Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 125687, 6 pages
http://dx.doi.org/10.1155/2013/125687

Research Article

A New Method for the Bisymmetric Minimum Norm Solution of the Consistent Matrix Equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$

Aijing Liu,[1,2] Guoliang Chen,[1] and Xiangyun Zhang[1]

[1] *Department of Mathematics, East China Normal University, Shanghai 200241, China*
[2] *School of Mathematical Sciences, Qufu Normal University, Shandong 273165, China*

Correspondence should be addressed to Guoliang Chen; glchen@math.ecnu.edu.cn

Received 18 October 2012; Accepted 16 January 2013

Academic Editor: Alberto Cabada

We propose a new iterative method to find the bisymmetric minimum norm solution of a pair of consistent matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$. The algorithm can obtain the bisymmetric solution with minimum Frobenius norm in finite iteration steps in the absence of round-off errors. Our algorithm is faster and more stable than Algorithm 2.1 by Cai et al. (2010).

1. Introduction

Let $\mathbf{R}^{m\times n}$ denote the set of $m \times n$ real matrices. A matrix $X = (x_{ij}) \in \mathbf{R}^{n\times n}$ is said to be bisymmetric if $x_{ij} = x_{ji} = x_{n-i+1,n-j+1}$ for all $1 \le i, j \le n$. Let $\mathbf{BSR}^{n\times n}$ denote $n \times n$ real bisymmetric matrices. For any $X \in \mathbf{R}^{m\times n}$, X^T, $\mathrm{tr}(X)$, $\|X\|$, and $\|X\|_2$ represent the transpose, trace, Frobenius norm, and Euclidean norm of X, respectively. The symbol $\mathrm{vec}(\cdot)$ stands for the vec operator; that is, for $X = (x_1, x_2, \ldots, x_n) \in \mathbf{R}^{m\times n}$, where x_i $(i = 1, 2, \ldots, n)$ denotes the ith column of X, $\mathrm{vec}(X) = (x_1^T, x_2^T, \ldots, x_n^T)^T$. Let $\mathrm{mat}(\cdot)$ represent the inverse operation of vec operator. In the vector space $\mathbf{R}^{m\times n}$, we define the inner product as $\langle X, Y\rangle = \mathrm{tr}(Y^TX)$ for all $X, Y \in \mathbf{R}^{m\times n}$. Two matrices X and Y are said to be orthogonal if $\langle X, Y\rangle = 0$. Let $S_n = (e_n, e_{n-1}, \ldots, e_1)$ denote the $n\times n$ reverse unit matrix where e_i $(i = 1, 2, \ldots, n)$ is the ith column of $n\times n$ unit matrix I_n; then $S_n^T = S_n$, $S_n^2 = I_n$.

In this paper, we discuss the following consistent matrix equations:

$$A_1XB_1 = C_1, \quad A_2XB_2 = C_2, \quad X \in \mathbf{BSR}^{n\times n}, \tag{1}$$

where $A_1 \in \mathbf{R}^{p_1\times n}$, $B_1 \in \mathbf{R}^{n\times q_1}$, $C_1 \in \mathbf{R}^{p_1\times q_1}$, $A_2 \in \mathbf{R}^{p_2\times n}$, $B_2 \in \mathbf{R}^{n\times q_2}$, and $C_2 \in \mathbf{R}^{p_2\times q_2}$ are given matrices, and $X \in \mathbf{BSR}^{n\times n}$ is unknown bisymmetric matrix to be found.

Research on solving a pair of matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$ has been actively ongoing for the past 30 or more years (see details in [1–6]). Besides the works on finding the common solutions to the matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$, there are some valuable efforts on solving a pair of the matrix equations with certain linear constraints on solution. For instance, Khatri and Mitra [7] derived the Hermitian solution of the consistent matrix equations $AX = C$, $XB = D$. Deng et al. [8] studied the consistent conditions and the general expressions about the Hermitian solutions of the matrix equations $(AX, XB) = (C, D)$ and designed an iterative method for its Hermitian minimum norm solutions. Peng et al. [9] presented an iterative method to obtain the least squares reflexive solutions of the matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$. Cai et al. [10, 11] proposed iterative methods to solve the bisymmetric solutions of the matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$.

In this paper, we propose a new iterative algorithm to solve the bisymmetric solution with the minimum Frobenius norm of the consistent matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$, which is faster and more stable than Cai's algorithm (Algorithm 2.1) in [10].

