

Research Article

LSMR Iterative Method for General Coupled Matrix Equations

F. Toutounian,^{1,2} D. Khojasteh Salkuyeh,³ and M. Mojarab¹

¹ Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Iran

² The Center of Excellence on Modelling and Control Systems, Ferdowsi University of Mashhad, Iran

³ Faculty of Mathematical Sciences, University of Guilan, Iran

Correspondence should be addressed to M. Mojarab; maryam.modjarrab@gmail.com

Received 18 March 2014; Accepted 28 July 2014

Academic Editor: D. R. Sahu

Copyright © F. Toutounian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

By extending the idea of LSMR method, we present an iterative method to solve the general coupled matrix equations $\sum_{k=1}^q A_{ik}X_kB_{ik} = C_i$, $i = 1, 2, \dots, p$, (including the generalized (coupled) Lyapunov and Sylvester matrix equations as special cases) over some constrained matrix groups (X_1, X_2, \dots, X_q) , such as symmetric, generalized bisymmetric, and (R, S) -symmetric matrix groups. By this iterative method, for any initial matrix group $(X_1^{(0)}, X_2^{(0)}, \dots, X_q^{(0)})$, a solution group $(X_1^*, X_2^*, \dots, X_q^*)$ can be obtained within finite iteration steps in absence of round-off errors, and the minimum Frobenius norm solution or the minimum Frobenius norm least-squares solution group can be derived when an appropriate initial iterative matrix group is chosen. In addition, the optimal approximation solution group to a given matrix group $(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_q)$ in the Frobenius norm can be obtained by finding the least Frobenius norm solution group of new general coupled matrix equations. Finally, numerical examples are given to illustrate the effectiveness of the presented method.

1. Introduction

In control and system theory [1–7], we often encounter Lyapunov and Sylvester matrix equations which have been playing a fundamental role. Due to the important roles of the matrix equations, the studies on the matrix equations have been addressed in a large body of papers [8–27]. By using the hierarchical identification principle [9–11, 28–32], a gradient-based iterative (GI) method was presented to solve the solutions and the least-squares solutions of the general coupled matrix equations. In [19, 33], Zhou et al. deduced the optimal parameter of the GI method for solving the solutions and the weighted least-squares solutions of the general coupled matrix equations. Dehghan and Hajarian [34–36] introduced several iterative methods to solve various linear matrix equations.

In [12, 17], Huang et al. presented finite iterative algorithms for solving generalized coupled Sylvester systems.

Li and Huang [37] proposed a matrix LSQR iterative method to solve the constrained solutions of the generalized coupled Sylvester matrix equations. Hajarian [38] presented the generalized QMRCGSTAB algorithm for solving Sylvester-transpose matrix equations. Recently, Lin and Simoncini [39] established minimal residual methods for large scale Lyapunov equations. They explored the numerical solution of this class of linear matrix equations when a minimal residual (MR) condition is used during the projection step.

In this paper, we construct a matrix iterative method based on the LSMR algorithm [40] to solve the constrained solutions of the following problems.

Compatible matrix equations are as follows

$$\sum_{k=1}^q A_{ik}X_kB_{ik} = C_i, \quad i = 1, 2, \dots, p. \quad (1)$$

Least-squares problem is as follows:

$$\min \left(\sum_{i=1}^p \left\| \sum_{k=1}^q A_{ik} X_k B_{ik} - C_i \right\|^2 \right)^{1/2}. \quad (2)$$

Matrix nearness problem is as follows:

$$\min_{(X_1, X_2, \dots, X_q) \in S_E} \left\{ \sum_{k=1}^q \|X_k - \bar{X}_k\|^2 \right\}, \quad (3)$$

where $A_{ik}, B_{ik}, C_i, i = 1, 2, \dots, p, k = 1, 2, \dots, q$, are constant matrices with suitable dimensions, $X_k, k = 1, 2, \dots, q$, are unknown matrices to be solved, $\bar{X}_k, k = 1, 2, \dots, q$, are given matrices, and S_E is the solution set of (1) or problem (2).

This paper is organized as follows. In Section 2, we will briefly review the LSMR algorithm for solving linear systems of equations. In Section 3, we propose the matrix LSMR iterative algorithms for solving the problems (1)-(2). In Section 4, we solve the problem (3) by finding the minimum Frobenius norm solution group of the corresponding new general coupled matrix equations. In Section 5, numerical examples are given to illustrate the efficiency of the proposed iterative method. Finally, we make some concluding remarks in Section 6.

The notations used in this paper can be summarized as follows. $\text{tr}(A)$ represents the trace of the matrix A . For $A, B \in \mathbb{R}^{m \times n}$, notation $A \otimes B$ is Kronecker product and $\langle A, B \rangle = \text{tr}(B^T A)$ is the inner product with the Frobenius norm $\|A\| = \sqrt{\langle A, A \rangle} = \sqrt{\text{tr}(A^T A)}$. The use of $\text{vec}(A)$ represents the vector operator defined as

$$\text{vec}(A) = (a_1^T \ a_2^T \ \dots \ a_n^T)^T, \quad (4)$$

where a_k is the k th column of A . The generalized bisymmetric matrices, the (R, S) -symmetric matrices, and the symmetric orthogonal matrices can be defined as follows.

Definition 1 (see [41]). A matrix $P \in \mathbb{R}^{n \times n}$ is said to be a symmetric orthogonal matrix ($P \in \text{SOR}^{n \times n}$), if $P^T = P$ and $P^2 = I_n$.

Definition 2 (see [42]). For given symmetric orthogonal matrices $R \in \mathbb{R}^{m \times m}, S \in \mathbb{R}^{n \times n}$, we say a matrix $X \in \mathbb{R}^{m \times n}$ is (R, S) -symmetric ($X \in \text{RSSR}^{n \times n}$), if $X = RXS$.

Definition 3 (see [43]). For a given symmetric orthogonal matrix $P \in \mathbb{R}^{n \times n}$, a matrix $X \in \mathbb{R}^{n \times n}$ is said to be a generalized bisymmetric matrix ($X \in \text{GBRS}^{n \times n}$), if $X^T = X$ and $X = PXP$.

2. LSMR Algorithm

In this section, we briefly review some fundamental properties of the LSMR algorithm [40], which is an iterative

method for computing a solution x to either of the following problems.

Compatible linear systems are as follows:

$$Mx = f. \quad (5)$$

Least-squares problem is as follows:

$$\min \|Mx - f\|_2, \quad (6)$$

where $M \in \mathbb{R}^{m \times n}$ and $f \in \mathbb{R}^m$. The LSMR algorithm uses an algorithm of Golub and Kahan [44], which stated as procedure Bidiag 4, to reduce M to lower bidiagonal form. The procedure Bidiag 4 can be described as follows.

Bidiag 4 (starting vector f ; reduction to lower bidiagonal form).

$$\begin{aligned} \beta_1 u_1 &= f, & \alpha_1 v_1 &= M^T u_1, \\ \beta_{i+1} u_{i+1} &= M v_i - \alpha_i u_i, \\ \alpha_{i+1} v_{i+1} &= M^T u_{i+1} - \beta_{i+1} v_i, \\ & & i &= 1, 2, \dots \end{aligned} \quad (7)$$

The scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen such that $\|u_i\|_2 = \|v_i\|_2 = 1$.

The following properties presented in [7] illustrate that the procedure Bidiag 4 has finite termination property.

Property 1. Suppose that k steps of the procedure Bidiag 4 have been taken; then the vectors v_1, v_2, \dots, v_k and $u_1, u_2, \dots, u_k, u_{k+1}$ are the orthonormal basis of the Krylov subspaces $K_k(M^T M, v_1)$ and $K_{k+1}(M M^T, u_1)$, respectively.

Property 2. The procedure Bidiag 4 will stop at step m if and only if $\min\{\mu, \lambda\}$ is m , where μ is the grade of v_1 with respect to $M^T M$ and λ is the grade of u_1 with respect to $M M^T$.

