Numer Algor (2014) 66:379–397
DOI 10.1007/s11075-013-9740-9

ORIGINAL PAPER

Cyclic and simultaneous iterative methods to matrix equations of the form $A_i X B_i = F_i$

Yuchao Tang · Jigen Peng · Shigang Yue

Received: 5 March 2012 / Accepted: 3 July 2013 / Published online: 26 July 2013

Abstract This paper deals with a general type of linear matrix equation problem. It presents new iterative algorithms to solve the matrix equations of the form $A_i X B_i = F_i$. These algorithms are based on the incremental subgradient and the parallel subgradient methods. The convergence region of these algorithms are larger than other existing iterative algorithms. Finally, some experimental results are presented to show the efficiency of the proposed algorithms.

Keywords Linear matrix equations · Cyclic iterative algorithm · Simultaneous iterative algorithm

1 Introduction

Linear matrix equations (include Lyapunov equation, Sylvester equation, Riccati equation etc.,) often occur in areas of computational mathematics, image processing and control and system theory and so on. Iterative methods to solve these matrix equations is one of the active topics in the computational mathematics, and a large number of papers have raised various methods for solving such matrix equations. For a complete review on iterative methods for large linear matrix equations, see [1]. Recently, Ding and Chen [2–4] presented a few efficient gradient based and least squares based iterative algorithms for solving generalized Sylvester $AXB + CXD = F$ (including the Sylvester equation $AX + XB = F$) and coupled Sylvester equations. The basic idea of these approaches is based on a hierarchical identification

Y. Tang (✉) · J. Peng
Department of Mathematics, Xi'an Jiaotong University, Xi'an 710049, Peoples Republic of China
e-mail: hhaaoo1331@yahoo.com.cn

S. Yue
School of Computer Science, University of Lincoln, Lincoln LN6 7TS, UK

principle [5–8], which regards the unknown matrix as the system parameter matrix to be identified, and then construct a recursive formula to approximate the unknown solution. Some recent works can be found in [9–14] and references therein.

In this paper, we will consider the following linear systems of matrix equations:

$$\begin{cases} A_1XB_1 &= F_1 \\ A_2XB_2 &= F_2 \\ &\vdots \\ A_NXB_N &= F_N, \end{cases} \tag{1.1}$$

where $A_i \in \mathbb{R}^{p_i\times m}$, $B_i \in \mathbb{R}^{n\times q_i}$, $F_i \in \mathbb{R}^{p_i\times q_i}$, $i = 1, 2, \cdots, N$, N is a positive interger and $X \in \mathbb{R}^{m\times n}$ is the unknown matrix to be solved. As a matter of fact, the matrix equations (1.1) includes several important matrix equations as a special case. In particular, when $N = 1$, then (1.1) reduces to the linear matrix equation:

$$AXB = F, \tag{1.2}$$

where $A \in \mathbb{R}^{p\times m}$, $B \in \mathbb{R}^{n\times q}$, $F \in \mathbb{R}^{p\times q}$ and $X \in \mathbb{R}^{m\times n}$. Huang et al. [21] introduced an iterative method to solve the linear matrix equation (1.2) over skew-symmetric matrix X. Liang et al. [22] constructed an iterative algorithm to solve (1.2) over generalized centro-symmetric matrix X. Peng [23] presented two iterative methods to solve the matrix equation (1.2) over symmetric, symmetric R-symmetric and (R, S)-symmetric matrix X. By extending the well-known Jacobi and Gauss-Seidel iterations for $Ax = b$, Ding et al. [8] proposed a gradient based and a least-squares based iterative algorithms for the solution of (1.2) and the generalized Sylvester equations.

In the other hand, the matrix equation (1.2) also represents as an image deblurring problem under the assumption of the point spread function (PSF) is separable, and arbitrary boundary condition (see [15]). Where A and B represent blurring matrix, X is the original image and F is the recorded values of the noisy blurred image. If the matrix A and B are nonsingular, then the solution of (1.2) can be represented as

$$X = A^{-1}FB^{-1}.$$

However, since the observed image F is always be recorded by randomize noise, then the above native solution is not good for a real deblurring problem. Some well-known methods have been used to overcome this, such as, the truncated singular value decomposition (TSVD) and the Tikhonov regularization methods. We are dedicated to use the gradient projection algorithm (GPA, for short) to solve the matrix equation (1.2). It is equivalent to solve the following optimization problem:

$$\min_{X\in C} \|AXB - F\|_F, \tag{1.3}$$

where $\|\cdot\|_F$ denotes the Frobenious norm, and C is a closed convex set. Such as the box

$$C_1 = \{X \in \mathbb{R}^{n\times p} : L \le X \le U\}$$

and also

$$C_2 = \{X \in \mathbb{R}^{n\times p} : \|X\|_F \le \delta\},$$

where L and U are given matrices and $\delta > 0$ is a given scalar. If the set C is the set of symmetric matrices or skew-symmetric etc., then (1.3) reduces to the problem which was considered in Huang et al. [21], Liang et al. [22] and Peng [23], respectively.

If $N = 2$, then (1.1) reduces to the following:

$$\begin{cases} A_1 X B_1 = F_1 \\ A_2 X B_2 = F_2. \end{cases} \tag{1.4}$$

Wang [24] studied the matrix equations (1.4) over an arbitrary regular ring with identity and derived the necessary and sufficient conditions for the existence and the expression of the general solution to the system. Sheng and Chen [25] presented an iterative algorithm to solve (1.4) and also obtained the least norm solution and the optimal approximation solution. Cai and Chen [26] proposed an iterative algorithm to solve the matrix equations (1.4) over bisymmetric matrices. By choosing a special kind of initial matrix, they also obtained the unique optimal approximation solution. Dehghan and Hajarian [27] proposed some iterative algorithms to compute a generalized centro-symmetric solution of the linear matrix equations (1.4).