The rest of the paper is organized as follows. In Section 2, we propose an iterative algorithm to obtain the bisymmetric minimum Frobenius norm solution of (1) and present some basic properties of the algorithm. Some numerical examples are given in Section 3 to show the efficiency of the proposed iterative method.

2. A New Iterative Algorithm

Firstly, we give the following lemmas.

Lemma 1 (see [12]). *There is a unique matrix $P(m,n) \in \mathbf{R}^{mn\times mn}$ such that* $\mathrm{vec}(X^T) = P(m,n)\,\mathrm{vec}\,(X)$ *for all* $X \in \mathbf{R}^{m\times n}$. *This matrix $P(m,n)$ depends only on the dimensions m and n. Moreover, $P(m,n)$ is a permutation matrix and* $P(n,m) = P(m,n)^T = P(m,n)^{-1}$.

Lemma 2. *If $y_0, y_1, y_2, \ldots \in \mathbf{R}^m$ are orthogonal to each other, then there exists a positive integer $\hat{l} \le m$ such that $y_{\hat{l}} = 0$.*

Proof. If there exists a positive integer $\hat{l} \le m-1$ such that $y_{\hat{l}} = 0$, then Lemma 2 is proved.

Otherwise, we have $y_i \ne 0$, $i = 0,1,2,\ldots,m-1$, and $y_0, y_1, \ldots, y_{m-1}$ are orthogonal to each other in the m-dimension vector space of $\mathbf{R}^m$. So $y_0, y_1, \ldots, y_{m-1}$ form a set of orthogonal basis of $\mathbf{R}^m$.

Hence y_m can be expressed by the linear combination of $y_0, y_1, \ldots, y_{m-1}$. Denote

$$y_m = a_0 y_0 + a_1 y_1 + \cdots + a_{m-1} y_{m-1} \tag{2}$$

in which $a_i \in \mathbf{R}$, $i = 0,1,2\ldots,m-1$. Then

$$\begin{aligned} \langle y_i, y_m \rangle &= a_0 \langle y_i, y_0 \rangle + a_1 \langle y_i, y_1 \rangle + \cdots + a_{m-1} \langle y_i, y_{m-1} \rangle \\ &= a_i \langle y_i, y_i \rangle + \sum_{\substack{j=1 \\ j\ne i}}^{m-1} a_j \langle y_i, y_j \rangle \\ &= a_i \langle y_i, y_i \rangle, \quad i = 0,1,2,\ldots,m-1. \end{aligned} \tag{3}$$

From $\langle y_i, y_m \rangle = 0$ and $\langle y_i, y_i \rangle \ne 0$, $i = 0,1,2,\ldots,m-1$, we have $a_i = 0, i = 0,1,2,\ldots,m-1$; that is,

$$y_m = 0. \tag{4}$$

This completes the proof. □

Lemma 3. *A matrix $X \in \mathbf{BSR}^{n\times n}$ if and only if $X = X^T = S_n X S_n$.*

Lemma 4. *If $Y \in \mathbf{R}^{n\times n}$, then $Y + Y^T + S_n(Y+Y^T)S_n \in \mathbf{BSR}^{n\times n}$.*

Next, we review the algorithm proposed by Paige [13] for solving the following consistent problem:

$$Mx = f, \tag{5}$$

with given $M \in \mathbf{R}^{s\times t}$, $f \in \mathbf{R}^s$.

Algorithm 5 (Paige algorithm). (i) Initialization

$$\begin{gathered} \tau_0 = 1; \qquad \xi_0 = -1; \qquad \theta_0 = 0; \\ z_0 = 0; \qquad w_0 = 0; \qquad \beta_1 u_1 = f; \qquad \alpha_1 v_1 = M^T u_1. \end{gathered} \tag{6}$$

(ii) Iteration. For $i = 1,2,\ldots$, until $\{x_i\}$ convergence, do

(a) $\xi_i = -\xi_{i-1}\beta_i/\alpha_i$; $z_i = z_{i-1} + \xi_i v_i$;

(b) $\theta_i = (\tau_{i-1} - \beta_i\theta_{i-1})/\alpha_i$; $w_i = w_{i-1} + \theta_i v_i$;

(c) $\beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i$;

(d) $\tau_i = -\tau_{i-1}\alpha_i/\beta_{i+1}$;

(e) $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i$;

(f) $\gamma_i = \beta_{i+1}\xi_i/(\beta_{i+1}\theta_i - \tau_i)$;

(g) $x_i = z_i - \gamma_i w_i$.