By using the procedure Bidiag 4, the LSMR method constructs an approximation solution of the form $x_k = V_k y_k$, where $V_k = (v_1, v_2, \dots, v_k)$, which solves the least-squares problem, $\min_{y_k} \|M^T r_k\|$, where $r_k = f - M x_k$ is the residual for the approximate solution x_k . The main steps of the LSMR algorithm can be summarized as shown in Algorithm 1.

More details about the LSMR algorithm can be found in [40].

The stopping criterion may be used as $\|M^T r_k\| = \|M^T (f - M x_k)\|_2 = |\bar{c}_{k+1}|$ for the compatible linear systems (5) and for the least-squares problem (6). Other stopping criteria can also be used and are not listed here. Reader can see [40] for details. Clearly, the sequence $x_k \in \text{Range}(M^T)$ generated by the LSMR algorithm converges to the unique minimum norm solution of (5) or the unique minimum norm least-squares solution of problem (6).

Set $\beta_1 u_1 = f, \alpha_1 v_1 = M^T u_1, \bar{\alpha}_1 = \alpha_1, \bar{\zeta}_1 = \alpha_1 \beta_1, \rho_0 = 1, \bar{\rho}_0 = 1, \bar{c}_0 = 1, \bar{s}_0 = 0, h_1 = v_1, \bar{h}_0 = 0, x_0 = 0$
 For $k = 1, 2, \dots$, until convergence Do:

$$\beta_{k+1} u_{k+1} = M v_k - \alpha_k u_k$$

$$\alpha_{k+1} v_{k+1} = M^T u_{k+1} - \beta_{k+1} v_k$$

$$\rho_k = (\bar{\alpha}_k^2 + \beta_{k+1}^2)^{1/2}$$

$$c_k = \bar{\alpha}_k / \rho_k$$

$$s_k = \beta_{k+1} / \rho_k$$

$$\theta_{k+1} = s_k \alpha_{k+1}$$

$$\bar{\alpha}_{k+1} = c_k \alpha_{k+1}$$

$$\bar{\theta}_k = \bar{s}_{k-1} \rho_k$$

$$\bar{\rho}_k = ((\bar{c}_{k-1} \rho_k)^2 + \theta_{k+1}^2)^{1/2}$$

$$\bar{c}_k = \bar{c}_{k-1} \rho_k / \bar{\rho}_k$$

$$\bar{s}_k = \theta_{k+1} / \bar{\rho}_k$$

$$\bar{\zeta}_k = \bar{c}_k \bar{\zeta}_k$$

$$\bar{\zeta}_{k+1} = -\bar{s}_k \bar{\zeta}_k$$

$$\bar{h}_k = h_k - (\bar{\theta}_k \rho_k / (\rho_{k-1} \bar{\rho}_{k-1})) \bar{h}_{k-1}$$

$$x_k = x_{k-1} + (\bar{\zeta}_k / (\rho_k \bar{\rho}_k)) \bar{h}_k$$

$$h_{k+1} = v_{k+1} - (\theta_{k+1} / \rho_k) h_k$$
 If $|\bar{\zeta}_{k+1}|$ is small enough then stop
 End Do.

ALGORITHM 1: LSMR algorithm.

3. A Matrix LSMR Iterative Method

In this section, we will present our matrix iterative method based on the LSMR algorithm, for solving (1) and problem (2). For the unknown matrices $X_i \in \mathbb{R}^{n \times n}, i = 1, 2, \dots, q$, by using the Kronecker product, (1) and problem (2) are equivalent to (5) and problem (6), respectively, with

$$M = \begin{pmatrix} B_{11}^T \otimes A_{11} & B_{12}^T \otimes A_{12} & \cdots & B_{1q}^T \otimes A_{1q} \\ B_{21}^T \otimes A_{21} & B_{22}^T \otimes A_{22} & \cdots & B_{2q}^T \otimes A_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ B_{p1}^T \otimes A_{p1} & B_{p2}^T \otimes A_{p2} & \cdots & B_{pq}^T \otimes A_{pq} \end{pmatrix}, \quad (8)$$

$$x = \begin{pmatrix} \text{vec}(X_1) \\ \text{vec}(X_2) \\ \vdots \\ \text{vec}(X_q) \end{pmatrix}, \quad f = \begin{pmatrix} \text{vec}(C_1) \\ \text{vec}(C_2) \\ \vdots \\ \text{vec}(C_p) \end{pmatrix}.$$

Hence, by using the invariance of the Frobenius norm under unitary transformations, it is easy to prove that the vector form $\beta_1 u_1 = f, \alpha_1 v_1 = M^T u_1, \beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i$, and $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i$ ($i = 1, 2, \dots$) in LSMR algorithm can be rewritten in matrix forms, respectively, as

$$\beta_1 U_1^{(j)} = C_j, \quad j = 1, 2, \dots, p,$$

$$\beta_1 = \left(\sum_{j=1}^p \|C_j\|^2 \right)^{1/2},$$

$$\alpha_1 V_1^{(j)} = \sum_{i=1}^p A_{ij}^T U_1^{(i)} B_{ij}^T, \quad j = 1, 2, \dots, q,$$

$$\alpha_1 = \left(\sum_{k=1}^q \left(\left\| \sum_{i=1}^p A_{ik}^T U_1^{(i)} B_{ik}^T \right\|^2 \right) \right)^{1/2},$$

$$\beta_{i+1} U_{i+1}^{(j)} = \left(\sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} \right) - \alpha_i U_i^{(j)}, \quad j = 1, 2, \dots, p,$$

$$\beta_{i+1} = \left(\sum_{j=1}^p \left(\left\| \sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} - \alpha_i U_i^{(j)} \right\|^2 \right) \right)^{1/2},$$

$$\alpha_{i+1} V_{i+1}^{(k)} = \sum_{j=1}^p A_{jk}^T U_{i+1}^{(j)} B_{jk}^T - \beta_{i+1} V_i^{(k)}, \quad k = 1, 2, \dots, q,$$

$$\alpha_{i+1} = \left(\sum_{k=1}^q \left(\left\| \sum_{j=1}^p A_{jk}^T U_{i+1}^{(j)} B_{jk}^T - \beta_{i+1} V_i^{(k)} \right\|^2 \right) \right)^{1/2}. \quad (9)$$

If the unknown matrices $X_1, X_2, \dots, X_q \in \text{GBRS}^{n \times n}$, then (1) and problem (2) are equivalent to (5) and problem (6), respectively, with

$$M = \begin{pmatrix}
 B_{11}^T \otimes A_{11} & B_{12}^T \otimes A_{12} & \cdots & B_{1q}^T \otimes A_{1q} \\
 B_{21}^T \otimes A_{11} & B_{22}^T \otimes A_{22} & \cdots & B_{2q}^T \otimes A_{2q} \\
 \vdots & \vdots & & \vdots \\
 B_{p1}^T \otimes A_{p1} & B_{p2}^T \otimes A_{p2} & \cdots & B_{pq}^T \otimes A_{pq} \\
 A_{11} \otimes B_{11}^T & A_{12} \otimes B_{12}^T & \cdots & A_{1q} \otimes B_{1q}^T \\
 A_{21} \otimes B_{21}^T & A_{22} \otimes B_{22}^T & \cdots & A_{2q} \otimes B_{2q}^T \\
 \vdots & \vdots & & \vdots \\
 A_{p1} \otimes B_{p1}^T & A_{p2} \otimes B_{p2}^T & \cdots & A_{pq} \otimes B_{pq}^T \\
 (B_{11}^T P_1) \otimes (A_{11} P_1) & (B_{12}^T P_2) \otimes (A_{12} P_2) & \cdots & (B_{1q}^T P_q) \otimes (A_{1q} P_q) \\
 (B_{21}^T P_1) \otimes (A_{21} P_1) & (B_{22}^T P_2) \otimes (A_{22} P_2) & \cdots & (B_{2q}^T P_q) \otimes (A_{2q} P_q) \\
 \vdots & \vdots & & \vdots \\
 (B_{p1}^T P_1) \otimes (A_{p1} P_1) & (B_{p2}^T P_2) \otimes (A_{p2} P_2) & \cdots & (B_{pq}^T P_q) \otimes (A_{pq} P_q) \\
 (A_{11} P_1) \otimes (B_{11}^T P_1) & (A_{12} P_2) \otimes (B_{12}^T P_2) & \cdots & (A_{1q} P_q) \otimes (B_{1q}^T P_q) \\
 (A_{21} P_1) \otimes (B_{21}^T P_1) & (A_{22} P_2) \otimes (B_{22}^T P_2) & \cdots & (A_{2q} P_q) \otimes (B_{2q}^T P_q) \\
 \vdots & \vdots & & \vdots \\
 (A_{p1} P_1) \otimes (B_{p1}^T P_1) & (A_{p2} P_2) \otimes (B_{p2}^T P_2) & \cdots & (A_{pq} P_q) \otimes (B_{pq}^T P_q)
 \end{pmatrix},$$