To solve the linear matrix equations (1.1), Ding et al. [16] developed a gradient algorithm and a least squares algorithm to solve the matrix equations (1.1) by using a block matrix inner product. First, the gradient iterative sequence $X(k)$ is given as follows:

$$X(k) = X(k-1) + \mu G_N^T \begin{pmatrix} F_1 - A_1 X(k-1) B_1 \\ F_2 - A_2 X(k-1) B_2 \\ \vdots \\ F_N - A_N X(k-1) B_N \end{pmatrix} \star H_N^T, , \tag{1.5}$$

where $0 < \mu < \frac{2}{\rho(G_N G_N^T)\rho(H_N^T H_N)}$, $G_N = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{pmatrix}$ and $H_N = \begin{pmatrix} B_1 & B_2 & \cdots & B_N \end{pmatrix}$.

Second, they introduced the least squares based iterative algorithm:

$$X(k) = X(k-1) + \mu \left(G_N^T G_N\right)^{-1} G_N^T \begin{pmatrix} F_1 - A_1 X(k-1) B_1 \\ F_2 - A_2 X(k-1) B_2 \\ \vdots \\ F_N - A_N X(k-1) B_N \end{pmatrix} \star H_N^T \left(H_N H_N^T\right)^{-1}, \tag{1.6}$$

where $0 < \mu < 2$.

In the numerical experiments, Ding et al. [16] found that the upper bound of the iterative parameter μ in (1.5) was not the best but was relatively conservative. They tried larger μ and found the iterative sequence also convergent, this should not be possible because it exceeded the convergence region of the algorithm (1.5). This discovery led to our interested in this research.

The purpose of this paper is to introduce new iterative algorithms to solve the matrix equations (1.1). We provide two types of algorithms, one is cyclic and the other is simultaneous, which are based on the incremental subgradient (see e.g.,

[17, 18]) and parallel subgradient methods (see e.g., [19, 20]), respectively. The key characteristic of our algorithm is that: (i) for the cyclic iterative algorithm, we only use one of the matrix equation in each iteration, while the simultaneous iterative algorithm is easier to implement in a distributed framework; (ii) both of these algorithms have the same range of parameters which are bigger in comparison with other iterative algorithms.

The paper will be organized as follows. In Section 2, we introduce our notations and provide some preliminary results. In Section 3, we propose a gradient projection iterative algorithm to solve the constrained matrix equations (1.3). In Section 4, we propose a cyclic iterative algorithm to solve the matrix equations (1.1) and we prove the convergence of this algorithm. In Section 5, we give a simultaneous iterative algorithm and prove its convergence. In Section 6, we give numerical examples to demonstrate the convergence results. Lastly, we make conclusion and give some recommendation for future work.

2 Preliminaries

In this section, we collect some important definitions and prove some useful lemmas which will be used in the following section.

We denote by $\mathbb{R}$ the set of real numbers. Let $\mathbb{R}^{m\times n}$ be the set of all $m \times n$ real matrices, $\mathbb{R}^m = \mathbb{R}^{m\times 1}$. Denoted by the superscripts T be the transpose. For matrices $A = (a_1, a_2, \cdots, a_n) \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times n}$, $a_i \in \mathbb{R}^m$, $\|A\|$ denotes the usual 2-norm, unless otherwise stated. $\rho(A)$ and $tr(A)$ represent its spectral radius and trace, respectively. Symbol $vec(\cdot)$ represents the vec operator, i.e., $vec(A) = \left(a_1^T, a_2^T, \cdots, a_n^T\right)^T$; $A \otimes B$ stands for the Kronecker product of matrices A and B; Moreover, $\langle A, B\rangle = tr(B^T A)$ is defined as the inner product of the two matrices, which generates the Frobenius norm, i.e., $\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{tr(A^T A)}$. Additionally, the 2-norm and Frobenius norm of A satisfy: $\|A\| \le \|A\|_F \le \sqrt{n}\|A\|$. We introduce the block-matrix inner product, i.e., the star ($\star$) product for short. Let

$$X := \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix} \in \mathbb{R}^{(mp)\times n}, \qquad Y := \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{pmatrix} \in \mathbb{R}^{(np)\times m},$$

where $X_i, Y_i^T \in \mathbb{R}^{m\times n}, i = 1, 2, \cdots, p$. Then the block matrix star product $\star$ is defined as

$$X \star Y = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix} \star \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{pmatrix} = \begin{pmatrix} X_1 Y_1 \\ X_2 Y_2 \\ \vdots \\ X_p Y_p \end{pmatrix}$$

The following two lemmas were obtained in [8].

Lemma 2.1 *([8]) For the matrix equation (1.2), if A is a full column-rank matrix and B is a full row-rank matrix ($p \geq m, n \leq q$), then the iterative solution $X(k)$ given by the following gradient based iterative algorithm converges to the exact solution X (i.e., $\lim_{k\to\infty} X(k) = X$) for any initial values $X(0)$:*

$$X(k) = X(k-1) + \mu A^T[F - AX(k-1)B]B^T,\ k \geq 0,$$

where $0 < \mu < \frac{2}{\rho(AA^T)\rho(B^TB)}$.

Lemma 2.2 *([8]) If the conditions of Lemma 2.1 hold, then the least squares based iterative algorithm:*

$$X(k) = X(k-1) + \mu(A^TA)^{-1}A^T[F - AX(k-1)B]B^T(BB^T)^{-1},\ 0 < \mu < 2.$$

yields $\lim_{k\to\infty} X(k) = X$.

