

An accelerated gradient based iterative algorithm for solving systems of coupled generalized Sylvester-transpose matrix equations

A. M. E. Bayoumi^a, M. A. Ramadan^b, M. Nili Ahmadabadi^{c,*}

^aDepartment of Mathematics, Faculty of Education, Ain Shams University, Cairo, Egypt.

^bDepartment of Mathematics, Faculty of Science, Menoufia University, Shebeen El- Koom, Egypt.

^cDepartment of Mathematics, Najafabad Branch, Islamic Azad University, Najafabad, Iran.

Received 6 July 2018; Revised 27 January 2019; Accepted 29 January 2019.

Communicated by Hamidreza Rahimi

Abstract. In this paper, an accelerated gradient based iterative algorithm for solving systems of coupled generalized Sylvester-transpose matrix equations is proposed. The convergence analysis of the algorithm is investigated. We show that the proposed algorithm converges to the exact solution for any initial value under certain assumptions. Finally, some numerical examples are given to demonstrate the behavior of the proposed method and to support the theoretical results of this paper.

© 2019 IAUCTB. All rights reserved.

Keywords: Coupled matrix equations, Frobenius norm, relaxation parameters, gradient algorithm, modified gradient, the accelerated gradient algorithm.

2010 AMS Subject Classification: 15-04.

1. Introduction

In this paper, we consider the system of coupled generalized Sylvester-transpose matrix equations

$$\sum_{i=1}^n A_{ij} X_i B_{ij} + C_{ij} X_i^T D_{ij} = F_j \quad j = 1 \dots, n, \quad (1)$$

*Corresponding author.

E-mail address: ame-bayoumi@yahoo.com (A. M. E. Bayoumi); mramadan@eun.eg (M. A. Ramadan); mneely59@hotmail.com (M. Nili Ahmadabadi)

where $A_{ij}, B_{ij}, C_{ij}, D_{ij}, F_j \in \mathbb{R}^{n \times n}$ are given matrices and $X_i \in \mathbb{R}^{n \times n}$ are the unknown matrices.

Numerical methods for solving matrix equations become interesting as soon as they play an important role in several areas such as control theory, stability theory, perturbation analysis and some other fields of applied and pure mathematics. In the matrix algebra field, matrix equations have attracted much attention from many researchers. Ding et al. [6] derived iterative solutions of the Sylvester matrix equation $AXB + CXD = F$ and linear matrix equation $AXB = F$ the. In [7], Huang et al. proposed an iterative method for solving the linear matrix equation $AXB = F$ over a skew-symmetric matrix. The least squares gradient based iterative algorithm and gradient based iterative algorithm for solving (coupled) matrix equations is a novel and efficient numerical algorithms were presented in Ding and Chen [3–5]. Also, Niu et al. [8] proposed a relaxed gradient based iterative algorithm for solving Sylvester equations $AX + XB = C$. The numerical experiments show that the convergence behavior of Niu et al. algorithm is better than Ding algorithm [2]. Moreover, Wang et al. [10] proposed a modified gradient based algorithm for solving the Sylvester equation $AX + XB = C$. Recently, by applying the hierarchical identification principle, an iterative algorithm was constructed in [12] for solving extended Sylvester-conjugate matrix equations. Ramadan et al. [9] proposed a relaxed gradient based iterative algorithm for solving extended Sylvester-conjugate matrix equations $AXB + C\bar{X}D = F$. Wu et al. [13] considered the matrix equation $X - A\bar{X}B = C$ which includes the Yakubovich-conjugate, Sylvester-conjugate matrix equations as special cases. Then, Wu et al. [11] proposed a finite iterative algorithm to solve more general complex matrix equations. In [14], Xie et al. presented an accelerated gradient based algorithm for solving the generalized Sylvester-transpose matrix equation $AXB + CX^T D = F$. In [1], Bayoumi and Ramadan presented an accelerated gradient based iterative algorithm for solving extended Sylvester-conjugate matrix equations.

2. Preliminaries

The following notations and lemma will be used to develop the proposed work. We use A^T and $tr(A)$ to denote the transpose and the trace of A , respectively. We denote the set of all $m \times n$ real matrices by $\mathbb{R}^{m \times n}$. The Frobenius norm of the matrix is denoted by $\|A\|^2 = tr(A^T A)$.

Lemma 2.1 [6] For matrix equation $AXB = F$, if A is a full column-rank matrix and B is a full row-rank matrix, then the iterative solution $X(k)$ given by the following gradient based iterative algorithm converges to the exact solution X (i.e., $\lim_{k \rightarrow \infty} X(k) = X$) for any initial value $X(0)$: $X(k) = X(k-1) + \mu A^T [F - AX(k-1)B] B^T$ for $0 < \mu < \frac{2}{\lambda_{max}[A^T A] \lambda_{max}[B^T B]}$.