$$x = \begin{pmatrix} \text{vec}(X_1) \\ \text{vec}(X_2) \\ \vdots \\ \text{vec}(X_q) \end{pmatrix}, \quad f = \begin{pmatrix} \text{vec}(C_1) \\ \text{vec}(C_2) \\ \vdots \\ \text{vec}(C_p) \\ \text{vec}(C_1^T) \\ \text{vec}(C_2^T) \\ \vdots \\ \text{vec}(C_p^T) \\ \text{vec}(C_1) \\ \text{vec}(C_2) \\ \vdots \\ \text{vec}(C_p) \\ \text{vec}(C_1^T) \\ \text{vec}(C_2^T) \\ \vdots \\ \text{vec}(C_p^T) \end{pmatrix},$$

(10)

where $P_1, P_2, \dots, P_q \in \text{SOR}^{n \times n}$. Hence, the vector forms of $\beta_1 u_1 = f$, $\alpha_1 v_1 = M^T u_1$, $\beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i$, and $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i$ ($i = 1, 2, \dots$) in LSMR algorithm can be rewritten in matrix forms, respectively, as

$$\beta_1 U_1^{(j)} = C_j, \quad j = 1, 2, \dots, p,$$

$$\beta_1 = 2 \left(\sum_{j=1}^p \|C_j\|^2 \right)^{1/2},$$

$$\alpha_1 V_1^{(j)} = \sum_{i=1}^p (A_{ij}^T U_1^{(i)} B_{ij}^T + B_{ij} U_1^{(i)T} A_{ij} + (P_j A_{ij}^T) U_1^{(i)} (B_{ij}^T P_j) + (P_j B_{ij}) U_1^{(i)T} (A_{ij} P_i)), \quad j = 1, 2, \dots, q,$$

$$\alpha_1 = \left(\sum_{k=1}^q \left\| \sum_{i=1}^p A_{ik}^T U_1^{(i)} B_{ik}^T + B_{ik} U_1^{(i)T} A_{ik} \right. \right. \\ \left. \left. + (P_k A_{ik}^T) U_1^{(i)} (B_{ik}^T P_k) + (P_k B_{ik}) U_1^{(i)T} (A_{ik} P_k) \right\|^2 \right)^{1/2}, \\ \beta_{i+1} U_{i+1}^{(j)} = \left(\sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} \right) - \alpha_i U_i^{(j)}, \quad j = 1, 2, \dots, p, \\ \beta_{i+1} = 2 \left(\sum_{j=1}^p \left\| \sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} - \alpha_i U_i^{(j)} \right\|^2 \right)^{1/2}, \\ \alpha_{i+1} V_{i+1}^{(k)} = \sum_{j=1}^p (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T + B_{jk} U_{i+1}^{(j)T} A_{jk} + (P_k A_{jk}^T) \\ \times U_{i+1}^{(j)} (B_{jk}^T P_k) + (P_k B_{jk}) U_{i+1}^{(j)T} (A_{jk} P_k)) \\ - \beta_{i+1} V_i^{(k)}, \quad k = 1, 2, \dots, q, \\ \alpha_{i+1} = \left(\sum_{k=1}^q \left\| \sum_{j=1}^p (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T + B_{jk} U_{i+1}^{(j)T} A_{jk} \right. \right. \\ \left. \left. + (P_k A_{jk}^T) U_{i+1}^{(j)} (B_{jk}^T P_k) \right. \right. \\ \left. \left. + (P_k B_{jk}) U_{i+1}^{(j)T} (A_{jk} P_k) \right\|^2 \right)^{1/2}. \tag{11}$$

If the unknown matrices $X_1, X_2, \dots, X_q \in \text{RSSR}^{n \times n}$, then (1) and problem (2) are equivalent to (5) and problem (6), respectively, with

$$M = \begin{pmatrix} B_{11}^T \otimes A_{11} & B_{12}^T \otimes A_{12} & \cdots & B_{1q}^T \otimes A_{1q} \\ B_{21}^T \otimes A_{21} & B_{22}^T \otimes A_{22} & \cdots & B_{2q}^T \otimes A_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ B_{p1}^T \otimes A_{p1} & B_{p2}^T \otimes A_{p2} & \cdots & B_{pq}^T \otimes A_{pq} \\ (B_{11}^T S_1) \otimes (A_{11} R_1) & (B_{12}^T S_2) \otimes (A_{12} R_2) & \cdots & (B_{1q}^T S_q) \otimes (A_{1q} R_q) \\ (B_{21}^T S_1) \otimes (A_{21} R_1) & (B_{22}^T S_2) \otimes (A_{22} R_2) & \cdots & (B_{2q}^T S_q) \otimes (A_{2q} R_q) \\ \vdots & \vdots & \vdots & \vdots \\ (B_{p1}^T S_1) \otimes (A_{p1} R_1) & (B_{p2}^T S_2) \otimes (A_{p2} R_2) & \cdots & (B_{pq}^T S_q) \otimes (A_{pq} R_q) \end{pmatrix}, \tag{12}$$

$$x = \begin{pmatrix} \text{vec}(X_1) \\ \text{vec}(X_2) \\ \vdots \\ \text{vec}(X_q) \end{pmatrix}, \quad f = \begin{pmatrix} \text{vec}(C_1) \\ \text{vec}(C_2) \\ \vdots \\ \text{vec}(C_p) \\ \text{vec}(C_1) \\ \text{vec}(C_2) \\ \vdots \\ \text{vec}(C_p) \end{pmatrix},$$

where $P_i, R_i \in \text{SOR}^{n \times n}, i = 1, 2, \dots, q$. Hence, the vector forms of $\beta_1 u_1 = f, \alpha_1 v_1 = M^T u_1, \beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i$, and $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i (i = 1, 2, \dots)$ in LSMR algorithm can be rewritten in matrix forms, respectively, as

$$\beta_1 U_1^{(j)} = C_j, \quad j = 1, 2, \dots, p, \\ \beta_1 = \sqrt{2} \left(\sum_{j=1}^p \|C_j\|^2 \right)^{1/2}, \\ \alpha_1 V_1^{(j)} = \sum_{i=1}^p (A_{ij}^T U_1^{(i)} B_{ij}^T + R_j (A_{ij}^T U_1^{(i)} B_{ij}^T) S_j), \\ j = 1, 2, \dots, q,$$

$$\alpha_1 = \left(\sum_{k=1}^q \left\| \sum_{i=1}^p A_{ik}^T U_1^{(i)} B_{ik}^T + R_k (A_{ik}^T U_1^{(i)} B_{ik}^T) S_k \right\|^2 \right)^{1/2}, \\ \beta_{i+1} U_{i+1}^{(j)} = \left(\sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} \right) - \alpha_i U_i^{(j)}, \quad j = 1, 2, \dots, p, \\ \beta_{i+1} = \sqrt{2} \left(\sum_{j=1}^p \left\| \sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} - \alpha_i U_i^{(j)} \right\|^2 \right)^{1/2}, \\ \alpha_{i+1} V_{i+1}^{(k)} = \sum_{j=1}^p (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T + R_k (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T) S_k) \\ - \beta_{i+1} V_i^{(k)}, \quad k = 1, 2, \dots, q,$$

$$\alpha_{i+1} = \left(\sum_{k=1}^q \left\| \sum_{j=1}^p (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T + R_k (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T) S_k) - \beta_{i+1} V_i^{(k)} \right\|^2 \right)^{1/2}. \quad (13)$$