Define

$$S := \begin{pmatrix} B_1^T \otimes A_1 \\ B_2^T \otimes A_2 \\ \vdots \\ B_N^T \otimes A_p \end{pmatrix} \in \mathbb{R}^{(\sum_{i=1}^N p_i q_i)\times(mn)},\ F = vec[F_1, F_2, \cdots, F_N]$$

The unique solution of matrix equations (1.1) is guaranteed by the following lemma.

Lemma 2.3 *([8, 16]) Matrix equations (1.1) has a unique solution if and only if $rank\{S, F\} = rank(S) = mn$; in this case, the unique solution is given by*

$$vec[x] = [S^TS]^{-1}S^TF, \tag{2.1}$$

and the corresponding homogeneous matrix equations $A_iXB_i = 0, i = 1, 2, \cdots, N$ has a unique solution: $X = 0$.

Lemma 2.4 *Let H be a Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and norm $\|\cdot\|$, respectively. Then*

(i) $\|x - y\|^2 = \|x\|^2 - 2\langle x, y\rangle + \|y\|^2$.

(ii) $\|\alpha x + (1-\alpha)y\|^2 = \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x - y\|^2,\ \forall x, y \in H \text{and} \forall t \in [0, 1]$.

The lemma above is well-known. We extend (ii) of Lemma 2.4 to more general.

Lemma 2.5 *Let H be a Hilbert space, then for all $x_1, x_2, \cdots, x_n \in H$,*

$$\left\|\sum_{i=1}^n \lambda_i x_i\right\|^2 = \sum_{i=1}^n \lambda_i\|x_i\|^2 - \sum_{i\neq j}^n \lambda_i\lambda_j\|x_i - x_j\|^2, n \geq 2, \tag{2.2}$$

where $\lambda_i \in [0, 1]$, for all $i = 1, 2, \cdots, n$ with $\sum_{i=1}^n \lambda_i = 1$.

The method of mathematical induction is applied to prove the above lemma.

Proof It is obvious that the equality (2.2) is satisfied when $n = 2$. Assume for $n = k$, $k \geq 2$, the equality (2.2) is true, we will prove is still correct for $n = k + 1$. In fact, we have

$$
\begin{aligned}
\left\|\sum_{i=1}^{k+1}\lambda_i x_i\right\|^2 &= \left\|\sum_{i=1}^{k}\lambda_i x_i + \lambda_{k+1}x_{k+1}\right\|^2 = \left\|\sum_{i=1}^{k}\lambda_i x_i + \left(1-\sum_{i=1}^{k}\lambda_i\right)x_{k+1}\right\|^2 \\
&= \left\|\sum_{i=1}^{k}\lambda_i\left(\frac{\sum_{i=1}^{k}\lambda_i x_i}{\sum_{i=1}^{k}\lambda_i}\right) + \left(1-\sum_{i=1}^{k}\lambda_i\right)x_{k+1}\right\|^2 \\
&= \sum_{i=1}^{k}\lambda_i\left\|\frac{\sum_{i=1}^{k}\lambda_i x_i}{\sum_{i=1}^{k}\lambda_i}\right\|^2 + \left(1-\sum_{i=1}^{k}\lambda_i\right)\|x_{k+1}\|^2 \\
&\quad - \sum_{i=1}^{k}\lambda_i\left(1-\sum_{i=1}^{k}\lambda_i\right)\left\|x_{k+1} - \frac{\sum_{i=1}^{k}\lambda_i x_i}{\sum_{i=1}^{k}\lambda_i}\right\|^2 \\
&= \sum_{i=1}^{k}\left\|\frac{\lambda_1}{\sum_{i=1}^{k}\lambda_i}x_1 + \frac{\lambda_2}{\sum_{i=1}^{k}\lambda_i}x_2 + \cdots + \frac{\lambda_k}{\sum_{i=1}^{k}\lambda_i}x_k\right\| + \lambda_{k+1}\|x_{k+1}\|^2 \\
&\quad - \sum_{i=1}^{k}\lambda_i\left(1-\sum_{i=1}^{k}\lambda_i\right)\left\|\frac{\lambda_1}{\sum_{i=1}^{k}\lambda_i}(x_{k+1}-x_1) + \cdots + \frac{\lambda_k}{\sum_{i=1}^{k}\lambda_i}(x_{k+1}-x_k)\right\|^2 \\
&= \sum_{i=1}^{k}\lambda_i\left(\frac{\lambda_1}{\sum_{i=1}^{k}\lambda_i}\|x_1\|^2 + \cdots + \frac{\lambda_k}{\sum_{i=1}^{k}\lambda_i}\|x_k\|^2 - \sum_{i\neq j}^{k}\frac{\lambda_i}{\sum_{i=1}^{k}\lambda_i}\frac{\lambda_j}{\sum_{i=1}^{k}\lambda_i}\|x_i - x_j\|^2\right) \\
&\quad + \lambda_{k+1}\|x_{k+1}\|^2 - \lambda_{k+1}\sum_{i=1}^{k}\lambda_i\left(\frac{\lambda_1}{\sum_{i=1}^{k}\lambda_i}\|x_{k+1}-x_1\|^2 + \cdots + \frac{\lambda_k}{\sum_{i=1}^{k}\lambda_i}\|x_{k+1}-x_k\|^2\right. \\
&\quad \left. - \sum_{i\neq j}^{k}\frac{\lambda_i}{\sum_{i=1}^{k}\lambda_i}\frac{\lambda_j}{\sum_{i=1}^{k}\lambda_i}\|x_i - x_j\|^2\right) \\
&= \lambda_1\|x_1\|^2 + \cdots + \lambda_k\|x_k\|^2 + \lambda_{k+1}\|x_{k+1}\|^2 \\
&\quad - \lambda_1\lambda_{k+1}\|x_{k+1} - x_1\|^2 - \cdots - \lambda_{k+1}\lambda_k\|x_{k+1} - x_k\|^2 \\
&\quad - \left(\sum_{i=1}^{k}\lambda_i - \sum_{i=1}^{k}\lambda_i\lambda_{k+1}\right)\sum_{i\neq j}^{k}\frac{\lambda_i}{\sum_{i=1}^{k}\lambda_i}\frac{\lambda_j}{\sum_{i=1}^{k}\lambda_i}\|x_i - x_j\|^2 \\
&= \lambda_1\|x_1\|^2 + \cdots + \lambda_k\|x_k\|^2 + \lambda_{k+1}\|x_{k+1}\|^2 \\
&\quad - \lambda_1\lambda_{k+1}\|x_{k+1} - x_1\|^2 - \cdots - \lambda_{k+1}\lambda_k\|x_{k+1} - x_k\|^2 - \sum_{i\neq j}^{k}\lambda_i\lambda_j\|x_i - x_j\|^2.
\end{aligned}
\tag{2.3}
$$