3. The accelerated iterative algorithm for solving the matrix (1)

The basic idea is to regard (1) as two subsystems, and based on the least-squares optimization, the parameters of each subsystem are identified, respectively. In this way, we attain the iterative method. The details are shown as follows. Define the two matrices

$$M_{ij} = F_j - \sum_{k=1, k \neq i}^n A_{kj} X_k B_{kj} - \sum_{k=1}^n C_{kj} X_k^T D_{kj}, N_{ij} = F_j - \sum_{k=1}^n A_{kj} X_k B_{kj} - \sum_{k=1, k \neq i}^n C_{kj} X_k^T D_{kj}.$$

From (1), we get two fictitious subsystems

$$A_{ij}X_iB_{ij} = M_{ij} \quad , \quad C_{ij}X_i^T D_{ij} = N_{ij} \tag{2}$$

Now, for the two subsystems (2) we consider merit functions $L'_j = \frac{1}{2} \sum_{k=1}^n \|A_{kj}X_kB_{kj} - M_{kj}\|^2$ and $L''_j = \frac{1}{2} \sum_{k=1}^n \|C_{kj}X_kD_{kj} - N_{kj}\|^2$. We use the following properties of the matrix trace $tr(AXB) = tr(BAX) = tr(XBA)$ and $\frac{dtr(AX)}{dx} = \frac{dtr(X^T A^T)}{dx} = A^T$. Now, in order to identify the gradient, we have

$$\begin{aligned} \frac{\delta L'_j(X)}{\delta X_i} &= \frac{1}{2} \frac{\delta}{\delta X_i} \|A_{ij}X_iB_{ij} - M_{ij}\|^2 \delta L'_j \frac{X}{\delta X_i} \\ &= \frac{1}{2} \frac{dtr[(A_{ij}X_iB_{ij} - M_{ij})^T (A_{ij}X_iB_{ij} - M_{ij})]}{dX} \\ &= \frac{1}{2} \frac{dtr[(B_{ij}^T X_i^T A_{ij}^T - M_{ij}^T)(A_{ij}X_iB_{ij} - M_{ij})]}{dX} \\ &= \frac{1}{2} \frac{dtr[B_{ij}^T X_i^T A_{ij}^T A_{ij}X_iB_{ij} - B_{ij}^T X_i^T A_{ij}^T M_{ij} - M_{ij}^T A_{ij}X_iB_{ij} + M_{ij}^T M_{ij}]}{dX} \\ &= \frac{1}{2} \frac{dtr[B_{ij}^T X_i^T A_{ij}^T A_{ij}X_iB_{ij}]}{dX} - \frac{dtr[B_{ij}^T X_i^T A_{ij}^T M_{ij}]}{dX} \\ &= \frac{1}{2} \frac{dtr[X_i^T A_{ij}^T A_{ij}X_iB_{ij}B_{ij}^T]}{dX} - \frac{dtr[B_{ij}^T X_i^T A_{ij}^T M_{ij}]}{dX} \\ &= \frac{1}{2} (A_{ij}^T A_{ij}X_iB_{ij}B_{ij}^T + A_{ij}^T A_{ij}X_iB_{ij}B_{ij}^T) - A_{ij}^T M_{ij}B_{ij}^T \\ &= A_{ij}^T (A_{ij}X_iB_{ij} - M_{ij})B_{ij}^T A_{ij}^T (A_{ij}X_iB_{ij})B_{ij}^T \end{aligned}$$

and

$$\begin{aligned} \frac{\delta L''_j(X)}{\delta X_i} &= \frac{1}{2} \frac{\delta}{(\delta X_i)} \|C_{ij}X_i^T D_{ij} - N_{ij}\|^2 \frac{\delta L''_j(X)}{\delta X_i} \\ &= \frac{1}{2} \frac{dtr[(C_{ij}X_i^T D_{ij} - N_{ij})^T (C_{ij}X_i^T D_{ij} - N_{ij})]}{dX} \\ &= \frac{1}{2} \frac{dtr[(D_{ij}^T X_i C_{ij}^T - N_{ij}^T)(C_{ij}X_i^T D_{ij} - N_{ij})]}{dX} \\ &= \frac{1}{2} \frac{dtr[D_{ij}^T X_i C_{ij}^T C_{ij}X_i^T D_{ij} - D_{ij}^T X_i C_{ij}^T N_{ij} - N_{ij}^T C_{ij}X_i^T D_{ij} + N_{ij}^T N_{ij}]}{dX} \\ &= \frac{1}{2} \frac{dtr[D_{ij}^T X_i C_{ij}^T C_{ij}X_i^T D_{ij}]}{dX} - \frac{dtr[D_{ij}^T X_i C_{ij}^T N_{ij}]}{dX} \\ &= \frac{1}{2} \frac{dtr[C_{ij}^T C_{ij}X_i^T D_{ij}D_{ij}^T X_i]}{dX} - \frac{dtr[C_{ij}^T N_{ij}D_{ij}^T X_i]}{dX} \\ &= \frac{1}{2} (D_{ij}D_{ij}^T X_i C_{ij}^T C_{ij} + D_{ij}D_{ij}^T X_i C_{ij}^T C_{ij}) - D_{ij}N_{ij}^T C_{ij} = D_{ij}(C_{ij}X_i^T D_{ij} - N_{ij})^T C_{ij} \\ &= (D_{ij}D_{ij}^T X_i C_{ij}^T C_{ij}) - D_{ij}N_{ij}^T C_{ij} = D_{ij}(C_{ij}X_i^T D_{ij} - N_{ij})^T C_{ij} \end{aligned}$$

Now, we are in the position to present the accelerated gradient based iterative algorithm for solving systems of coupled generalized Sylvester-transpose matrix equations (1).