It is well known that if the consistent system of linear equations $Mx = f$ has a solution $x^* \in R(M^T)$, then x^* is the unique minimum Euclidean norm solution of $Mx = f$. It is obvious that x_i generated by Algorithm 5 belongs to $R(M^T)$ and this leads to the following result.

Theorem 6. *The solution generated by Algorithm 5 is the minimum Euclidean norm solution of* (5).

If $u_1, u_2, \ldots$ and $v_1, v_2, \ldots$ are generated by Algorithm 5, then $u_i^T u_j = \delta_{ij}$, $v_i^T v_j = \delta_{ij}$ (see details in [13]), in which

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \ne j. \end{cases} \tag{7}$$

If we denote

$$r_i = f - Mx_i, \tag{8}$$

where x_i is the approximation solution obtained by Algorithm 5 after the ith iteration, it follows that $r_i = -\beta_{i+1}\xi_i u_{i+1}$ (see details in [13]). So we have

$$r_i^T r_j = h_{ij}\delta_{ij} \tag{9}$$

in which $h_{ij} = \beta_{i+1}\beta_{j+1}\xi_i\xi_j$.

Now we derive our new algorithm, which is based on Paige algorithm.

Noting that X is the bisymmetric solution of (1) if and only if X is the bisymmetric solution of the following linear equations:

$$\begin{aligned} A_1 X B_1 &= C_1, & B_1^T X A_1^T &= C_1^T, \\ A_1 S_n X S_n B_1 &= C_1, & B_1^T S_n X S_n A_1^T &= C_1^T, \\ A_2 X B_2 &= C_2, & B_2^T X A_2^T &= C_2^T, \\ A_2 S_n X S_n B_2 &= C_2, & B_2^T S_n X S_n A_2^T &= C_2^T. \end{aligned} \tag{10}$$

Furthermore, suppose (10) is consistent; let Y be a solution of (10). If Y is a bisymmetric matrix, then Y is a bisymmetric solution of (1); otherwise we can obtain a bisymmetric solution of (10) by $X = (Y + Y^T + S_n\ (Y+Y^T)S_n)/4$.

The system of (10) can be transformed into (5) with coefficient matrix M and vector f as

$$M = \begin{pmatrix} B_1^T \otimes A_1 \\ A_1 \otimes B_1^T \\ B_1^T S_n \otimes A_1 S_n \\ A_1 S_n \otimes B_1^T S_n \\ B_2^T \otimes A_2 \\ A_2 \otimes B_2^T \\ B_2^T S_n \otimes A_2 S_n \\ A_2 S_n \otimes B_2^T S_n \end{pmatrix}, \qquad f = \begin{pmatrix} \operatorname{vec}(C_1) \\ \operatorname{vec}(C_1^T) \\ \operatorname{vec}(C_1) \\ \operatorname{vec}(C_1^T) \\ \operatorname{vec}(C_2) \\ \operatorname{vec}(C_2^T) \\ \operatorname{vec}(C_2) \\ \operatorname{vec}(C_2^T) \end{pmatrix}. \tag{11}$$

Therefore, $\beta_1 u_1 = f$, $\alpha_1 v_1 = M^T u_1$, $\beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i$, and $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i$ can be written as