This completes the proof. □

Consider the following constrained convex optimization problem:

$$\min_{x\in C} f(x), \tag{2.4}$$

where C is a closed and convex subset of a Hilbert space H, and $f : C \to \mathbb{R}$ is a real valued convex function. If f is Fréchet differentiable, then the gradient projection algorithm is defined by

$$\begin{cases} x_0 \in C, \\ x_{n+1} = P_C(x_n - \alpha_n \nabla f(x_n)), n \geq 0, \end{cases} \tag{2.5}$$

where $\{\alpha_n\}$ are positive real numbers, P_C is the metric projection from H onto C. In fact, the gradient projection algorithm could be seen as a special case of the forward-backward splitting algorithm [32, 33].

The following convergence theorem of gradient projection algorithm can be found in [34] and [30].

Theorem 2.1 *Assume that the minimization problem (2.4) is consistent and the gradient ∇f satisfies the Lipschitz condition, i.e., $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, where L is the Lipschitz constant. Let the sequence of parameters $\{\alpha_n\}$ satisfy the condition: $0 < \liminf_{n\to\infty} \alpha_n \leq \limsup_{n\to\infty} \alpha_n < \frac{2}{L}$. Then the sequence $\{x_n\}$ generated by the gradient projection algorithm (2.5) or (2.6) converges weakly to a minimizer of (2.4).*

Remark 2.1 For the case of unconstrained minimization problem (2.4), i.e., $C = \mathbb{R}^n$, the iterative algorithm reduces to the gradient algorithm without projection operator which is given by the following:

$$\begin{cases} x_0 \in \mathbb{R}^n, \\ x_{n+1} = x_n - \alpha_n \nabla f(x_n), n \geq 0, \end{cases} \tag{2.6}$$

Then the convergence result of Theorem 2.1 still remains true but to the unconstrained minimization problem.

The following definition of Fejér-monotone sequence is central in the study of various iterative methods. For some other important properties can be found in Chapter 5 of [31].

Definition 2.1 Let C be a nonempty closed convex subset of H and $\{x_n\}$ is a sequence in H. The sequence $\{x_n\}$ is called Fejér-monotone with respect to C, if

$$\|x_{n+1} - z\| \leq \|x_n - z\|, \ n \geq 0, z \in C.$$

3 Gradient projection algorithm

In this section, we analysis the gradient projection algorithm to solve the constrained matrix equation (1.3). The convergence of gradient projection algorithm comply with Theorem 2.1.

Theorem 3.1 *Assume that the problem (1.3) is consistent (i.e., (1.3) is solvable). For any initial matrix $X_0 \in C$, a sequence $\{x_n\}$ be generated by the following:*

$$X_{n+1} = P_C(X_n - \alpha_n \nabla f(X_n)), n \geq 0, \tag{3.1}$$

where the sequences $\{\alpha_n\}$ satisfying the condition: $0 < \liminf_{n\to\infty} \alpha_n \leq \limsup_{n\to\infty} \alpha_n < \frac{2}{\rho(A^TA)\rho(BB^T)}$. Then the sequence $\{X_n\}$ converges to a minimizer of (1.3).

Proof It is sufficient to prove that the gradient of $f(X)$ is Lipschitz with Lipschitz constant $L = \rho(A^TA)\rho(BB^T)$. In fact, since $f(X) = \frac{1}{2}\|AXB - F\|_F^2 = \frac{1}{2}tr(AXB - F)^T(AXB - F)$, then $\nabla f(X) = A^T(AXB - F)B^T$. For any matrices $X, Y \in \mathbb{R}^{n\times n}$, we have

$$\begin{aligned}\|\nabla f(X) - \nabla f(Y)\| &= \|A^T(AXB - F)B^T - A^T(AYB - F)B^T\| \\ &\leq \|A^TA\|\|BB^T\|\|X - Y\| \\ &= \rho(A^TA)\rho(BB^T)\|X - Y\| \\ &= L\|X - Y\|.\end{aligned}$$

Therefore ∇f is a Lipschitz function and the Lipschitz constant $L = \rho(A^TA)\rho(BB^T)$. By Theorem 2.1, we can conclude the proof. □

4 Cyclic iterative method

We define the following objective function

$$\min F(X), \tag{4.1}$$

where $F(X) = \frac{1}{2}\sum_{i=1}^{N}\|A_iXB_i - F_i\|_F^2$. To find a solution of the matrix equations (1.1) is equivalent to solve the above minimization problem. In other words, if $\min\{F(X)\} = 0$, then the matrix equations (1.1) is solved naturally. It is easy to verify that $F(X)$ is convex, and the gradient of $F(X)$ is $\nabla F(X) = \sum_{i=1}^{N} A_i^T(A_iXB_i - F_i)B_i^T$. Then we can define the gradient iterative algorithm with constant stepsizes as follows:

$$X(k+1) = X(k) - \mu\nabla F(X(k)),\ k \geq 0, \tag{4.2}$$

where $0 < \mu < \frac{2}{L}$, $L = \sum_{i=1}^{N}\rho\left(A_iA_i^T\right)\rho\left(B_i^TB_i\right)$. With the help of Theorem 2.1, we know that the sequence $\{X(k)\}$ converges to the solution of matrix equations (1.1). We will propose new iterative algorithms to solve the matrix equations (1.1) as well as the convergence region of these algorithms which are larger than the iterative algorithms (4.2) and (1.5).