Algorithm 3.1

Step 1. Input matrices $A_{ij}, B_{ij}, C_{ij}, D_{ij}, F1_j \in R^{n \times n}$ and given any small positive numbers ϵ_i and ω_i . Choose initial matrices $X'_i(0)$ and $X''_i(0)$. Compute $X_i(0) = (1 - \omega_i)X'_i(0) + \omega_i X''_i(0)$. Set $k := 1$.

Step 2. if $\delta_{kj} = \left\| \sum_{i=1}^n A_{ij}X_i(k-1)B_{ij} + C_{ij}X_i^T(k-1)D_{ij} - F_j \right\| / \|F_j\| < \epsilon_j$, stop; otherwise go to step 3.

Step3. Update the sequence

$$X'_i(k) = X_i(k-1) + \omega_i \tau_i \sum_{j=1}^n A_{ij}^T (F_j - A_{ij}X_i(k-1)B_{ij} - C_{ij}X_i^T(k-1)D_{ij})B_{ij}^T$$

Step 4. Compute $X_i(k-1) = (1 - \omega_i)X'_i(k) + \omega_i X''_i(k-1)$.

Step 5. Update the sequence

$$X''_i(k) = X_i(k-1) + (1 - \omega_i) \tau_i \sum_{j=1}^n D_{ij} (F_j - A_{ij}X_i(k-1)B_{ij} + C_{ij}X_i^T(k-1)D_{ij})^T C_{ij}$$

Step 6. Compute $X_i(k) = (1 - \omega_i)X'_i(k) + \omega_i X''_i(k)$.

Step 7. Set $k := k + 1$ and return to step 2.

Step 8. End.

ω_i are relaxation parameters satisfying $0 < \omega_i < 1$, where they control the relative importance of two residual matrices and τ_i are the convergence factors.

4. convergence analysis

In this subsection, the theorem is stated and proved to investigate the convergence properties of the proposed algorithm.

Theorem 4.1 If the system of matrix equation (1) is consistent and has a unique solutions X_i^* and

$$0 < \tau_i < \min \left\{ \frac{2}{\omega_i \sum_{j=1}^n \|A_{ij}\|^2 \|B_{ij}\|^2}, \frac{2}{(1 - \omega_i) \sum_{j=1}^n \|C_{ij}\|^2 \|D_{ij}\|^2} \right\} \tag{3}$$

then the iterative sequence $\{X_i(k)\}$ generated by our algorithm converges to X_i^* that is $\lim_{k \rightarrow \infty} X_i(k) = X_i^*$;or the error $\tilde{X}_i(k) = X_i(k) - X_i^*$ converges to zero for any initial matrices $X_i(0)$.

Proof. First, we define the estimation error matrices as

$$\begin{aligned} \tilde{X}_i(k) &= X_i(k) - X_i^*, & \tilde{X}'_i(k) &= X'_i(k) - X_i^*, \\ \tilde{X}''_i(k) &= X''_i(k) - X_i^*, & \tilde{X}_i &= \tilde{X}_i(k) - X_i^* \end{aligned} \tag{4}$$

for $k = 1, 2, \dots$ and

$$\begin{aligned} \epsilon_i(k) &= \sum_{j=1}^n A_{ij} \tilde{X}_i(k-1) B_{ij}, & \eta_i(k) &= \sum_{j=1}^n C_{ij} \tilde{X}_i^T(k-1) D_{ij}, \\ \delta_i(k) &= \sum_{j=1}^n A_{ij} \tilde{X}_i(k-1) B_{ij}, & \zeta_i(k) &= \sum_{j=1}^n C_{ij} \tilde{X}_i^T(k-1) D_{ij} \end{aligned} \tag{5}$$

for $i = 1, 2, \dots$. Using the above error matrices (4), (5), our algorithm and the matrix equation (1), it is easy to get

$$\begin{aligned} \tilde{X}'_i &= \tilde{X}_i(k-1) + \omega_i \tau_i \sum_{j=1}^n A_{ij}^T \left(\sum_{j=1}^n A_{ij} X_i^* B_{ij} + C_{ij} X_i^{*T} D_{ij} - A_{ij} X_i(k-1) B_{ij} \right. \\ &\quad \left. - C_{ij} X_i^T(k-1) D_{ij} \right) B_{ij}^T \\ &= \tilde{X}_i(k-1) + \omega_i \tau_i \sum_{j=1}^n A_{ij}^T (-\epsilon_i(k) - \eta_i(k)) B_{ij}^T. \end{aligned} \tag{6}$$