$$\begin{aligned} \beta_1 u_1 &= \begin{pmatrix} \operatorname{vec}(C_1) \\ \operatorname{vec}(C_1^T) \\ \operatorname{vec}(C_1) \\ \operatorname{vec}(C_1^T) \\ \operatorname{vec}(C_2) \\ \operatorname{vec}(C_2^T) \\ \operatorname{vec}(C_2) \\ \operatorname{vec}(C_2^T) \end{pmatrix}, \\ \alpha_1 v_1 &= \begin{pmatrix} B_1^T \otimes A_1 \\ A_1 \otimes B_1^T \\ B_1^T S_n \otimes A_1 S_n \\ A_1 S_n \otimes B_1^T S_n \\ B_2^T \otimes A_2 \\ A_2 \otimes B_2^T \\ B_2^T S_n \otimes A_2 S_n \\ A_2 S_n \otimes B_2^T S_n \end{pmatrix}^T u_1, \\ \beta_{i+1} u_{i+1} &= \begin{pmatrix} B_1^T \otimes A_1 \\ A_1 \otimes B_1^T \\ B_1^T S_n \otimes A_1 S_n \\ A_1 S_n \otimes B_1^T S_n \\ B_2^T \otimes A_2 \\ A_2 \otimes B_2^T \\ B_2^T S_n \otimes A_2 S_n \\ A_2 S_n \otimes B_2^T S_n \end{pmatrix} v_i - \alpha_i u_i, \quad i = 1, 2, \ldots, \\ \alpha_{i+1} v_{i+1} &= \begin{pmatrix} B_1^T \otimes A_1 \\ A_1 \otimes B_1^T \\ B_1^T S_n \otimes A_1 S_n \\ A_1 S_n \otimes B_1^T S_n \\ B_2^T \otimes A_2 \\ A_2 \otimes B_2^T \\ B_2^T S_n \otimes A_2 S_n \\ A_2 S_n \otimes B_2^T S_n \end{pmatrix}^T u_{i+1} - \beta_{i+1} v_i, \quad i = 1, 2, \ldots. \end{aligned} \tag{12}$$

From (12), we have

$$u_i = \begin{pmatrix} \operatorname{vec}(U_{i1}) \\ \operatorname{vec}(U_{i1}^T) \\ \operatorname{vec}(U_{i1}) \\ \operatorname{vec}(U_{i1}^T) \\ \operatorname{vec}(U_{i2}) \\ \operatorname{vec}(U_{i2}^T) \\ \operatorname{vec}(U_{i2}) \\ \operatorname{vec}(U_{i2}^T) \end{pmatrix}, \qquad v_i = \operatorname{vec}(V_i), \tag{13}$$

where $U_{i1} \in \mathbf{R}^{p_1 \times q_1}$, $U_{i2} \in \mathbf{R}^{p_2 \times q_2}$, $V_i \in \mathbf{R}^{n \times n}$, and V_i is a bisymmetric matrix.

And so, the vector form 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 $\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i$ in Algorithm 5 can be rewritten as matrix form. Then we now propose the following matrix-form algorithm.

Algorithm 7. (i) Initialization

$\tau_0 = 1$; $\xi_0 = -1$; $\theta_0 = 0$; $Z_0 = 0\ (\in \mathbf{R}^{n \times n})$; $W_0 = Z_0$;

$\beta_1 = 2\sqrt{\|C_1\|^2 + \|C_2\|^2}$; $U_{1j} = C_j / \beta_1$, $j = 1, 2$;

$T_1 = A_1^T U_{11} B_1^T + A_2^T U_{12} B_2^T$; $\overline{V}_1 = T_1 + T_1^T + S_n (T_1 + T_1^T) S_n$; $\alpha_1 = \|\overline{V}_1\|$; $V_1 = \overline{V}_1 / \alpha_1$.

(ii) Iteration. For $i = 1, 2, \ldots$, until $\{X_i\}$ convergence, do

(a) $\xi_i = -\xi_{i-1} \beta_i / \alpha_i$; $Z_i = Z_{i-1} + \xi_i V_i$;

(b) $\theta_i = (\tau_{i-1} - \beta_i \theta_{i-1}) / \alpha_i$; $W_i = W_{i-1} + \theta_i V_i$;

(c) $\overline{U}_{i+1,j} = A_j V_i B_j - \alpha_i U_{ij}$, $j = 1, 2$;

$\beta_{i+1} = 2\sqrt{\|\overline{U}_{i+1,1}\|^2 + \|\overline{U}_{i+1,2}\|^2}$;

$U_{i+1,j} = \overline{U}_{i+1,j} / \beta_{i+1}$, $j = 1, 2$;

(d) $\tau_i = -\tau_{i-1} \alpha_i / \beta_{i+1}$;

(e) $T_{i+1} = A_1^T U_{i+1,1} B_1^T + A_2^T U_{i+1,2} B_2^T$;

$\overline{V}_{i+1} = T_{i+1} + T_{i+1}^T + S_n (T_{i+1} + T_{i+1}^T) S_n - \beta_{i+1} V_i$;

$\alpha_{i+1} = \|\overline{V}_{i+1}\|$; $V_{i+1} = \overline{V}_{i+1} / \alpha_{i+1}$;

(f) $\gamma_i = \beta_{i+1} \xi_i / (\beta_{i+1} \theta_i - \tau_i)$;

(g) $X_i = Z_i - \gamma_i W_i$.

