\ -4 & 2 & 1 & 2 & -1 & 0 & 5 \end{pmatrix},$$

$$B = \begin{pmatrix} 2 & 3 & 5 & 1 & -2 & 3 \\ -1 & 1 & 0 & 2 & 1 & 1 \\ 4 & -2 & 2 & -6 & -4 & -2 \\ -5 & -6 & 1 & -1 & 5 & -8 \\ -1 & 1 & 0 & 2 & 1 & 1 \\ -3 & -2 & -5 & 1 & 3 & -2 \\ 1 & 4 & 5 & 3 & -1 & 4 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 2 & -1 & 3 & -4 & 0 & -1 \\ -1 & -3 & 1 & -3 & 2 & 3 & 1 \\ -1 & -4 & 1 & -3 & 0 & 6 & 1 \\ 0 & -1 & 0 & 0 & -2 & 3 & 0 \\ 1 & 3 & -1 & 3 & -2 & -3 & -1 \end{pmatrix},$$

$$D = \begin{pmatrix} 2 & 1 & 3 & -2 \\ -3 & -1 & -4 & 3 \\ 1 & 2 & 3 & -1 \\ 0 & 4 & 4 & 0 \\ -2 & 0 & -2 & 2 \\ 1 & -5 & -4 & -1 \\ -1 & -2 & -3 & 1 \end{pmatrix},$$

$$E = \begin{pmatrix} 0.9712 & 3.5021 & -2.1723 & 2.5309 & -0.9712 & 4.6097 \\ 10.0548 & 13.8518 & 0.1898 & 3.7970 & -10.0548 & 17.8046 \\ -0.9712 & -3.5021 & 2.1723 & -2.5309 & 0.9712 & -4.6097 \\ 11.0260 & 17.3539 & -1.9825 & 6.3279 & -11.0260 & 22.4143 \\ -8.3152 & -30.9627 & 35.7533 & -22.6475 & 8.3152 & -43.4679 \\ -10.0548 & -13.8518 & -0.1898 & -3.7970 & 10.548 & -17.8046 \end{pmatrix},$$

$$F = \begin{pmatrix} 39.8207 & -4.9069 & 34.9138 & -39.8207 \\ -18.0400 & 7.0162 & -11.0238 & 18.0400 \\ 3.7407 & 9.1255 & 12.8662 & -3.7407 \\ 21.7807 & 2.1093 & 23.8900 & -21.7807 \\ 18.0400 & -7.0162 & 11.0238 & -18.0400 \end{pmatrix}.$$

We use Algorithm 3.5 to solve Problem I. Let$X_0 = I_7$. After twenty-eight iterations, we get the solution of Problem I as follows:

$$\begin{aligned} \widehat{X} &\approx X_{28} \\ &= \begin{pmatrix} 1.1889 & -0.1572 & 0.0761 & 0.2460 & -0.6019 & 0.1987 & -0.4129 \\ -0.1572 & 0.3337 & -0.0625 & -0.7588 & 0.6397 & 0.1959 & -0.2890 \\ 0.0761 & -0.0625 & 0.9666 & -0.1515 & 0.0694 & 0.2631 & 0.5798 \\ 0.2460 & -0.7588 & -0.1515 & 0.6134 & -0.2046 & 0.5212 & 0.6396 \\ -0.6019 & 0.6397 & 0.0694 & -0.2046 & 1.9315 & -0.3662 & 0.4055 \\ 0.1987 & 0.1959 & 0.2631 & 0.5212 & -0.3662 & 0.7248 & 0.1193 \\ -0.4129 & -0.2890 & 0.5798 & 0.6396 & 0.4055 & 0.1193 & 0.728 \end{pmatrix}, \end{aligned}$$

and its residual error

$$E(28) \approx \|E - AX_{28}B\| + \|F - CX_{28}D\| = 8.81 \times 10^{-10}.$$

Example 4.1 shows that Algorithm 3.5 can feasibly solve Problem I.