Similarly to the above, we can write

$$\tilde{X}''_i = \tilde{X}_i(k-1) + (1 - \omega_i) \tau_i \sum_{j=1}^n D_{ij} (-\delta_i(k) - \zeta_i(k)) C_{ij} \tag{7}$$

Now, by taking the frobenius norm of both sides of (6) and (7), it follows that

$$\begin{aligned} \|\tilde{X}'_i(k)\|^2 &= tr(\tilde{X}_i'^T(k) \tilde{X}'_i(k)) \\ &= \|\tilde{X}_i(k-1)\|^2 - 2\omega_i \tau_i tr(B_{ij}^T \tilde{X}_i^T(k) A_{ij}^T (\epsilon_i(k) + \eta_i(k))) \\ &\quad + \omega_i^2 \tau_i^2 \left[\left\| \sum_{j=1}^n A_{ij}^T (-\epsilon_i(k) - \eta_i(k)) B_{ij}^T \right\|^2 \right] \\ &= \|\tilde{X}_i(k-1)\|^2 - 2\omega_i \tau_i tr(\epsilon_i^T(k) (\epsilon_i(k) + \eta_i(k))) \\ &\quad + \omega_i^2 \tau_i^2 \left[\sum_{j=1}^n \|A_{ij}\|^2 \|B_{ij}\|^2 \|\epsilon_i(k) + \eta_i(k)\|^2 \right] \\ &\leq \|\tilde{X}_i(k-1)\|^2 - [2\omega_i \tau_i - \omega_i^2 \tau_i^2 \sum_{j=1}^n \|A_{ij}\|^2 \|B_{ij}\|^2] \|\epsilon_i(k) + \eta_i(k)\|^2 \end{aligned}$$

Similarly,

$$\begin{aligned} \|\tilde{X}''_i(k)\|^2 &= tr(\tilde{X}_i''^T(k) \tilde{X}''_i(k)) \\ &= \|\tilde{X}_i(k-1)\|^2 - 2(1 - \omega_i) \tau_i tr(C_{ij} \tilde{X}_i^T(k-1) D_{ij} (\delta_i(k) + \zeta_i(k))^T) \\ &\quad + (1 - \omega_i)^2 \tau_i^2 \left[\left\| \sum_{j=1}^n D_{ij} (-\delta_i(k) - \zeta_i(k)) C_{ij} \right\|^2 \right] \\ &= \|\tilde{X}_i(k-1)\|^2 - 2(1 - \omega_i) \tau_i tr(\zeta_i(k) (\delta_i(k) + \zeta_i(k))^T) \\ &\quad + (1 - \omega_i)^2 \tau_i^2 \left[\sum_{j=1}^n D_{ij} (-\delta_i(k) - \zeta_i(k)) C_{ij} \right]^2 \\ &\leq \|\tilde{X}_i(k-1)\|^2 - [2(1 - \omega_i) \tau_i - (1 - \omega_i)^2 \tau_i^2 \sum_{j=1}^n \|C_{ij}\|^2 \|D_{ij}\|^2] \|\delta_i(k) + \zeta_i(k)\|^2 \end{aligned}$$

According to $\tilde{X}_i(k) = (1 - \omega_i)\tilde{X}_i'(k) + \omega_i\tilde{X}_i''(k)$, we have

$$\begin{aligned} \|\tilde{X}_i(k)\|^2 &= \|(1 - \omega_i)\tilde{X}_i'(k) + \omega_i\tilde{X}_i''(k)\|^2 \\ &\leq (1 - \omega_i)^2\|\tilde{X}_i'(k)\|^2 + \omega_i^2\|\tilde{X}_i''(k)\|^2 \\ &\leq (1 - \omega_i)^2\|\tilde{X}_i(k-1)\|^2 - (1 - \omega_i)^2[2\omega_i\tau_i - \omega_i^2\tau_i^2 \sum_{j=1}^n \|A_{ij}\|^2\|B_{ij}\|^2]\|\epsilon_i(k) + \eta_i(k)\|^2 \\ &\quad + \omega_i^2\|\tilde{X}_i(k-1)\|^2 - \omega_i^2[2(1 - \omega_i)\tau_i - (1 - \omega_i)^2\tau_i^2 \sum_{j=1}^n \|C_{ij}\|^2\|D_{ij}\|^2]\|\delta_i(k) + \zeta_i(k)\|^2 \\ &\leq \|\tilde{X}_i(k-1)\|^2 + \omega_i^2\|\tilde{X}_i(k-1)\|^2 \\ &\quad - (1 - \omega_i)^2[2\omega_i\tau_i - \omega_i^2\tau_i^2 \sum_{j=1}^n \|A_{ij}\|^2\|B_{ij}\|^2]\|\epsilon_i(k) + \eta_i(k)\|^2 \\ &\quad - \omega_i^2[2(1 - \omega_i)\tau_i - (1 - \omega_i)^2\tau_i^2 \sum_{j=1}^n \|C_{ij}\|^2\|D_{ij}\|^2]\|\delta_i(k) + \zeta_i(k)\|^2 \\ &\leq \|\tilde{X}_i(0)\|^2 + \omega_i^2 \sum_{s=1}^k \|\tilde{X}_i(k-s)\|^2 \\ &\quad - (1 - \omega_i)^2[2\omega_i\tau_i - \omega_i^2\tau_i^2 \sum_{j=1}^n \|A_{ij}\|^2\|B_{ij}\|^2] \sum_{s=1}^k \|\epsilon_i(k) + \eta_i(s)\|^2 \\ &\quad - \omega_i^2[2(1 - \omega_i)\tau_i - (1 - \omega_i)^2\tau_i^2 \sum_{j=1}^n \|C_{ij}\|^2\|D_{ij}\|^2] \sum_{s=1}^k \|\delta_i(s) + \zeta_i(s)\|^2 \end{aligned}$$