Let us introduce our first iterative algorithm.

Algorithm 1 Cyclic gradient iterative algorithm

For any initial matrix $X(0)$, compute

$$X(k) = X(k-1) + \mu A_{[k]}^T \left(F_{[k]} - A_{[k]} X(k-1) B_{[k]}\right) B_{[k]}^T, \ k \geq 0, \tag{4.3}$$

where $[k] = (k \ mod \ N)$ which takes value in $\{1, 2, \cdots, N\}$, and $\mu \in \left(0, \frac{2}{L}\right)$, where $L = \max\left\{\rho\left(A_i A_i^T\right)\rho\left(B_i^T B_i\right), i = 1, 2, \cdots, N\right\}$.

The following result discuss the convergence of the algorithm above.

Theorem 4.1 *If the matrix equations (1.1) is consistent and has a unique solution X, then the iterative sequences $\{X(k)\}$ generated by the Algorithm 1 converges to X, i.e., $\lim_{k\to\infty} X(k) = X$; or the error $X(k) - X$ converges to zero for any initial value $X(0)$ as $k \to \infty$.*

Proof To facilitate proof of theorem, we define the error matrix $\widetilde{X}(k) := X(k) - X$. By using (4.3), we get

$$\begin{aligned}
\widetilde{X}(k) &= X(k) - X \\
&= X(k-1) - X + \mu A_{[k]}^T \left(F_{[k]} - A_{[k]} X(k-1) B_{[k]}\right) B_{[k]}^T \\
&= \widetilde{X}(k-1) - \mu A_{[k]}^T \left(A_{[k]} \widetilde{X}(k-1) B_{[k]}\right) B_{[k]}^T.
\end{aligned} \tag{4.4}$$

With the help of Lemma 2.4 (i) and the formula $tr[AB] = tr[BA]$ and $tr[A^T] = tr[A]$, it follows that

$$\begin{aligned}
\left\|\widetilde{X}(k)\right\|^2 =& \left\| \widetilde{X}(k-1) - \mu A_{[k]}^T \left(A_{[k]} \widetilde{X}(k-1) B_{[k]}\right) B_{[k]}^T \right\|^2 \\
=& \left\|\widetilde{X}(k-1)\right\|^2 - 2\mu tr\left(\widetilde{X}^T(k-1) A_{[k]}^T \left(A_{[k]} \widetilde{X}(k-1) B_{[k]}\right) B_{[k]}^T\right) \\
&+ \mu^2 \left\| A_{[k]}^T (A_{[k]} \widetilde{X}(k-1) B_{[k]}) B_{[k]}^T \right\|^2 \\
=& \left\|\widetilde{X}(k-1)\right\|^2 - 2\mu \left\|A_{[k]} \widetilde{X}(k-1) B_{[k]}\right\|_F^2 + \mu^2 \left\| A_{[k]}^T (A_{[k]} \widetilde{X}(k-1) B_{[k]}) B_{[k]}^T \right\|^2 \\
\leq& \left\|\widetilde{X}(k-1)\right\|^2 - 2\mu \left\|A_{[k]} \widetilde{X}(k-1) B_{[k]}\right\|^2 + \mu^2 \rho\left(A_{[k]} A_{[k]}^T\right) \\
&\times \rho\left(B_{[k]}^T B_{[k]}\right) \left\|A_{[k]} \widetilde{X}(k-1) B_{[k]}\right\|^2 \\
=& \left\|\widetilde{X}(k-1)\right\|^2 - \mu\left(2 - \mu\rho\left(A_{[k]} A_{[k]}^T\right)\rho\left(B_{[k]}^T B_{[k]}\right)\right) \left\|A_{[k]} \widetilde{X}(k-1) B_{[k]}\right\|^2.
\end{aligned} \tag{4.5}$$

Since $0 < \mu < \frac{2}{L}$, it follows from (4.5) that

$$\left\|\widetilde{X}(k)\right\|^2 \leq \left\|\widetilde{X}(k-1)\right\|^2,$$

i.e., $\|X(k) - X\|^2 \leq \|X(k-1) - X\|^2$, which means that $\{X(k)\}$ is Fejér-monotone sequence.