Remark 8. The stopping criteria on Algorithm 7 can be used as

$$\begin{gathered} \|C_1 - A_1 X_i B_1\| + \|C_2 - A_2 X_i B_2\| \le \epsilon, \\ |\xi_i| \le \epsilon \quad \text{or} \quad \|X_i - X_{i-1}\| \le \epsilon, \end{gathered} \tag{14}$$

where ϵ is a small tolerance.

Remark 9. As V_i, Z_i, and W_i in Algorithm 7 are bisymmetric matrices, we can see that X_i obtained by Algorithm 7 are also bisymmetric matrices.

Some basic properties of Algorithm 7 are listed in the following theorems.

Theorem 10. *The solution generated by Algorithm 7 is the bisymmetric minimum Frobenius norm solution of* (1).

Theorem 11. *The iteration of Algorithm 7 will be terminated in at most* $p_1q_1 + p_2q_2$ *steps in the absence of round-off errors.*

Proof. By (8) and (11), we have by simple calculation that

$$r_i = \begin{pmatrix} \operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i1}^T) \\ \operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i1}^T) \\ \operatorname{vec}(R_{i2}) \\ \operatorname{vec}(R_{i2}^T) \\ \operatorname{vec}(R_{i2}) \\ \operatorname{vec}(R_{i2}^T) \end{pmatrix}, \tag{15}$$

in which $R_{i1} = C_1 - A_1X_iB_1$, and $R_{i2} = C_2 - A_2X_iB_2$, where X_i is the approximation solution obtained by Algorithm 7 after the ith iteration.

By Lemma 1, we have that

$$\begin{aligned} \operatorname{vec}(R_{i1}^T) &= P(p_1,q_1)\operatorname{vec}(R_{i1}), \\ \operatorname{vec}(R_{i2}^T) &= P(p_2,q_2)\operatorname{vec}(R_{i2}), \end{aligned} \tag{16}$$

where $P(p_1,q_1) \in \mathbf{R}^{p_1q_1\times p_1q_1}$ and $P(p_2,q_2) \in \mathbf{R}^{p_2q_2\times p_2q_2}$ are permutation matrices. For simplicity, we denote $P_1 = P(p_1,q_1)$, $P_2 = P(p_2,q_2)$. Then $P_1^TP_1 = I_{p_1q_1}$, $P_2^TP_2 = I_{p_2q_2}$.

Hence

$$\begin{aligned} r_i^T r_j &= \begin{pmatrix} \operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i1}^T) \\ \operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i1}^T) \\ \operatorname{vec}(R_{i2}) \\ \operatorname{vec}(R_{i2}^T) \\ \operatorname{vec}(R_{i2}) \\ \operatorname{vec}(R_{i2}^T) \end{pmatrix}^T \begin{pmatrix} \operatorname{vec}(R_{j1}) \\ \operatorname{vec}(R_{j1}^T) \\ \operatorname{vec}(R_{j1}) \\ \operatorname{vec}(R_{j1}^T) \\ \operatorname{vec}(R_{j2}) \\ \operatorname{vec}(R_{j2}^T) \\ \operatorname{vec}(R_{j2}) \\ \operatorname{vec}(R_{j2}^T) \end{pmatrix} \\ &= \begin{pmatrix} \operatorname{vec}(R_{i1}) \\ P_1\operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i1}) \\ P_1\operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i2}) \\ P_2\operatorname{vec}(R_{i2}) \\ \operatorname{vec}(R_{i2}) \\ P_2\operatorname{vec}(R_{i2}) \end{pmatrix}^T \begin{pmatrix} \operatorname{vec}(R_{j1}) \\ P_1\operatorname{vec}(R_{j1}) \\ \operatorname{vec}(R_{j1}) \\ P_1\operatorname{vec}(R_{j1}) \\ \operatorname{vec}(R_{j2}) \\ P_2\operatorname{vec}(R_{j2}) \\ \operatorname{vec}(R_{j2}) \\ P_2\operatorname{vec}(R_{j2}) \end{pmatrix} \\ &= 2(\operatorname{vec}(R_{i1}))^T\operatorname{vec}(R_{j1}) + 2(\operatorname{vec}(R_{i1}))^TP_1^TP_1\operatorname{vec}(R_{j1}) \\ &\quad + 2(\operatorname{vec}(R_{i2}))^T\operatorname{vec}(R_{j2}) \\ &\quad + 2(\operatorname{vec}(R_{i2}))^TP_2^TP_2\operatorname{vec}(R_{j2}) \\ &= 4\left((\operatorname{vec}(R_{i1}))^T\operatorname{vec}(R_{j1}) + (\operatorname{vec}(R_{i2}))^T\operatorname{vec}(R_{j2})\right) \\ &= 4\begin{pmatrix} \operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i2}) \end{pmatrix}^T \begin{pmatrix} \operatorname{vec}(R_{j1}) \\ \operatorname{vec}(R_{j2}) \end{pmatrix}. \end{aligned} \tag{17}$$