If the unknown matrices $X_1, X_2, \dots, X_q \in SR^{n \times n}$ (the set of symmetric matrices), then (1) and problem (2) are equivalent to (5) and problem (6), respectively, with

$$M = \begin{pmatrix} B_{11}^T \otimes A_{11} & B_{12}^T \otimes A_{12} & \cdots & B_{1q}^T \otimes A_{1q} \\ B_{21}^T \otimes A_{21} & B_{22}^T \otimes A_{22} & \cdots & B_{2q}^T \otimes A_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ B_{p1}^T \otimes A_{p1} & B_{p2}^T \otimes A_{p2} & \cdots & B_{pq}^T \otimes A_{pq} \\ A_{11} \otimes B_{11}^T & A_{12} \otimes B_{12}^T & \cdots & A_{1q} \otimes B_{1q}^T \\ A_{21} \otimes B_{21}^T & A_{22} \otimes B_{22}^T & \cdots & A_{2q} \otimes B_{2q}^T \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} \otimes B_{p1}^T & A_{p2} \otimes B_{p2}^T & \cdots & A_{pq} \otimes B_{pq}^T \end{pmatrix}, \quad (14)$$

$$x = \begin{pmatrix} \text{vec}(X_1) \\ \text{vec}(X_2) \\ \vdots \\ \text{vec}(X_q) \end{pmatrix}, \quad f = \begin{pmatrix} \text{vec}(C_1) \\ \text{vec}(C_2) \\ \vdots \\ \text{vec}(C_p) \\ \text{vec}(C_1^T) \\ \text{vec}(C_2^T) \\ \vdots \\ \text{vec}(C_p^T) \end{pmatrix}. \quad (15)$$

Hence, the vector forms of $\beta_1 u_1 = f$, $\alpha_1 v_1 = M^T u_1$, $\beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i$, and $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i$ ($i = 1, 2, \dots$) in LSMR algorithm can be rewritten in matrix forms, respectively, as

$$\beta_1 U_1^{(j)} = C_j, \quad j = 1, 2, \dots, p,$$

$$\beta_1 = \sqrt{2} \left(\sum_{j=1}^p \|C_j\|^2 \right)^{1/2},$$

$$\alpha_1 V_1^{(j)} = \sum_{i=1}^p (A_{ij}^T U_1^{(i)} B_{ij}^T + B_{ij} U_1^{(i)T} A_{ij}), \quad j = 1, 2, \dots, q,$$

$$\alpha_1 = \left(\sum_{k=1}^q \left\| \sum_{i=1}^p A_{ik}^T U_1^{(i)} B_{ik}^T + B_{ik} U_1^{(i)T} A_{ik} \right\|^2 \right)^{1/2},$$

$$\beta_{i+1} U_{i+1}^{(j)} = \left(\sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} \right) - \alpha_i U_i^{(j)}, \quad j = 1, 2, \dots, p,$$

$$\beta_{i+1} = \sqrt{2} \left(\sum_{j=1}^p \left\| \sum_{k=1}^q A_{jk} V_i^{(k)} B_{jk} - \alpha_i U_i^{(j)} \right\|^2 \right)^{1/2},$$

$$\alpha_{i+1} V_{i+1}^{(k)} = \sum_{j=1}^p (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T + B_{jk} U_{i+1}^{(j)T} A_{jk}) - \beta_{i+1} V_i^{(k)},$$

$$k = 1, 2, \dots, q,$$

$$\alpha_{i+1} = \left(\sum_{k=1}^q \left\| \sum_{j=1}^p (A_{jk}^T U_{i+1}^{(j)} B_{jk}^T + B_{jk} U_{i+1}^{(j)T} A_{jk}) - \beta_{i+1} V_i^{(k)} \right\|^2 \right)^{1/2}. \quad (16)$$

From above results, we can obtain the matrix form iteration method of LSMR algorithm for solving the constrained solution group of (1) and problem (2). When the unknown matrices $X_1, X_2, \dots, X_q \in SR^{n \times n}$, the matrix form iterative method is given as shown in Algorithm 2.

4. The Solution Group of Problem (3)

Now, we consider the solution group of the matrix nearness problem (3) for given matrix group $(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_q)$, where $\bar{X}_k \in \mathbb{R}^{n \times n}$, $k = 1, 2, \dots, q$. If $X_k \in SR^{n \times n}$, it is easy to prove that

$$\begin{aligned} \min_{X_k \in SR^{n \times n}, k=1,2,\dots,q} \left\{ \sum_{k=1}^q \|X_k - \bar{X}_k\|^2 \right\} \\ = \min_{X_k \in SR^{n \times n}, k=1,2,\dots,q} \left\{ \sum_{k=1}^q \left\| X_k - \frac{\bar{X}_k + \bar{X}_k^T}{2} \right\|^2 \right\} \\ + \left\{ \sum_{k=1}^q \left\| \frac{\bar{X}_k - \bar{X}_k^T}{2} \right\|^2 \right\}. \end{aligned} \quad (17)$$

Let

$$\widehat{X}_k = X_k - \frac{\bar{X}_k + \bar{X}_k^T}{2}, \quad k = 1, 2, \dots, q, \quad (18)$$

$$\widehat{C}_j = C_j - \sum_{k=1}^q A_{jk} \frac{\bar{X}_k + \bar{X}_k^T}{2} B_{jk}, \quad j = 1, 2, \dots, p;$$

then problem (3) is equivalent to finding the minimum Frobenius norm symmetric solution group or minimum Frobenius norm least-squares symmetric solution group of the following problems, respectively.

Compatible matrix equations are as follows:

$$\sum_{k=1}^q A_{ik} \widehat{X}_k B_{ik} = \widehat{C}_i, \quad i = 1, 2, \dots, p. \quad (19)$$

Set $X_0^{(k)} = 0, \quad k = 1, 2, \dots, q$
 $\beta_1 = \sqrt{2} \left(\sum_{j=1}^p \|C_j\|^2 \right)^{1/2}, U_1^{(j)} = C_j / \beta_1, j = 1, 2, \dots, p$
 $\alpha_1 = \left(\sum_{k=1}^q \left\| \sum_{i=1}^p A_{ik}^T U_1^{(i)} B_{ik}^T + B_{ik} U_1^{(i)T} A_{ik} \right\|^2 \right)^{1/2}$
 $V_1^{(k)} = \sum_{i=1}^p (A_{ik}^T U_1^{(i)} B_{ik}^T + B_{ik} U_1^{(i)T} A_{ik}) / \alpha_1, k = 1, 2, \dots, q$
Set $\bar{\alpha}_1 = \alpha_1, \bar{\zeta}_1 = \alpha_1 \beta_1, \rho_0 = 1, \bar{\rho}_0 = 1, \bar{c}_0 = 1, \bar{s}_0 = 0, h_1^{(k)} = V_1^{(k)}, \quad k = 1, 2, \dots, q, \bar{h}_0^{(k)} = 0, \quad k = 1, 2, \dots, q$
For $k = 1, 2, \dots$, until converges Do:
 $\beta_{k+1} = \sqrt{2} \left(\sum_{i=1}^p \left\| \sum_{j=1}^q A_{ij} V_k^{(j)} B_{ij} - \alpha_k U_k^{(i)} \right\|^2 \right)^{1/2}$
 $U_{k+1}^{(i)} = \sum_{j=1}^q A_{ij} V_k^{(j)} B_{ij} - \alpha_k U_k^{(i)} / \beta_{k+1}, i = 1, 2, \dots, p$
 $\alpha_{k+1} = \left(\sum_{j=1}^q \left\| \sum_{i=1}^p (A_{ij}^T U_{k+1}^{(i)} B_{ij}^T + B_{ij} U_{k+1}^{(i)T} A_{ij}) - \beta_{k+1} V_k^{(j)} \right\|^2 \right)^{1/2}$
 $V_{k+1}^{(i)} = \left(\sum_{j=1}^q (A_{ij}^T U_{k+1}^{(i)} B_{ij}^T + B_{ij} U_{k+1}^{(i)T} A_{ij}) - \beta_{k+1} V_k^{(j)} \right) / \alpha_{k+1}, i = 1, 2, \dots, q$
 $\rho_k = (\bar{\alpha}_k^2 + \beta_{k+1}^2)^{1/2}$
 $c_k = \bar{\alpha}_k / \rho_k$
 $s_k = \beta_{k+1} / \rho_k$
 $\theta_{k+1} = s_k \alpha_{k+1}$
 $\bar{\alpha}_{k+1} = c_k \alpha_{k+1}$
 $\bar{\theta}_k = \bar{s}_{k-1} \rho_k$
 $\bar{\rho}_k = ((\bar{c}_{k-1} \rho_k)^2 + \theta_{k+1}^2)^{1/2}$
 $\bar{c}_k = \bar{c}_{k-1} \rho_k / \bar{\rho}_k$
 $\bar{s}_k = \theta_{k+1} / \bar{\rho}_k$
 $\zeta_k = \bar{c}_k \bar{\zeta}_k$
 $\bar{\zeta}_{k+1} = -\bar{s}_k \bar{\zeta}_k$
 $\bar{h}_k^{(i)} = h_k^{(i)} - (\bar{\theta}_k \rho_k / (\rho_{k-1} \bar{\rho}_{k-1})) \bar{h}_{k-1}^{(i)}, i = 1, 2, \dots, q$
 $X_k^{(i)} = X_{k-1}^{(i)} + (\zeta_k / (\rho_k \bar{\rho}_k)) \bar{h}_k^{(i)}, i = 1, 2, \dots, q$
 $h_{k+1}^{(i)} = V_{k+1}^{(i)} - (\theta_{k+1} / \rho_k) h_k^{(i)}, i = 1, 2, \dots, q$
If $|\zeta_{k+1}|$ is small enough then stop
End Do.