On the other hand, from (4.5), we obtain

$$\sum_{k=1}^{\infty} \left\| A_{[k]} \widetilde{X}(k-1) B_{[k]} \right\|^2 < +\infty.$$

Therefore,

$$\lim_{k\to\infty} \left\| A_{[k]} \widetilde{X}(k-1) B_{[k]} \right\| = 0. \tag{4.6}$$

With the aid of (4.4) and (4.6), it is found that

$$\begin{aligned} \left\| \widetilde{X}(k) - \widetilde{X}(k-1) \right\|^2 &= \mu \left\| A_{[k]}^T \left(A_{[k]} \widetilde{X}(k-1) B_{[k]} \right) B_{[k]}^T \right\|^2 \\ &\leq \mu\rho \left(A_{[k]} A_{[k]}^T \right) \rho \left(B_{[k]}^T B_{[k]} \right) \left\| A_{[k]} \widetilde{X}(k-1) B_{[k]} \right\|^2 \\ &\to 0 \ \text{ as } \ k \to \infty. \end{aligned}$$

Then, for any $i \in \{1, 2, \cdots, N\}$, we get

$$\begin{aligned} &\left\| \widetilde{X}(k+i-1) - \widetilde{X}(k-1) \right\| \\ &\leq \left\| \widetilde{X}(k+i-1) - \widetilde{X}(k+i-2) \right\| + \cdots + \left\| \widetilde{X}(k) - \widetilde{X}(k-1) \right\| \\ &\to 0 \ \text{ as } \ k \to \infty. \end{aligned} \tag{4.7}$$

Combine the results of (4.6) and (4.7), we have

$$\begin{aligned} &\left\| A_{[k+i]} \widetilde{X}(k-1) B_{[k+1]} \right\| \\ &\leq \left\| A_{[k+i]} \widetilde{X}(k-1) B_{[k+i]} - A_{[k+i]} \widetilde{X}(k+i-1) B_{[k+i]} \right\| + \left\| A_{k+i} \widetilde{X}(k+i-1) B_{[k+i]} \right\| \\ &= \left\| A_{[k+i]} \left(\widetilde{X}(k+i-1) - \widetilde{X}(k-1) \right) B_{[k+i]} \right\| + \left\| A_{[k+i]} \widetilde{X}(k+i-1) B_{[k+i]} \right\| \\ &\to 0 \ \text{ as } \ k \to \infty. \end{aligned}$$

Therefor, for each $l \in \{1, 2, \cdots, N\}$, there exists $i \in \{1, 2, \cdots, N\}$ such that $l = (k+i) \ mod \ N$,

$$\lim_{k\to\infty} \left\| A_l \widetilde{X}(k-1) B_l \right\| = \lim_{k\to\infty} \left\| A_{[k+i]} \widetilde{X}(k-1) B_{[k+i]} \right\| = 0.$$

According to Lemma 2.3, we have $\widetilde{X}(k) \to 0$ as $k \to \infty$, that is $X(k) \to X$ as $k \to \infty$. This completes the proof. □

Using Lemma 2.2, in addition, if A_i is a non-square $p_i \times m$ full column-rank matrix and B_i is a non-square $n \times q_i$ full row-rank matrix, the least squares based iterative algorithm can be given as follows:

Algorithm 2 The least squares based cyclic gradient iterative algorithm

For any initial matrix $X(0)$, compute

$$X(k) = X(k-1) + \mu \left(A_{[k]}^T A_{[k]}\right)^{-1} A_{[k]}^T \left(F_{[k]} - A_{[k]} X(k-1) B_{[k]}\right) B_{[k]}^T \left(B_{[k]} B_{[k]}^T\right)^{-1}, k \geq 0, \tag{4.8}$$

where $[k] = (k \ mod \ N)$ which takes value in $\{1, 2, \cdots, N\}$ and $0 < \mu < 2$.

Theorem 4.2 *If the matrix equations (1.1) have a unique solution X, then the iterative solution $X(k)$ generated by the Algorithm 2 converges to X, i.e.,* $\lim_{k\to\infty} X(k) = X$.

The proof can be obtained by using a similar way to that above, and is therefore omitted here.

5 Simultaneous iterative method

In the previous section, we have proposed an cyclic iterative algorithm to solve the matrix equations (1.1) and the proof of convergence of the algorithm is also given. In this section, we will continue to develop some iterative algorithms to solve the matrix equations (1.1). Let's introduce a simultaneous iterative algorithm to solve (1.1).

Algorithm 3 Simultaneous gradient iterative algorithm

For any initial matrix $X(0)$. Given the current iterative $X(k-1)$, compute

$$\begin{aligned} X_1(k) &= X(k-1) + \mu A_1^T (F_1 - A_1 X(k-1) B_1) B_1^T \\ X_2(k) &= X(k-1) + \mu A_2^T (F_2 - A_2 X(k-1) B_2) B_2^T \\ &\vdots \\ X_N(k) &= X(k-1) + \mu A_N^T (F_N - A_N X(k-1) B_N) B_N^T. \end{aligned} \tag{5.1}$$

The iterate matrix $X(k)$ is added as

$$X(k) = \sum_{i=1}^{N} w_i X_i(k), \ k \geq 0,$$

where $w_i \in [0, 1]$ for all $i = 1, 2, \cdots, N$ with $\sum_{i=1}^{N} w_i = 1$ and $\mu \in \left(0, \frac{2}{L}\right)$, where $L = \max\left\{\rho\left(A_i A_i^T\right) \rho\left(B_i^T B_i\right), i = 1, 2, \cdots, N\right\}$.

Theorem 5.1 *If the matrix equations (1.1) is consistent and has a unique solution X, then the iterative sequences $\{X(k)\}$ generated by the Algorithm 3 converges to X, i.e.,* $\lim_{k\to\infty} X(k) = X$*; or the error $X(k) - X$ converges to zero for any initial value $X(0)$.*

Proof For the clarify of proof, we define the error matrices

$$\widetilde{X}_i(k) = X_i(k) - X, i = 1, 2, \cdots, N.$$
$$\widetilde{X}(k) = X(k) - X.$$

By using (5.1), it is easy to get

$$\begin{aligned}\widetilde{X}_1(k) &= X(k-1) - X + \mu A_1^T (F_1 - A_1 X(k-1) B_1) B_1^T \\ &= \widetilde{X}(k-1) - \mu A_1^T \left(A_1 \widetilde{X}(k-1) B_1\right) B_1^T.\end{aligned}$$