If we let $t_i = \begin{pmatrix} \operatorname{vec}(R_{i1}) \\ \operatorname{vec}(R_{i2}) \end{pmatrix} \in \mathbf{R}^{p_1q_1+p_2q_2}$, then we have by (9) that $t_0, t_1, t_2, \ldots$ are orthogonal to each other in $\mathbf{R}^{p_1q_1+p_2q_2}$. By Lemma 2, there exists a positive integer $\hat{l} \le (p_1q_1 + p_2q_2)$ such that $t_{\hat{l}} = 0$. Hence

$$R_{\hat{l}1} = R_{\hat{l}2} = 0, \tag{18}$$

that is, the iteration of Algorithm 7 will be terminated in at most $p_1q_1 + p_2q_2$ steps in the absence of round-off errors. □

3. Numerical Examples

In this section, we use some numerical examples to illustrate the efficiency of our algorithm. The computations are carried out at PC computer, with software MATLAB 7.0. The machine precision is around 10^{-16}.

We stop the iteration when $\|R_{i1}\| + \|R_{i2}\| \le 10^{-12}$.

Example 12. Given matrices A_1, B_1, C_1, A_2, B_2, and C_2 as follows:

$$A_1 = \begin{pmatrix} 1 & -4 & -2 & -1 & 0 & 1 & -3 \\ 3 & 1 & -1 & 3 & -1 & -2 & 1 \\ 4 & -3 & -3 & 2 & -1 & -1 & -2 \\ 2 & 5 & 1 & 4 & -1 & -3 & 4 \\ -1 & 4 & 2 & 1 & 0 & -1 & 3 \\ -3 & -1 & 1 & -3 & 1 & 2 & -1 \end{pmatrix},$$

$$B_1 = \begin{pmatrix} -3 & 2 & -1 & 3 & -2 & 1 \\ 2 & -3 & -1 & -2 & 3 & -4 \\ -1 & 1 & 0 & 1 & -1 & 1 \\ 0 & 1 & 1 & 0 & -1 & 2 \\ 1 & 2 & 3 & -1 & -2 & 5 \\ 3 & -3 & 0 & -3 & 3 & -3 \\ 0 & -1 & -1 & 0 & 1 & -2 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} -19 & 30 & 11 & 19 & -30 & 41 \\ -55 & 47 & -8 & 55 & -47 & 39 \\ -74 & 77 & 3 & 74 & -77 & 80 \\ -36 & 17 & -19 & 36 & -17 & -2 \\ 19 & -30 & -11 & -19 & 30 & -41 \\ 55 & -47 & 8 & -55 & 47 & -39 \end{pmatrix},$$

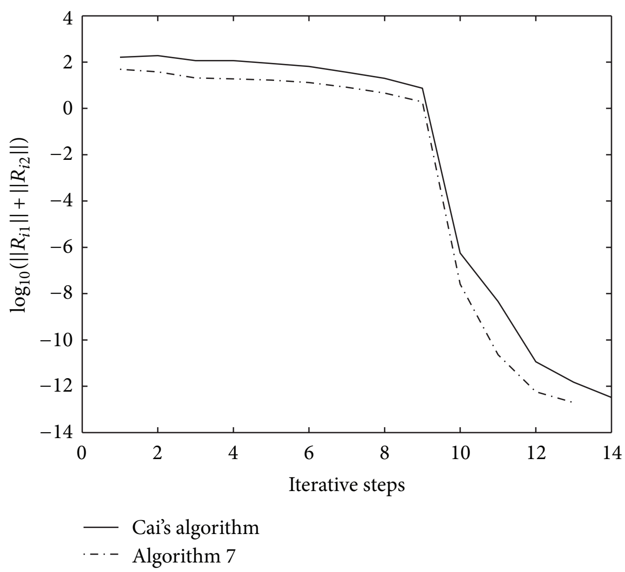

FIGURE 1: Convergence curves of $\log_{10}(\|R_{i1}\| + \|R_{i2}\|)$.