Thus, $\sum_{s=1}^k \|\tilde{X}_i(k-s)\|^2 < \infty$. For the necessary condition of the series convergence, we have $\lim_{k \rightarrow \infty} \tilde{X}_i(k) = 0$ when the convergence factor τ_i satisfies $0 < \tau_i < \frac{2}{\omega_i \sum_{j=1}^n \|A_{ij}\|^2\|B_{ij}\|^2}$ and $0 < \tau_i < \frac{2}{(1-\omega_i) \sum_{j=1}^n \|C_{ij}\|^2\|D_{ij}\|^2}$. Namely,

$$0 < \tau_i < \min \left\{ \frac{2}{\omega_i \sum_{j=1}^n \|A_{ij}\|^2\|B_{ij}\|^2}, \frac{2}{(1-\omega_i) \sum_{j=1}^n \|C_{ij}\|^2\|D_{ij}\|^2} \right\}.$$

We have

$$\sum_{s=1}^k \|\epsilon_i(s)\eta_i(s)\|^2 < \infty, \quad \sum_{s=1}^k \|\delta_i(s)\zeta_i(s)\|^2 < \infty.$$

It follows from the convergence of the series that $\lim_{k \rightarrow \infty} \sum_{s=1}^{\infty} \|\epsilon_i(s) + \eta_i(s)\|^2 = 0$ and $\lim_{k \rightarrow \infty} \sum_{s=1}^{\infty} \|\delta_i(s) + \zeta_i(s)\|^2 = 0$. This implies that $\lim_{k \rightarrow \infty} \epsilon_i(k) + \eta_i(k) = 0$ and $\lim_{k \rightarrow \infty} \delta_i(k) + \zeta_i(k) = 0$, i.e., $\sum_{i=1}^n A_{ij}\tilde{X}_i(k-1)B_{ij} + C_{ij}\tilde{X}_i^T(k-1)D_{ij} \rightarrow 0$, which give $\tilde{X}_i(k) \rightarrow 0$ as $k \rightarrow \infty$ and we get $\lim_{k \rightarrow \infty} X_i(k) = X_i^*$. This completes the proof of the theorem.

5. Numerical examples

In this section, we present two numerical examples to demonstrate the efficiency of our proposed iterative methods.

Example 5.1 In this example, we consider systems of coupled generalized Sylvester-transpose matrix equations of two unknowns in the following form

$$\begin{cases} A_{11}X_1B_{11} + A_{21}X_2B_{21} + C_{11}X_1^T D_{11} + C_{21}X_2^T D_{21} = F_1 \\ A_{12}X_1B_{12} + A_{22}X_2B_{22} + C_{12}X_1^T D_{12} + C_{22}X_2^T D_{22} = F_2 \end{cases} \tag{8}$$

Given

$$A_{11} = \begin{bmatrix} -1 & 2 \\ -2 & 3 \end{bmatrix}, B_{11} = \begin{bmatrix} 0 & -1 \\ 0.5 & 3 \end{bmatrix}, A_{21} = \begin{bmatrix} 1 & -3 \\ 0 & 2 \end{bmatrix}, B_{21} = \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}, C_{11} = \begin{bmatrix} 1 & -2 \\ 3 & 3 \end{bmatrix}, D_{11} = \begin{bmatrix} -2 & 0 \\ 1 & 2 \end{bmatrix}$$

$$C_{21} = \begin{bmatrix} 2.5 & -4 \\ -2 & 0 \end{bmatrix}, D_{21} = \begin{bmatrix} 1 & 0 \\ 3 & -1 \end{bmatrix}, A_{12} = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}, B_{12} = \begin{bmatrix} 4 & 1 \\ 0.5 & -2 \end{bmatrix}, A_{22} = \begin{bmatrix} -1 & 0 \\ 3 & 1 \end{bmatrix}, B_{22} = \begin{bmatrix} 0 & 2 \\ 2 & -2 \end{bmatrix}$$

$$C_{12} = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}, D_{12} = \begin{bmatrix} 0 & -2 \\ -1 & -1 \end{bmatrix}, C_{22} = \begin{bmatrix} 12 & 3 \\ 1 & 2 \end{bmatrix}, D_{22} = \begin{bmatrix} 1 & -3 \\ 2 & -1 \end{bmatrix}, F_1 = \begin{bmatrix} -26 & 35 \\ -21 & -25 \end{bmatrix}, F_2 = \begin{bmatrix} 19 & -29 \\ 3 & 11 \end{bmatrix}$$