Similarly, we have

$$\widetilde{X}_i(k) = \widetilde{X}(k-1) - \mu A_i^T \left(A_i \widetilde{X}(k-1) B_i\right) B_i^T,\ i = 2, \cdots, N.$$

Using Lemma 2.4 (i) and the formula $tr[AB] = tr[BA]$ and $tr[A^T] = tr[A]$, we have

$$\begin{aligned}\left\|\widetilde{X}_i(k)\right\|^2 &= \left\|\widetilde{X}(k-1) - \mu A_i^T (A_i \widetilde{X}(k-1) B_i) B_i^T\right\|^2 \\ &= \left\|\widetilde{X}(k-1)\right\|^2 - 2\mu tr\left(\widetilde{X}^T(k-1) A_i^T (A_i \widetilde{X}(k-1) B_i) B_i^T\right) \\ &\quad + \mu^2 \left\|A_i^T (A_i \widetilde{X}(k-1) B_i) B_i^T\right\|^2 \\ &\le \left\|\widetilde{X}(k-1)\right\|^2 - 2\mu \left\|A_i \widetilde{X}(k-1) B_i\right\|^2 + \mu^2 \rho\left(A_i A_i^T\right) \\ &\quad \times \rho\left(B_i^T B_i\right) \left\|A_i \widetilde{X}(k-1) B_i\right\|^2, \end{aligned} \tag{5.2}$$

for all $i = 1, 2, \cdots, N$.

By Lemma 2.2 and (5.2), we have

$$\begin{aligned}\left\|\widetilde{X}(k)\right\|^2 &= \left\|\sum_{i=1}^N w_i (X_i(k) - X)\right\|^2 = \left\|\sum_{i=1}^N w_i \widetilde{X}_i(k)\right\|^2 \\ &\le \sum_{i=1}^N w_i \left\|\widetilde{X}_i(k)\right\|^2 \\ &\le \left\|\widetilde{X}(k-1)\right\|^2 - 2\mu \sum_{i=1}^N w_i \left\|A_i \widetilde{X}(k-1) B_i\right\|^2 \\ &\quad + \mu^2 \sum_{i=1}^N w_i \rho\left(A_i A_i^T\right) \rho\left(B_i^T B_i\right) \left\|A_i \widetilde{X}(k-1) B_i\right\|^2 \\ &\le \left\|\widetilde{X}(k-1)\right\|^2 - \sum_{i=1}^N \mu \left(2 - \mu \rho\left(A_i A_i^T\right) \rho\left(B_i^T B_i\right)\right) w_i \left\|A_i \widetilde{X}(k-1) B_i\right\|^2. \end{aligned} \tag{5.3}$$

Since $0 < \mu < \frac{2}{L}$, it follows from (5.3) that

$$\left\|\widetilde{X}(k)\right\|^2 \le \left\|\widetilde{X}(k-1)\right\|^2,$$

i.e.,

$$\|X(k) - X\|^2 \leq \|X(k-1) - X\|^2,$$

which means that $\{X(k)\}$ is Fejér-monotone sequence. From (5.3), we obtain

$$\sum_{k=1}^{\infty} \left\| A_i \widetilde{X}(k-1) B_i \right\|^2 < +\infty, \quad \text{for all } i = 1, 2, \cdots, N.$$

Therefore,

$$\lim_{k\to\infty} \|A_i \widetilde{X}(k) B_i\|^2 = 0, \quad \text{for all } i = 1, 2, \cdots, N.$$

According to Lemma 2.3, we have $\widetilde{X}(k) \to 0$ as $k \to \infty$. The proof is complete. □

Similarly, according to Lemma 2.2, we can obtain the least squares based simultaneous gradient iterative algorithm:

Algorithm 4 The least squares based simultaneous gradient iterative algorithm

For any initial matrix $X(0)$. Given the current iterative $X(k-1)$, compute

$$\begin{aligned}
X_1(k) &= X(k-1) + \mu \left(A_1^T A_1\right)^{-1} A_1^T \left(F_1 - A_1 X(k-1) B_1\right) B_1^T \left(B_1 B_1^T\right)^{-1} \\
X_2(k) &= X(k-1) + \mu \left(A_2^T A_2\right)^{-1} A_2^T \left(F_2 - A_2 X(k-1) B_2\right) B_2^T \left(B_2 B_2^T\right)^{-1} \\
&\vdots \\
X_N(k) &= X(k-1) + \mu \left(A_N^T A_N\right)^{-1} A_N^T (F_N - A_N X(k-1) B_N) B_N^T \left(B_N B_N^T\right)^{-1}.
\end{aligned} \tag{5.4}$$

The iterate matrix $X(k)$ is added as

$$X(k) = \sum_{i=1}^{N} w_i X_i(k), \ k \geq 0,$$

where $w_i \in [0, 1]$ for all $i = 1, 2, \cdots, N$ with $\sum_{i=1}^{N} w_i = 1$ and $0 < \mu < 2$.

Remark 5.1 There is no difficult to present the Algorithm 1 and the Algorithm 4.1 with variable stepsizes if ask

$$0 < \liminf_{k\to\infty} \mu_k \leq \limsup_{k\to\infty} \mu_k < \frac{2}{L},$$

where $L = \max\left\{\rho\left(A_i A_i^T\right) \rho\left(B_i^T B_i\right), \ i = 1, 2, \cdots, N\right\}$. For the convenience of numerical calculation, we choose it as constant.

6 Numerical examples

In this section, we give some examples to support the theoretical results obtained before. All experiments are performed using MATLAB (R2009a) on a Dell Optiplex 780 with an Intel Core2 Quad CPU with 4GB of RAM.