$$A_2 = \begin{pmatrix} 3 & -2 & -1 & 1 & -4 & 0 & -1 \\ 0 & -3 & 1 & -3 & 2 & 3 & 1 \\ -2 & -4 & 1 & -3 & 0 & 3 & 1 \\ 0 & 3 & -1 & 3 & -2 & -3 & -1 \\ 1 & -6 & 0 & -2 & -4 & 3 & 0 \end{pmatrix},$$
$$B_2 = \begin{pmatrix} 2 & 1 & 3 & -2 \\ -3 & -1 & -4 & 3 \\ 1 & 2 & 3 & -1 \\ 0 & 4 & 4 & 0 \\ -2 & 0 & -2 & 2 \\ 1 & -5 & -4 & -1 \\ -1 & -2 & -3 & 1 \end{pmatrix},$$
$$C_2 = \begin{pmatrix} 33 & 107 & 140 & -33 \\ 17 & -34 & -17 & -17 \\ 27 & -29 & -2 & -27 \\ -17 & 34 & 17 & 17 \\ 60 & 78 & 138 & -60 \end{pmatrix} \quad (19)$$

then (1) is consistent, for one can easily verify that it has a bisymmetric solution:

$$\widehat{X} = \begin{pmatrix} 1 & -1 & 1 & 2 & 1 & -1 & 1 \\ -1 & 3 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 0 & -2 & -1 & 1 & 1 \\ 2 & 1 & -2 & 1 & -2 & 1 & 2 \\ 1 & 1 & -1 & -2 & 0 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 3 & -1 \\ 1 & -1 & 1 & 2 & 1 & -1 & 1 \end{pmatrix}. \quad (20)$$

We choose the initial matrix $X_0 = 0$, then using Algorithm 7 and iterating 13 steps, we have the unique bisymmetric minimum Frobenius norm solution of (1) as follows:

$$X_{13} = \begin{pmatrix} 0.4755 & -0.6822 & 0.6274 & 1.4586 & 0.2774 & -1.2112 & -0.1053 \\ -0.6822 & 2.6628 & 0.4046 & 0.0716 & 1.0133 & 0.4001 & -1.2112 \\ 0.6274 & 0.4046 & -1.0215 & -2.2128 & -1.6176 & 1.0133 & 0.2774 \\ 1.4586 & 0.0716 & -2.2128 & -1.1548 & -2.2128 & 0.0716 & 1.4586 \\ 0.2774 & 1.0133 & -1.6176 & -2.2128 & -1.0215 & 0.4046 & 0.6274 \\ -1.2112 & 0.4001 & 1.0133 & 0.0716 & 0.4046 & 2.6628 & -0.6822 \\ -0.1053 & -1.2112 & 0.2774 & 1.4586 & 0.6274 & -0.6822 & 0.4755 \end{pmatrix}, \quad (21)$$

with $\|R_{13,1}\| + \|R_{13,2}\| = 6.2303e - 013$.

Figure 1 illustrates the performance of our algorithm and Cai's algorithm [10]. From Figure 1, we see that our algorithm is faster than Cai's algorithm.

Example 13. Let

$$A_1 = \text{hilb}(7), \qquad B_1 = \text{pascal}(7),$$
$$A_2 = \text{rand}(7,7), \qquad B_2 = \text{rand}(7,7), \quad (22)$$