This system of matrix equation has unique solution $X_1 = \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}$ and $X_2 = \begin{bmatrix} 2 & -2 \\ 0 & 3 \end{bmatrix}$

We apply Algorithm 3.1 to solve generalized Sylvester matrix equation (8). When the initial matrices are chosen as $X_1 = X_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$. According theorem 3.1 the generalized Sylvester matrix equation (8) is convergent for $0 < \tau_1 < 0.0082$ and $0 < \tau_2 < 0.0092$. While, the relaxation parameters ω is chosen as $\omega = 0.5$. We can see in Fig.1 that for $\tau_1 0.0082$ and $\tau_2 = 0.0092$, then the iteration stops at $k = 79$. Define the relative iterative error as

$$f(k) = \sqrt{\frac{\|X_1(k) - X_1\|^2 + \|X_2(k) - X_2\|^2}{\|X_1\|^2 + \|X_2\|^2}}$$

From Fig.1, it is clear that the error $f(k)$ is becoming smaller and approaches zero as iteration number k increases. This indicates that the proposed algorithm is effective and convergent.

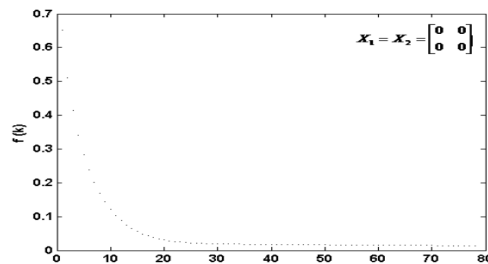


Figure 1. the convergence performance of Algorithm 3.1

Table 1. The convergence factors τ_1 and τ_2 versus the relaxation parameters ω and the number of iteration k .

convergence factor μ	the relaxation parameter ω	the number of iterations k
$\tau_1 = .0082, \tau_2 = .0092$.5	79

Example 5.2 In this example, we illustrate our theoretical results of Algorithm 3.1 for solving system of coupled generalized Sylvester-transpose matrix equations of two unknowns in the form (8) Given

$$A_{11} = \begin{bmatrix} -1 & 2 & -3 & 4 & 1 \\ -2 & 3 & 1 & -1 & 4 \\ 4 & 2 & 1 & -2 & 0 \\ 1 & 3 & -2 & 1 & 2 \\ 1 & 3 & -2 & 1 & 4 \end{bmatrix}, B_{11} = \begin{bmatrix} 0 & -1 & -1 & 4 & -2 \\ 1 & 3 & 2 & 1 & 1 \\ 3 & -2 & 1 & -4 & 2 \\ 0 & 2 & -4 & 1 & 3 \\ 1 & -3 & -2 & 0 & 4 \end{bmatrix}, A_{21} = \begin{bmatrix} 1 & -3 & 1 & 3 & 1 \\ -3 & 2 & 5 & 4 & 2 \\ 1 & -2 & 1 & 3 & 2 \\ -3 & 2 & 1 & 4 & 3 \\ 1 & -4 & 3 & 1 & -2 \end{bmatrix}$$

$$B_{21} = \begin{bmatrix} 2 & 3 & -1 & 2 & 1 \\ 1 & -2 & 3 & -1 & 3 \\ 3 & -2 & -1 & 1 & 0 \\ 2 & 4 & 1 & 3 & -1 \\ 1 & -1 & 4 & 1 & 1 \end{bmatrix}, C_{11} = \begin{bmatrix} 1 & -2 & 1 & 2 & 0 \\ 3 & 3 & 1 & -2 & 1 \\ 4 & 1 & 2 & 0 & -2 \\ 2 & 1 & 0 & 4 & 3 \\ 1 & -2 & 0 & 4 & 3 \end{bmatrix}, D_{11} = \begin{bmatrix} -2 & -1 & 1 & 4 & 2 \\ 1 & 2 & -1 & 3 & 3 \\ -1 & 2 & 3 & -2 & 2 \\ 1 & 4 & -2 & 1 & 4 \\ -1 & 1 & -1 & 2 & 3 \end{bmatrix}$$