Example 4.2. Consider Problem I with

$$\begin{aligned} A &= \mathrm{rand}(50, n), \\ B &= \mathrm{rand}(n, 30), \\ E &= A * \mathrm{ones}(n) * B, \\ C &= \mathrm{rand}(65, n), \\ D &= \mathrm{rand}(n, 51), \\ F &= C * \mathrm{ones}(n) * D, \end{aligned}$$

where $\mathrm{rand}(s, t)$ stands for an $s \times t$ random matrix and $\mathrm{ones}(n)$ stands for an $n \times n$ matrix whose entries are all 1. Let the initial matrix be $X_0 = O_n$. We use Algorithm 3.5 (denoted by APM) and the algorithms proposed in [3] and [11], which were denoted by LSQR and QCG, respectively, to compute the symmetric solution of (1.3). The experimental results are presented in Figure 2. From Figure 2 we can see that our algorithm has a faster convergence rate than the algorithms proposed in [3] and [11].

5. Conclusion

We state Problem I as finding the intersection point of three closed convex sets in the vector space $R^{n\times n}$. From this point of view, we can use the alternating projection algorithm to compute the intersection point, after which we can derive the solution of Problem I. The new algorithm has a number of advantages. One is higher speed and lower computational cost at each step than the existing numerical algorithms. It also involves only matrix-matrix multiplication at each iteration

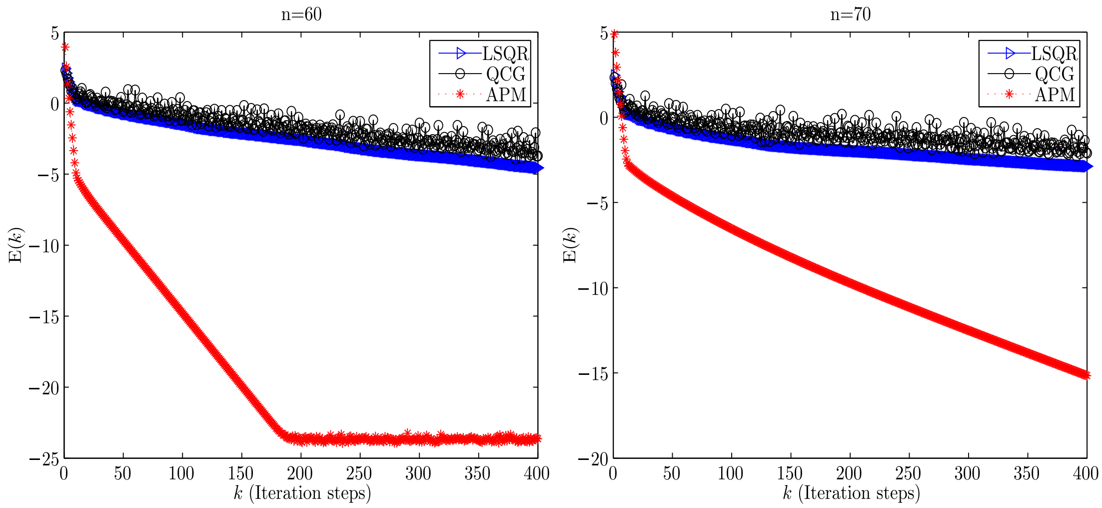

FIGURE 2. Convergence curves of the function $E(k) = \|E - AX_kB\| + \|F - CX_kD\|$.

step, so it is suitable for parallel computation. Finally, this new algorithm can be extended to the generalized matrix equations

$$\begin{cases} A_1XB_1 = C_1, \\ A_2XB_2 = C_2, \\ \vdots \\ A_nXB_n = C_n, \end{cases}$$

over an arbitrary closed convex set $\pounds$, as long as the projection $P_{\pounds}(Z)$ can be derived.

Acknowledgments. The authors wish to thank Mohammad Sal Moslehian and the anonymous referee for providing very useful suggestions, as well as Zhenyun Peng for his insightful and beneficial comments and suggestions.

The authors' work was supported by National Natural Science Foundation of China (NSFC) grant 11561015, by Natural Science Foundation of Guangxi Province grants 2016GXNSFFA380009 and 2016GXNSFAA380074, and by Guangxi Key Laboratory of Cryptography and Information Security grants GCIS201616 and GCIS201615.