ALGORITHM 2: LSMR_SR_M algorithm.

Least-squares problem is as follows:

$$\min \left(\sum_{i=1}^p \left\| \sum_{k=1}^q A_{ik} \widehat{X}_k B_{ik} - \widehat{C}_i \right\|^2 \right)^{1/2}. \quad (20)$$

By LSMR_SR_M method, we can get the minimum Frobenius norm symmetric solution group $(\widehat{X}_1^*, \widehat{X}_2^*, \dots, \widehat{X}_q^*)$ of (19) (or the minimum Frobenius norm least-squares symmetric solution group of problem (20)). Then, the optimal approximate solution group $(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_q)$ of problem (3) can be obtained; that is, $\bar{X}_k = \widehat{X}_k^* + (\bar{X}_k + \bar{X}_k^T) / 2$.

5. Numerical Examples

To compare the behavior of the proposed matrix method discussed in the previous section with the CGNE method [43] and the matrix LSQR iterative method (LSQR_M) [37], we present in this section numerical results for three examples. All the numerical computations are performed in MATLAB 7.

Example 1. Suppose that the matrices $A_{ij}, B_{ij}, C_j, i, j = 1, 2$, are given by the following matrices:

$$A_{11} = \begin{pmatrix} 8 & -1 & 2 & -3 & 4 \\ 5 & 0 & 2 & 2 & 1 \\ 0 & 3 & -7 & 2 & 4 \\ -2 & -1 & -1 & 2 & 4 \\ 4 & 1 & 1 & 2 & 1 \end{pmatrix},$$

$$B_{11} = \begin{pmatrix} 9 & -2 & -3 & 1 & 1 \\ 0 & 5 & 4 & -2 & -2 \\ 2 & 5 & -4 & -8 & 1 \\ 9 & 0 & 2 & 5 & 1 \\ -3 & -3 & 1 & 1 & 9 \end{pmatrix},$$

$$A_{12} = \begin{pmatrix} 4 & -3 & 4 & -4 & 1 \\ -2 & -2 & 6 & -4 & 4 \\ 9 & 6 & 0 & 5 & 1 \\ 5 & 4 & 5 & 3 & 3 \\ 9 & 1 & 7 & -5 & 9 \end{pmatrix},$$

$$\begin{aligned}
 B_{12} &= \begin{pmatrix} -2 & -1 & -2 & -3 & 2 \\ 1 & 5 & 4 & 9 & 6 \\ 9 & -3 & 1 & 1 & 1 \\ -5 & 2 & 5 & 4 & 1 \\ 4 & -8 & 2 & 2 & 1 \end{pmatrix}, \\
 A_{21} &= \begin{pmatrix} 1 & 2 & 3 & -5 & 2 \\ 4 & 6 & 1 & -3 & 9 \\ 4 & 4 & -2 & 2 & 3 \\ 2 & 2 & 8 & 1 & -9 \\ 3 & 1 & 4 & -3 & 8 \end{pmatrix}, \\
 B_{21} &= \begin{pmatrix} 9 & 2 & 3 & -1 & -1 \\ 3 & -1 & 8 & 8 & 5 \\ -1 & -2 & -8 & -5 & -2 \\ -1 & -1 & 2 & 7 & 3 \\ 1 & -1 & -8 & 8 & 2 \end{pmatrix}, \\
 A_{22} &= \begin{pmatrix} -9 & 2 & 3 & 2 & 1 \\ -3 & 3 & 9 & 4 & 3 \\ 8 & 2 & 6 & -8 & 8 \\ 1 & 2 & -9 & 4 & -2 \\ 2 & -1 & -2 & -2 & -3 \end{pmatrix}, \\
 B_{22} &= \begin{pmatrix} -1 & -2 & 9 & 8 & 9 \\ 1 & -3 & -3 & -3 & 9 \\ -1 & -3 & -9 & -5 & 9 \\ 1 & -9 & -6 & 5 & -1 \\ 2 & -7 & -4 & 5 & 8 \end{pmatrix}.
 \end{aligned} \tag{21}$$

The C_1 and C_2 matrices are chosen such that $X_1 = I_n$ and $X_2 = E_n$, where I_n and E_n are the $n \times n$ identity matrix and the $n \times n$ matrix whose entries are all one, respectively.

In Figure 1, we display the convergence curves of the function $\log_{10} \delta_k$, with

$$\delta_k = \max \left\{ \frac{\|R_1^{(k)}\|_F}{\|R_1^{(0)}\|_F}, \frac{\|R_2^{(k)}\|_F}{\|R_2^{(0)}\|_F} \right\}, \tag{22}$$

where $R_i^{(k)}$, $i, k = 1, 2$, is the residual matrix of the i th equation in k th iteration. The initial iterative matrices in all the iterative methods are chosen as zero matrices of suitable size. Figure 1 confirms that the proposed algorithm has faster convergence rate and higher accuracy than the CGNE method and similar behavior to the matrix LSQR iterative method.

Example 2. Suppose that the matrices A_{ij} , B_{ij} , C_j , $i, j = 1, 2$, are given by the following matrices:

$$\begin{aligned}
 A_{11} &= \text{tridiag}(-1, 6, -1), & B_{11} &= \text{tridiag}(1, 8, -1), \\
 A_{12} &= (0.1)I_n, & B_{12} &= \text{tridiag}(1, 0, 1), \\
 A_{21} &= (0.1)I_n, & B_{21} &= \text{tridiag}(-2, 1, -2), \\
 A_{22} &= \text{tridiag}(-1, -3, -1), & B_{22} &= \text{tridiag}(1, 6, 2).
 \end{aligned} \tag{23}$$

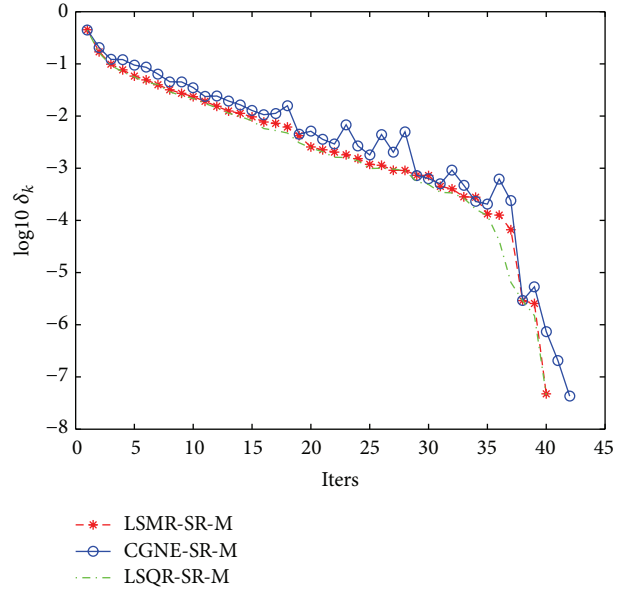


FIGURE 1: Convergence history of the LSMR, CGNE, and LSQR iterative methods for Example 1.