$$C_{21} = \begin{bmatrix} 2 & 4 & 1 & -1 & 3 \\ -2 & 0 & 0 & 1 & 2 \\ -1 & 3 & 2 & -1 & 2 \\ 1 & 4 & -2 & -1 & 1 \\ 1 & 2 & -3 & 4 & 0 \end{bmatrix}, D_{21} = \begin{bmatrix} 1 & -2 & 2 & 3 & -1 \\ 3 & -1 & -1 & 1 & 2 \\ 1 & -1 & -3 & 4 & 0 \\ -3 & 1 & -2 & 3 & -2 \\ 1 & -1 & -2 & -3 & -2 \end{bmatrix}, A_{12} = \begin{bmatrix} 2 & 3 & -2 & -1 & 0 \\ 1 & 2 & 3 & -2 & -2 \\ 0 & -1 & -2 & 3 & -4 \\ -1 & 3 & 0 & 2 & -1 \\ -3 & 1 & 1 & -1 & 1 \end{bmatrix}$$

$$B_{12} = \begin{bmatrix} 0 & 2 & -1 & 3 \\ 3 & -2 & 1 & 1 & 4 \\ -1 & -1 & 1 & 0 & 3 \\ 0 & -3 & 0 & 2 & 1 \\ 1 & -1 & 2 & -4 & -1 \end{bmatrix}, A_{22} = \begin{bmatrix} -1 & 0 & 1 & 1 & 2 \\ 3 & 1 & 2 & 1 & 4 \\ -2 & 3 & -1 & -4 & 1 \\ 1 & 1 & 2 & 2 & 5 \\ 1 & -1 & -3 & -4 & 0 \end{bmatrix}, B_{22} = \begin{bmatrix} 0 & 0 & 1 & 0 & -1 \\ 2 & -1 & -2 & -1 & 3 \\ 1 & 0 & 2 & 0 & 1 \\ -1 & 2 & 3 & -1 & 0 \\ 0 & 1 & 2 & -1 & -3 \end{bmatrix}$$

$$C_{12} = \begin{bmatrix} 1 & 2 & -1 & 3 & 2 \\ 2 & 1 & 1 & -1 & 3 \\ 1 & 3 & -2 & -1 & 2 \\ 1 & 4 & 2 & -1 & 4 \\ 4 & 1 & 1 & 2 & 3 \end{bmatrix}, D_{12} = \begin{bmatrix} 0 & -1 & 1 & 3 & -1 \\ -1 & -1 & 4 & 1 & 3 \\ -1 & -2 & -3 & 1 & 1 \\ 0 & 0 & 4 & -3 & 1 \\ 3 & 1 & -2 & -3 & 1 \end{bmatrix}, C_{22} = \begin{bmatrix} 1 & 3 & -4 & 2 & 1 \\ 1 & 2 & -2 & -1 & 0 \\ -2 & 2 & 1 & -2 & -3 \\ 1 & 0 & -1 & -4 & -1 \\ 1 & 1 & -3 & 0 & 1 \end{bmatrix}$$

$$D_{22} = \begin{bmatrix} 1 & -1 & 0 & 2 & -1 \\ 2 & -1 & 3 & 4 & 1 \\ 1 & 3 & -2 & 0 & 1 \\ 1 & 2 & 3 & 2 & -1 \\ 1 & 1 & 2 & -2 & -4 \end{bmatrix}, F_1 = \begin{bmatrix} 2 & -14 & -58 & -12 & -22 \\ 2 & 22 & 4 & 8 & 53 \\ -35 & 13 & -66 & 10 & -2 \\ 59 & -12 & 16 & 42 & 158 \\ 33 & -17 & -89 & 95 & 79 \end{bmatrix}, F_2 = \begin{bmatrix} 11 & -98 & 9 & 144 & 39 \\ -1 & -35 & 42 & 15 & -59 \\ -51 & -70 & -65 & -12 & 205 \\ -44 & -33 & 43 & -25 & -96 \\ -3 & -34 & -46 & 48 & -28 \end{bmatrix}$$

This system of matrix equations has a unique solution

$$X_1 = \begin{bmatrix} 1 & 1 & 0 & 2 & 0 \\ -1 & 1 & -2 & 1 & 1 \\ 1 & 2 & -1 & 0 & 2 \\ 0 & 2 & 1 & 0 & 0 \\ -1 & -2 & -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad X_2 = \begin{bmatrix} 1 & -2 & -2 & 1 & 2 \\ 1 & 3 & -1 & 2 & 2 \\ 1 & -1 & -2 & 1 & -1 \\ 1 & -1 & 3 & -1 & 2 \\ 1 & -2 & -1 & -2 & 1 \end{bmatrix}$$

We apply Algorithm 3.1 to solve generalized Sylvester matrix equation (8). When the initial matrices are chosen as

$$X_1 = X_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

According Theorem 4.1 the generalized Sylvester matrix equation (8) is convergent for $0 < \tau_1 < 0.00045281$ and $0 < \tau_2 < 0.00051403$ meanwhile, the relaxation factor ω is set to be . We can see in Fig.2 that for $\tau_1 = 0.00045281$ and $\tau_2 = 0.00051403$, the iteration stops at $k = 3419$. Define the relative iterative error as

$$f(k) = \sqrt{\frac{\|X_1(k) - X_1\|^2 + \|X_2(k) - X_2\|^2}{\|X_1\|^2 + \|X_2\|^2}}$$

From Fig. 2, it is clear that the error $f(k)$ is becoming smaller and approaches zero as iteration number k increases. This indicates that the proposed algorithm is effective and convergent.