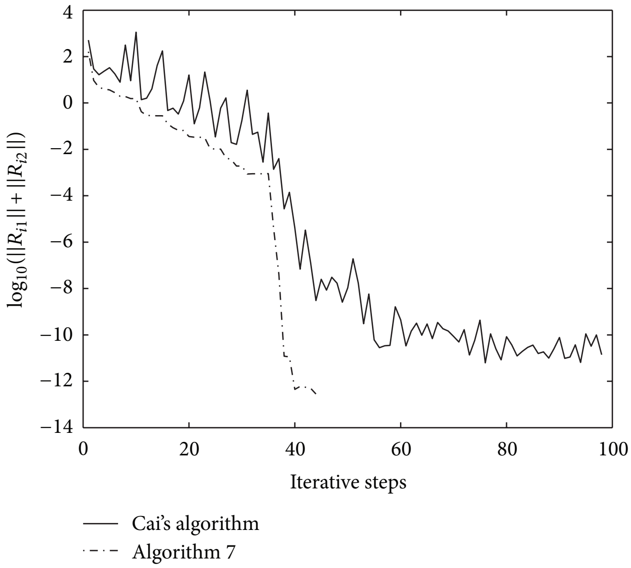

FIGURE 2: Convergence curves of $\log_{10}(\|R_{i1}\| + \|R_{i2}\|)$.

with hilb, pascal, and rand being functions in Matlab. And we let $C_1 = A_1\widehat{X}B_1$, $C_2 = A_2\widehat{X}B_2$, in which $\widehat{X}$ is defined in Example 12. Hence (1) is consistent.

We choose the initial matrix $X_0 = 0$; Figure 2 illustrates the performance of our algorithm and Cai's algorithm [10]. From Figure 2, we see that our algorithm is faster and more stable than Cai's algorithm.

Acknowledgments

This research was supported by the Natural Science Foundation of China (nos. 10901056, 11071079, and 11001167), the Shanghai Science and Technology Commission "Venus" Project (no. 11QA1402200), and the Natural Science Foundation of Zhejiang Province (no. Y6110043).

References

[1] X. P. Sheng and G. L. Chen, "A finite iterative method for solving a pair of linear matrix equations $(AXB, CXD) = (E, F)$," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1350–1358, 2007.

[2] Y. X. Yuan, "Least squares solutions of matrix equation $AXB = E$; $CXD = F$," *Journal of East China Shipbuilding Institute*, vol. 18, no. 3, pp. 29–31, 2004.

[3] A. Navarra, P. L. Odell, and D. M. Young, "A representation of the general common solution to the matrix equations $A_1XB_1 = C_1$ and $A_2XB_2 = C_2$ with applications," *Computers & Mathematics with Applications*, vol. 41, no. 7-8, pp. 929–935, 2001.

[4] S. K. Mitra, "A pair of simultaneous linear matrix equations and a matrix programming problem," *Linear Algebra and its Applications*, vol. 131, pp. 107–123, 1990.

[5] S. K. Mitra, "The matrix equations $AX = C, XB = D$," *Linear Algebra and its Applications*, vol. 59, pp. 171–181, 1984.

[6] P. Bhimasankaram, "Common solutions to the linear matrix equations $AX = C, XB = D$ and $FXG = H$," *Sankhyā A*, vol. 38, no. 4, pp. 404–409, 1976.

[7] C. G. Khatri and S. K. Mitra, "Hermitian and nonnegative definite solutions of linear matrix equations," *SIAM Journal on Applied Mathematics*, vol. 31, no. 4, pp. 579–585, 1976.

[8] Y.-B. Deng, Z.-Z. Bai, and Y.-H. Gao, "Iterative orthogonal direction methods for Hermitian minimum norm solutions of two consistent matrix equations," *Numerical Linear Algebra with Applications*, vol. 13, no. 10, pp. 801–823, 2006.

[9] Z.-H. Peng, X.-Y. Hu, and L. Zhang, "An efficient algorithm for the least-squares reflexive solution of the matrix equation $A_1XB_1 = C_1$," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 988–999, 2006.

[10] J. Cai, G. L. Chen, and Q. B. Liu, "An iterative method for the bisymmetric solutions of the consistent matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$," *International Journal of Computer Mathematics*, vol. 87, no. 12, pp. 2706–2715, 2010.

[11] J. Cai and G. L. Chen, "An iterative algorithm for the least squares bisymmetric solutions of the matrix equations $A_1XB_1 = C_1$," *Mathematical and Computer Modelling*, vol. 50, no. 7-8, pp. 1237–1244, 2009.

[12] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

[13] C. C. Paige, "Bidiagonalization of matrices and solutions of the linear equations," *SIAM Journal on Numerical Analysis*, vol. 11, pp. 197–209, 1974.