As Example 1, the C_1 and C_2 matrices are chosen such that $X_1 = I_n$ and $X_2 = E_n$ with $n = 400$ and the initial iterative matrices in all the iterative methods are chosen as zero matrices of suitable size. In Figure 2, as Figure 1, we display the convergence curves of the function $\log_{10} \delta_k$. This figure shows that the LSMR method outperforms the CGNE and LSQR methods.

Example 3 (see [45]). Consider the convection diffusion equation with the Dirichlet boundary conditions

$$\begin{aligned}
 L(u) &:= -\Delta u + 2\nu \frac{\partial u}{\partial x} + 2\nu \frac{\partial u}{\partial y} = f \quad \text{on } \Omega, \\
 u &= g \quad \text{on } \partial\Omega.
 \end{aligned} \tag{24}$$

Here Ω is the unit square $[0, 1] \times [0, 1]$. The operator L was discretized using central finite differences on Ω , with mesh size $h = 1/(n + 1)$ in the “ x ” direction and $k = 1/(p + 1)$ in the “ y ” direction. This yields a linear system of algebraic equations that can be written as a Sylvester matrix equation

$$AX - XB = C, \tag{25}$$

(as a particular case of (1) with $A_{11} = A$, $A_{12} = -I_n$, $B_{11} = I_p$, $B_{12} = B$, $C_1 = C$, and $X_1 = X_2 = X$) where tridiagonal matrices A and B are given by

$$\begin{aligned}
 A &= \frac{-1}{h^2} \text{tridiag}((1 + \nu h), -2, (1 - \nu h)), \\
 B &= \frac{1}{k^2} \text{tridiag}((1 + \nu k), -2, (1 - \nu k)).
 \end{aligned} \tag{26}$$

The right-hand side matrix C is obtained as follows:

$$\begin{aligned}
C(i, j) &= f(x_{i+1}, y_{j+1}), \quad \text{for } i = 2, \dots, n-1, \\
&\quad j = 2, \dots, p-1, \\
C(1, 1) &= f(x_2, y_2) + \frac{1+\nu h}{h^2} g(0, y_2) \\
&\quad + \frac{1+\nu k}{k^2} g(x_2, 0), \\
C(1, p) &= f(x_2, y_{p+1}) + \frac{1+\nu h}{h^2} g(0, y_{p+1}) \\
&\quad + \frac{1-\nu k}{k^2} g(x_2, 1), \\
C(n, 1) &= f(x_{n+1}, y_2) + \frac{1-\nu h}{h^2} g(1, y_2) \\
&\quad + \frac{1+\nu k}{k^2} g(x_{n+1}, 0), \\
C(n, p) &= f(x_{n+1}, y_{p-1}) + \frac{1-\nu h}{h^2} g(1, y_{p-1}) \\
&\quad + \frac{1-\nu k}{k^2} g(x_{n+1}, 1), \\
C(1, j) &= f(x_2, y_{j+1}) + \frac{1+\nu h}{h^2} g(0, y_{j+1}), \\
&\quad \text{for } j = 2, \dots, p-1, \\
C(n, j) &= f(x_{n+1}, y_{j+1}) + \frac{1-\nu h}{h^2} g(1, y_{j+1}), \\
&\quad \text{for } j = 2, \dots, p-1, \\
C(i, 1) &= f(x_{i+1}, y_2) + \frac{1+\nu k}{k^2} g(x_{i+1}, 0), \\
&\quad \text{for } i = 2, \dots, n-1, \\
C(i, p) &= f(x_{i+1}, y_{p+1}) + \frac{1-\nu k}{k^2} g(x_{i+1}, 1), \\
&\quad \text{for } i = 2, \dots, n-1.
\end{aligned} \tag{27}$$

In this example, the functions f and g were chosen such that the exact solution is

$$u(x, y) = xe^{-x^2-y^2} \tag{28}$$

on the domain Ω . In addition, we used the symmetric successive overrelaxation (SSOR) preconditioner for the matrix equation (25) to increase the convergence rate. It is easy to prove that the matrix equation (25) is equivalent to the $np \times np$ linear system:

$$\tilde{\mathcal{A}}\tilde{x} = \tilde{c}, \tag{29}$$

where $\tilde{\mathcal{A}} = I_p \otimes A - B^T \otimes I_n$, $\tilde{c} = \text{vec}(C)$, and $\tilde{x} = \text{vec}(X)$.

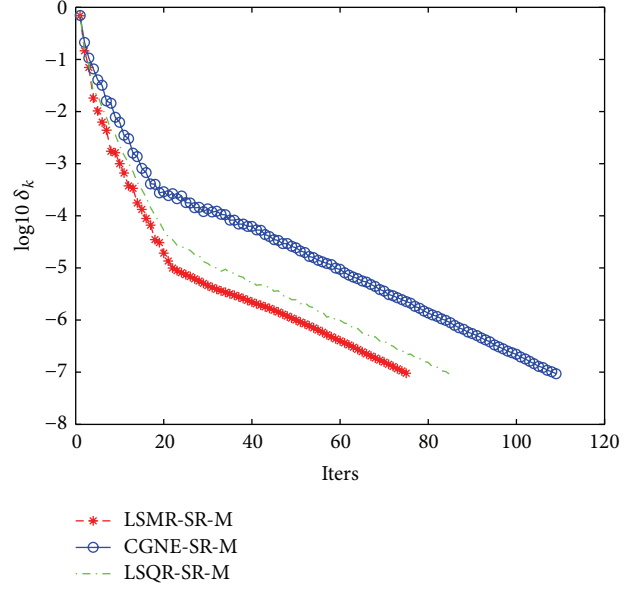


FIGURE 2: Convergence history of the LSMR, CGNE, and LSQR iterative methods for Example 2.

The matrices A and B can be written as

$$\begin{aligned}
A &= D_A - E_A - F_A, \\
B &= D_B - E_B - F_B,
\end{aligned} \tag{30}$$

where D_A is the diagonal of A and $-E_A$ and $-F_A$ are the strict lower and upper part of A , respectively. Then the splitting of the matrix $\tilde{\mathcal{A}}$ is given as

$$\tilde{\mathcal{A}} = \tilde{\mathcal{D}}_{\tilde{\mathcal{A}}} - \tilde{\mathcal{E}}_{\tilde{\mathcal{A}}} - \tilde{\mathcal{F}}_{\tilde{\mathcal{A}}}, \tag{31}$$

with

$$\begin{aligned}
\tilde{\mathcal{D}}_{\tilde{\mathcal{A}}} &= I_p \otimes D_A - D_B \otimes I_n, \\
\tilde{\mathcal{E}}_{\tilde{\mathcal{A}}} &= I_p \otimes E_A - F_B^T \otimes I_n, \\
\tilde{\mathcal{F}}_{\tilde{\mathcal{A}}} &= I_p \otimes F_A - E_B^T \otimes I_n.
\end{aligned} \tag{32}$$

Now instead of solving the matrix equation (25), we will apply the LSMR-M algorithm to the preconditioned system

$$\tilde{\mathcal{A}}\tilde{\mu}^{-1}\tilde{y} = \tilde{c} \quad \text{with } \tilde{y} = \tilde{\mu}\tilde{x}, \tag{33}$$

where $\tilde{\mu}$ is a preconditioner. As said, we use the SSOR preconditioner defined by

$$\tilde{\mu}_{\text{SSOR}} = \frac{1}{\omega(2-\omega)} \left(\tilde{\mathcal{D}}_{\tilde{\mathcal{A}}} - \omega\tilde{\mathcal{E}}_{\tilde{\mathcal{A}}} \right) \tilde{\mathcal{D}}_{\tilde{\mathcal{A}}}^{-1} \left(\tilde{\mathcal{D}}_{\tilde{\mathcal{A}}} - \omega\tilde{\mathcal{F}}_{\tilde{\mathcal{A}}} \right). \tag{34}$$