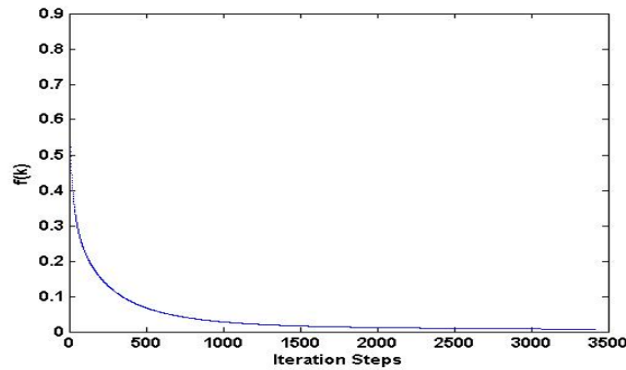


Figure 2. the convergence performance of Algorithm 3.1 for Example 5.2

Table 2. The convergence factors τ_1 and τ_2 versus the relaxation parameters ω and the number of iteration k .

convergence factor μ	the relaxation parameter ω	the number of iterations k
$\tau_1 = .00045281, \tau_2 = .0051403$.5	3419

6. Conclusions

In this study, we considered the accelerated gradient based iterative algorithm for solving systems of coupled generalized Sylvester-transpose matrix equations $\sum_{i=1}^n A_{ij}X_iB_{ij} + C_{ij}X_i^TD_{ij} = F_j$. A sufficient convergence condition for the proposed algorithm is presented. The iterative solution converges to the exact solution for any initial value. We presented four numerical examples, which demonstrate the superiority, efficiency and accuracy of the proposed method. We test the proposed algorithm using MATLAB and the results verify our theoretical findings.

Acknowledgments

The authors would like to thank professor Rahimi and anonymous referees for their valuable suggestions and recommendations.

References

- [1] A. M. E. Bayoumi, M. A. Ramadan, An accelerated gradient based iterative algorithm for solving extended Sylvester-conjugate matrix equations, *Trans. of Ins. Measur. Contr.* (2016), DOI: 10.1177/0142331216658092.
- [2] F. Ding, T. Chen, Gradient based iterative algorithms for solving a class of matrix equations, *IEEE Trans. Autom. Control.* 50 (2005), 1216-1221.
- [3] F. Ding, T. Chen, Hierarchical gradient-based identification of multivariable discrete-time systems, *Automatic.* 41 (2005), 315-325.
- [4] F. Ding, T. Chen, Hierarchical least squares identification methods for multivariable systems, *IEEE Trans. Autom. Control.* 50 (2005), 397-402.
- [5] F. Ding, T. Chen, On iterative solutions of general coupled matrix equations, *SIAM J. Control. Optim.* 44 (2006), 2269-2284.
- [6] F. Ding, P. X. Liu, J. Ding, Iterative solutions of the generalized Sylvester matrix equation by using the hierarchical identification principle, *Appl. Math. Comput.* 197 (2008), 41-50.
- [7] G. X. Huang, F. Yin, K. Guo, An iterative method for the skew-symmetric solution and the optimal approximate solution of the matrix equation $AXB = C$, *Appl. Math. Comput.* 212 (2008), 231-244.
- [8] Q. Niu, X. Wang, L. Z. Lu, A relaxed gradient based algorithm for solving Sylvester equations, *Asian J. of Control.* 13 (3) (2011), 461-464.
- [9] M. A. Ramadan, T. S. El-Danaf, A. M. E. Bayoumi, A relaxed gradient based algorithm for solving extended Sylvester-conjugate matrix equations, *Asian J. of Control.* 16 (5) (2014), 1-8.
- [10] X. Wang, L. Dai, D. Liao, A modified gradient based algorithm for solving Sylvester equations, *Appl. Math. Comput.* 218 (2012), 5620-5628.
- [11] A. G. Wu, G. Feng, G. R. Duan, Finite iterative solutions to a class of complex matrix equations with conjugate and transpose of the unknowns, *Math. Comput. Model.* 52 (9-10) (2010), 1463-1478.
- [12] A. G. Wu, X. Zeng, G. R. Duan, Iterative solutions to the extended Sylvester-conjugate matrix equation, *Appl. Math. Comput.* 217 (1) (2010), 130-142.
- [13] A. G. Wu, G. Feng, G. R. Duan, W. Liu, Iterative solutions to the Kalman-Yakubovich-conjugate matrix equation, *Appl. Math. Comput.* 217 (2011), 4427-4438.
- [14] Y. J. Xie, C. F. Ma, The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation, *Appl. Math. Comput.* 237 (2017), 1257-1269.