REFERENCES

1. H. H. Bauschke and J. M. Borwein, *Dykstra's alternating projection algorithm for two sets*, J. Approx. Theory. **79** (1994), no. 3, 418–443. Zbl 0833.46011. MR1305442. DOI 10.1006/jath.1994.1136. 8
2. L. M. Brègman, *Finding the common point of convex sets by the method of successive projection* (in Russian), Dokl. Akad. Nauk. **162** (1965), 487–490; English translation in Math. Dokl. **6** (1965), 688–692. Zbl 0142.16804. MR0198341. 10
3. Y. Chen, Z. Peng, and T. Zhou, *LSQR iterative common symmetric solutions to matrix equations $AXB = E$ and $CXD = F$*, Appl. Math. Comput. **217** (2010), no. 1, 230–236. Zbl 1206.65142. MR2672581. DOI 10.1016/j.amc.2010.05.053. 9, 12, 14

4. W. Cheney and A. Goldstein, *Proximity maps for convex sets*, Proc. Amer. Math. Soc. **10** (1959), 448–450. Zbl 0092.11403. MR0105008. DOI 10.2307/2032864. 10
5. H. Dai, *Matrix Theory*, Science Press, Beijing, 2001, available online at `http://vdisk.weibo.com/s/dyLs_WJEURz4u` (accessed 30 June 2017). 10
6. M. Dehghan and M. Hajarian, *On the reflexive and anti-reflexive solutions of the generalized coupled Sylvester matrix equations*, Internat. J. Systems Sci. **41** (2010), no. 6, 607–625. Zbl 1196.65081. MR2662731. DOI 10.1080/00207720903072357. 9
7. F. Ding and T. Chen, *On iterative solutions of general coupled matrix equations*, SIAM J. Control Optim. **44** (2006), no. 6, 2269–2284. Zbl 1115.65035. MR2248183. DOI 10.1137/S0363012904441350. 9
8. X. Duan, C. Li, A. Liao, and R. Wang, *On two classes of mixed-type Lyapunov equations*, Appl. Math. Comput. **219** (2013), no. 16, 8486–8495. Zbl 1291.65136. MR3044060. DOI 10.1016/j.amc.2013.02.045. 9
9. A. Liao, Y. Lei, and S. Yuan, *The matrix nearness problem for symmetric matrices associated with the matrix equations* $[A^TXA, B^TXB] = [C, D]$, Linear Algebra Appl. **418** (2006), no. 2–3, 939–954. Zbl 1115.65045. MR2260242. DOI 10.1016/j.laa.2006.03.032. 9
10. R. Orsi, *Numerical methods for solving inverse eigenvalue problems for nonnegative matrices*, SIAM J. Matrix Anal. Appl. **28** (2006), no. 1, 190–212. Zbl 1113.65034. MR2218949. DOI 10.1137/050634529. 10
11. Y. Peng, X. Hu, and L. Zhang, *An iterative method for symmetric solutions and optimal approximation solution of the system of matrix equations* $A_1XB_1 = C_1, A_2XB_2 = C_2$, Appl. Math. Comput. **183** (2006), no. 2, 1127–1137. Zbl 1134.65032. MR2294070. DOI 10.1016/j.amc.2006.05.124. 9, 12, 14
12. X. Sheng and G. Chen, *A finite iterative method for solving a pair of linear matrix equations* $(AXB, CXD) = (C, D)$, Appl. Math. Comput. **189** (2007), no. 2, 1350–1358. Zbl 1133.65026. MR2331808. DOI 10.1016/j.amc.2006.12.026. 9
13. Q. Wang, *A system of matrix equations and a linear matrix equation over arbitrary regular rings with identity*, Linear Algebra Appl. **384** (2004), 44–53. Zbl 1058.15015. MR2055342. DOI 10.1016/j.laa.2003.12.039. 9

[1]Guangxi Key Laboratory of Cryptography and Information Security, College of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin 541004, People's Republic of China.
E-mail address: `lengyue123@126.com`

[2]College of Mathematics and Computational Science, Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, People's Republic of China.
E-mail address: `duanxuefeng@guet.edu.cn`

[3]College of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin 541004, People's Republic of China.
E-mail address: `1584514828@qq.com`

[4]College of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin 541004, People's Republic of China.
E-mail address: `1250875535@qq.com`