We note that the $np \times np$ matrix $\tilde{\mathcal{A}}$ is not used explicitly. We only use the action of the linear operator \mathcal{A} on a matrix $V \in \mathbb{R}^{np \times p}$, defined by $\mathcal{A}(V) = AV - VB$. In addition, we use only matrix-by-vector products; then when using the SSOR

preconditioner we have to compute, for a given $V \in \mathbb{R}^{n \times p}$, the matrix $W \in \mathbb{R}^{n \times p}$ such that

$$\tilde{w} = \tilde{\mathcal{A}}\tilde{\mu}_{\text{SSOR}}^{-1}\tilde{v} \quad \text{with } \tilde{w} = \text{vec}(W), \quad \tilde{v} = \text{vec}(V), \quad (35)$$

or

$$\begin{aligned} \tilde{w} &= (\tilde{\mathcal{A}}\tilde{\mu}_{\text{SSOR}}^{-1})^T \tilde{v} = (\tilde{\mu}_{\text{SSOR}}^T)^{-1} \tilde{\mathcal{A}}^T \tilde{v} \quad \text{with } \tilde{w} = \text{vec}(W), \\ & \tilde{v} = \text{vec}(V). \end{aligned} \quad (36)$$

With setting

$$\tilde{r} = \tilde{\mu}^{-1}\tilde{v} \iff \tilde{v} = \tilde{\mu}\tilde{r} \quad \text{with } \tilde{r} = \text{vec}(R) \quad (37)$$

the linear system (35) is equivalent to

$$\tilde{w} = \tilde{\mathcal{A}}\tilde{\mu}_{\text{SSOR}}^{-1}\tilde{v} \iff \tilde{w} = \tilde{\mathcal{A}}\tilde{r}. \quad (38)$$

For computing R such that $\tilde{w} = \tilde{\mathcal{A}}\tilde{\mu}_{\text{SSOR}}^{-1}\tilde{v}$, we have to solve the following matrix equations:

$$(D_A - \omega E_A)Y - Y(D_B - \omega F_B) = \omega(2 - \omega)V, \quad (39)$$

$$D_A Y - Y D_B = Z, \quad (40)$$

$$(D_A - \omega F_A)R - R(D_B - \omega E_B) = Z. \quad (41)$$

The matrix equations (39) and (41) are also Sylvester matrix equations. But as was stated in [45], since the matrices involved in these equations are triangular, they are solved easily. In (39), the matrix Y can be computed from left to right and from top to bottom in each column; this corresponds to backward substitution. Equation (41) is solved in the opposite sense and this corresponds to forward substitution. Now, to compute W in (35), it is sufficient to use the action of the operator \mathcal{A} on the matrix \tilde{r} , defined by $\mathcal{A}(\tilde{r}) = A\tilde{r} - \tilde{r}B$.

To compute W in (36), first, we use the action of the operator \mathcal{A}^T on the matrix, defined by $\mathcal{A}^T(\tilde{v}) = A^T\tilde{v} - \tilde{v}B^T$. Then, by setting

$$\tilde{\tilde{v}} = A^T\tilde{v} - \tilde{v}B^T, \quad (42)$$

the linear system (36) is equivalent to

$$\tilde{w} = (\tilde{\mu}_{\text{SSOR}}^T)^{-1}\tilde{\tilde{v}} \iff \tilde{\mu}_{\text{SSOR}}^T\tilde{w} = \tilde{\tilde{v}}. \quad (43)$$

Therefore, W can be obtained by solving the following matrix equations:

$$(D_A - \omega E_A^T)Y - Y(D_B - \omega F_B^T) = \omega(2 - \omega)\tilde{\tilde{v}}, \quad (44)$$

$$D_A Y - Y D_B = Z, \quad (45)$$

$$(D_A - \omega F_A^T)\tilde{w} - \tilde{w}(D_B - \omega E_B^T) = Z. \quad (46)$$

Similarly, the matrix equations (44) and (46) are also Sylvester matrix equations. But since the matrices involved in these equations are triangular, in (44), the matrix Y can be computed from right to left and from bottom to top in

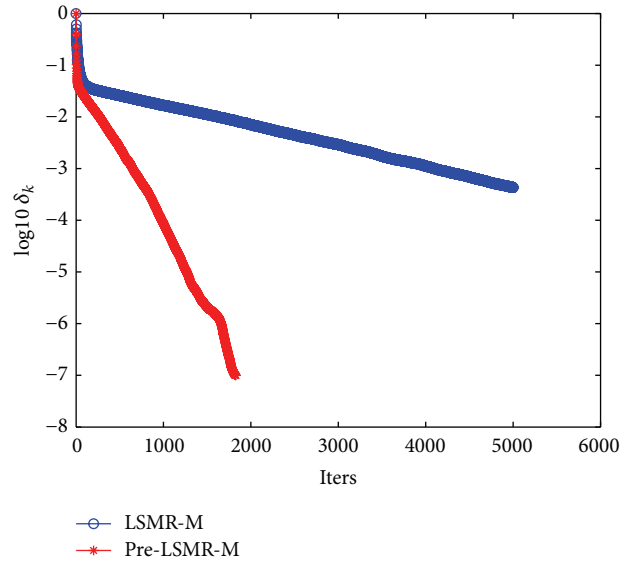


FIGURE 3: Convergence history of the LSMR-M and SSOR-LSMR-M.

each row; this corresponds to forward substitution. Equation (46) is solved in the opposite sense and this corresponds to backward substitution.

In Figure 3, we exhibited the function $\log_{10}\delta_k$ with

$$\delta_k = \frac{\|C - (AX_k - X_k B)\|_F}{\|C - (AX_0 - X_0 B)\|_F}, \quad (47)$$

versus the number of iterations for LSMR-M and the SSOR-LSMR-M. Furthermore, we note that for computing the quantity $\|R_k\|_F$, (R_k is the residual matrix in k th iteration) we used the pseudocode stated in [40]. These results were obtained for $\nu = 100$, $n = 300$, $p = 300$, and $\omega = 0.9$. The initial iterative matrix was chosen as zero matrix of suitable size. As we observe by using the SSOR preconditioner the convergence rate of the LSMR-M algorithm has increased, effectively.

6. Conclusion

Solving the linear matrix equations is an attractive part of research. By extending the idea of LSMR method, we have proposed Algorithm 2 to solve the coupled matrix equations (1) or the least-squares problem (2) over generalized symmetric matrices. By this new iterative method on the selection of special initial matrix group, we obtain the minimum Frobenius norm solutions or the minimum Frobenius norm least-squares solutions over generalized symmetric matrices. All the presented results show that the matrix LSMR iterative method is efficient to compute the solution group of the general coupled matrix equations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the referees for their valuable remarks and helpful suggestions.

References

- [1] T. Chen and B. A. Francis, *Optimal Sampled-Data Control Systems*, Springer, London, UK, 1995.
- [2] T. Chen and L. Qiu, " H_∞ design of general multirate sampled-data control systems," *Automatica*, vol. 30, no. 7, pp. 1139–1152, 1994.
- [3] L. P. LaSalle and S. Lefschetz, *Stability by Lyapunov's Direct Method*, Academic Press, New York, NY, USA, 1961.
- [4] Z. Y. Li, Y. Wang, B. Zhou, and G. R. Duan, "Least squares solution with the minimum-norm to general matrix equations via iteration," *Applied Mathematics and Computation*, vol. 215, no. 10, pp. 3547–3562, 2010.
- [5] B. C. Moore, "Principal component analysis in linear systems: controllability, observability, and model reduction," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.
- [6] L. Qiu and T. Chen, "Multirate sampled-data systems: all H_∞ suboptimal controllers and the minimum entropy controller," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 537–550, 1999.
- [7] F. Toutounian and S. Karimi, "Global least squares method (GLSQR) for solving general linear systems with several right-hand sides," *Applied Mathematics and Computation*, vol. 178, no. 2, pp. 452–460, 2006.
- [8] M. Dehghan and M. Hajarian, "An efficient algorithm for solving general coupled matrix equations and its application," *Mathematical and Computer Modelling*, vol. 51, no. 9-10, pp. 1118–1134, 2010.
- [9] F. Ding and T. Chen, "Gradient based iterative algorithms for solving a class of matrix equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.
- [10] F. Ding and T. Chen, "Iterative least-squares solutions of coupled Sylvester matrix equations," *Systems & Control Letters*, vol. 54, no. 2, pp. 95–107, 2005.
- [11] F. Ding and T. Chen, "On iterative solutions of general coupled matrix equations," *SIAM Journal on Control and Optimization*, vol. 44, no. 6, pp. 2269–2284, 2006.
- [12] G. X. Huang, N. Wu, F. Yin, Z. L. Zhou, and K. Guo, "Finite iterative algorithms for solving generalized coupled SYLvester systems—part I: one-sided and generalized coupled SYLvester matrix equations over generalized reflexive solutions," *Applied Mathematical Modelling*, vol. 36, no. 4, pp. 1589–1603, 2012.
- [13] I. Jonsson and B. Kågström, "Recursive blocked algorithm for solving triangular systems. I. One-sided and coupled Sylvester-type matrix equations," *Association for Computing Machinery. Transactions on Mathematical Software*, vol. 28, no. 4, pp. 392–415, 2002.
- [14] I. Jonsson and B. Kågström, "Recursive blocked algorithms for solving triangular systems-Part II: two-sided and generalized Sylvester and Lyapunov matrix equations," *ACM Transactions on Mathematical Software*, vol. 28, no. 4, pp. 416–435, 2002.
- [15] Z. H. Peng, X. Y. Hu, and L. Zhang, "The bisymmetric solutions of the matrix equation image and its optimal approximation," *Linear Algebra and Its Applications*, vol. 426, no. 2-3, pp. 583–595, 2007.
- [16] G. Starke and W. Niethammer, "SOR for $AX - XB = C$," *Linear Algebra and Its Applications*, vol. 154–156, no. 3, pp. 355–375, 1991.
- [17] C. Tsui, "New approach to robust observer design," *International Journal of Control*, vol. 47, no. 3, pp. 745–751, 1988.
- [18] F. Yin, G. Huang, and D. Chen, "Finite iterative algorithms for solving generalized coupled SYLvester systems, Part II: two-sided and generalized coupled SYLvester matrix equations over reflexive solutions," *Applied Mathematical Modelling*, vol. 36, no. 4, pp. 1604–1614, 2012.
- [19] B. Zhou, G. Duan, and Z. Li, "Gradient based iterative algorithm for solving coupled matrix equations," *Systems & Control Letters*, vol. 58, no. 5, pp. 327–333, 2009.
- [20] Z. Chen and L. Lu, "A gradient based iterative solutions for Sylvester tensor equations," *Mathematical Problems in Engineering*, vol. 2013, Article ID 819479, 7 pages, 2013.
- [21] D. Chen, F. Yin, and G. X. Huang, "An iterative algorithm for the generalized reflexive solution of the matrix equations $AXB = E$, $CXD = F$," *Journal of Applied Mathematics*, vol. 2012, Article ID 492951, 20 pages, 2012.
- [22] F. Ding, "Combined state and least squares parameter estimation algorithms for dynamic systems," *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 38, no. 1, pp. 403–412, 2014.
- [23] F. Ding, Y. Liu, and B. Bao, "Gradient-based and least-squares-based iterative estimation algorithms for multi-input multi-output systems," *Proceedings of the Institution of Mechanical Engineers. Part I. Journal of Systems and Control Engineering*, vol. 226, no. 1, pp. 43–55, 2012.
- [24] K. Wang, Z. Liu, and C. Xu, "A modified gradient based algorithm for solving matrix equations $AXB + CX^T D = F$," *Journal of Applied Mathematics*, vol. 2014, Article ID 954523, 6 pages, 2014.
- [25] H. Yin and H. Zhang, "Least squares based iterative algorithm for the coupled sylvester matrix equations," *Mathematical Problems in Engineering*, vol. 2014, Article ID 831321, 8 pages, 2014.
- [26] H. Zhang and F. Ding, "A property of the eigenvalues of the symmetric positive definite matrix and the iterative algorithm for coupled Sylvester matrix equations," *Journal of the Franklin Institute*, vol. 351, no. 1, pp. 340–357, 2014.
- [27] J. Zhou, R. Wang, and Q. Niu, "A preconditioned iteration method for solving Sylvester equations," *Journal of Applied Mathematics*, vol. 2012, Article ID 401059, 12 pages, 2012.
- [28] F. Ding and T. Chen, "Hierarchical gradient-based identification of multivariable discrete-time systems," *Automatica*, vol. 41, no. 2, pp. 315–325, 2005.
- [29] F. Ding and T. Chen, "Hierarchical least squares identification methods for multivariable systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 397–402, 2005.
- [30] F. Ding, X. Liu, H. Chen, and G. Yao, "Hierarchical gradient based and hierarchical least squares based iterative parameter identification for CARARMA systems," *Signal Processing*, vol. 97, pp. 31–39, 2014.
- [31] F. Ding, P. X. Liu, and J. Ding, "Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle," *Applied Mathematics and Computation*, vol. 197, no. 1, pp. 41–50, 2008.
- [32] Y. Liu, F. Ding, and Y. Shi, "An efficient hierarchical identification method for general dual-rate sampled-data systems," *Automatica*, vol. 50, no. 3, pp. 962–973, 2014.

- [33] B. Zhou and G. R. Duan, "On the generalized Sylvester mapping and matrix equations," *Systems & Control Letters*, vol. 57, no. 3, pp. 200–208, 2008.
- [34] M. Dehghan and M. Hajarian, "Efficient iterative method for solving the second-order Sylvester matrix equation $EVF^2 - AVF - CV = BW$," *IET Control Theory & Applications*, vol. 3, no. 10, pp. 1401–1408, 2009.
- [35] M. Dehghan and M. Hajarian, "The general coupled matrix equations over generalized bisymmetric matrices," *Linear Algebra and its Applications*, vol. 432, no. 6, pp. 1531–1552, 2010.
- [36] M. Hajarian and M. Dehghan, "The generalized centrosymmetric and least squares generalized centro-symmetric solutions of the matrix equation $AYB + CY^T D = E$," *Mathematical Methods in the Applied Sciences*, vol. 34, no. 13, pp. 1562–1579, 2011.
- [37] S. Li and T. Huang, "LSQR iterative method for generalized coupled Sylvester matrix equations," *Applied Mathematical Modelling*, vol. 36, no. 8, pp. 3545–3554, 2012.
- [38] M. Hajarian, "The generalized QMRCGSTAB algorithm for solving Sylvester-transpose matrix equations," *Applied Mathematics Letters*, vol. 26, no. 10, pp. 1013–1017, 2013.
- [39] Y. Lin and V. Simoncini, "Minimal residual methods for large scale Lyapunov equations," *Applied Numerical Mathematics*, vol. 72, pp. 52–71, 2013.
- [40] D. C. Fong and M. Saunders, "LSMR: an iterative algorithm for sparse least-squares problems," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2950–2971, 2011.
- [41] M. Dehghan and M. Hajarian, "An iterative algorithm for the reflexive solutions of the generalized coupled Sylvester matrix equations and its optimal approximation," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 571–588, 2008.
- [42] M. L. Liang, L. F. Dai, and S. F. Wang, "An iterative method for (R, S) -symmetric solution of matrix equation $AXB = C$," *Scientia Magna*, vol. 4, no. 3, pp. 60–70, 2008.
- [43] M. Dehghan and M. Hajarian, "An iterative method for solving the generalized coupled Sylvester matrix equations over generalized bisymmetric matrices," *Applied Mathematical Modelling*, vol. 34, no. 3, pp. 639–654, 2010.
- [44] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *Journal of the Society for Industrial and Applied Mathematics B: Numerical Analysis*, vol. 2, no. 2, pp. 205–224, 1965.
- [45] A. Bouhamidi and K. Jbilou, "A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications," *Applied Mathematics and Computation*, vol. 206, no. 2, pp. 687–694, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