First, we use the gradient projection algorithm to solve the image deblurring problem (1.2).

Example 6.1 We consider the image deblurring model (1.2), and the image deblurring problem is taken from Challenge 2 of [15]. The data is also taken from the the web-site of the book [15].

Since the gradient projection algorithm has the semi-convergence, we stop the iterations before convergence to get a clear deblurred image. The reconstructed image is shown on Fig. 1.

On the first row of Fig. 1, the left is the blurred and noisy image, the middle reconstructed image using the Tikhonov regularization and the right restored image is obtained by (TSVD). On the second row, the left of the reconstructed image is obtained by the (GPA) without constraint set. The middle is obtained by using nonnegative constraint set of (GPA), and the right reconstructed image is obtained by adding a bounded interval [0, 1] as a constraint of (GPA).

An advantage of the gradient projection algorithm is that it can add the prior constraint to the algorithm. We tried to choose the domain C by (i) The pixels of the image should be nonnegative; A zero matrix is chosen as the lower bound matrix, i.e., $C = \{X \in \mathbb{R}^{n\times p} : X \geq \mathbf{0}\}$; (ii) Considering the blurred image G,

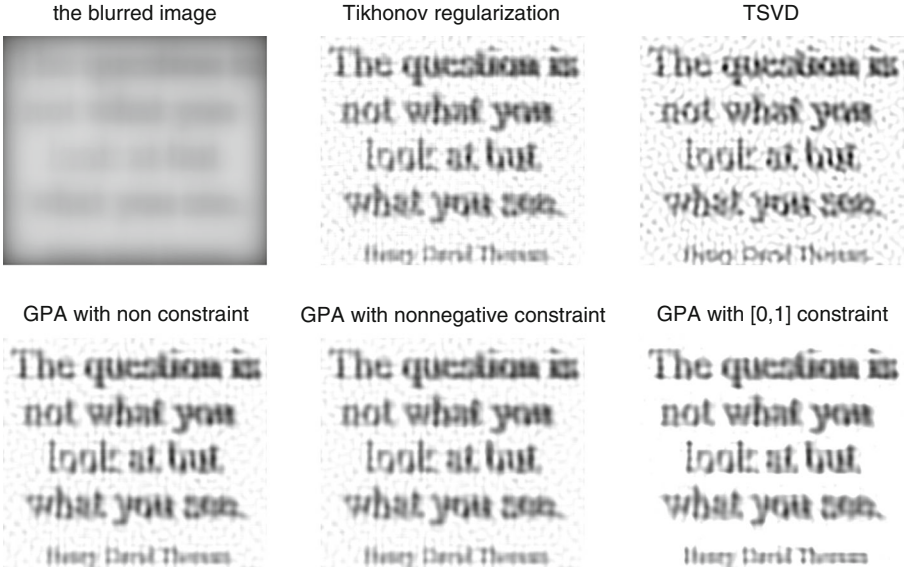

Fig. 1 Comparison of TSVD, Tikhonov regularization and Gradient projection algorithm for image deblurring

which the maximum pixel value is less than 1, then we assume the constraint set $C = \{X \in \mathbb{R}^{n\times p} : \mathbf{0} \le X \le \mathbf{1}\}$, where $\mathbf{1}$ represents matrix elements that are all equal to 1. We can observe that the reconstructed image obtained by (GPA) with [0, 1] constraint shows a better performance on the image background than other proposed methods.

The following test example is taken from [16] for ease of comparison.

Example 6.2 Consider the following coupled matrix equations:

$$\begin{cases} A_1 X B_1 = F_1 \\ A_2 X B_2 = F_2. \end{cases} \tag{6.1}$$

with

$$A_1 = \begin{pmatrix} 1 & -0.5 \\ 0.5 & 1 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 1 & 1 \\ -2 & 1 \\ 1 & 1.1 \end{pmatrix},$$

$$B_1 = \begin{pmatrix} 1 & 3 & 0.8 & 2 \\ 1 & 1 & -3 & 1.2 \\ -1.1 & -2.1 & 5 & 1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & 2.6 & 0.8 \\ 2.5 & -1.1 & 1 \\ 1 & -1.5 & 2 \end{pmatrix},$$

$$F_1 = \begin{pmatrix} -8.175 & -13.925 & 23.5 & -4.63 \\ 3.925 & 13.675 & 11.25 & 12.01 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 4.35 & 13.57 & 8.3 \\ 15.75 & 8.02 & -1.3 \\ 5.165 & 14.742 & 8.81 \end{pmatrix}.$$

Then the solution X from (2.1) is

$$X = \begin{pmatrix} 1 & -3 & 2.7 \\ 5 & 1.3 & -0.1 \end{pmatrix}.$$

We define the upper bound of the iterative parameter μ by μ_{up}. Then we have

	(1.5) of [16]	Algorithm 1
μ_{up}	0.0061	0.0236

It is obvious that the convergence region of our Algorithm 1 is larger than the algorithm (1.5) of [16].

Define the relative error $\delta := \|X(k) - X\|/\|X\|$. Taking the initial value $X(0) = 1e - 6 * eye(2, 3)$. Then we apply the Algorithms 1 and 3 to compute $X(k)$, respectively. Meanwhile, the weighted parameters w_1 and w_2 are set to be 0.5 in Algorithm 3. The iterative solutions $X(k)$ are shown in Figs. 2 and 3.

In Example 6.2, we have demonstrated that the convergence region of our iterative algorithms are larger than iterative algorithm (1.5) of [16]. In the next example, we will apply our iterative algorithms to solve some large matrix equations which usually occurred in real world applications. The obtained results is depicted in Fig. 4. The residual vector is defined by

$$r_k = \sum_{i=1}^{2} \|A_i X(k) B_i - F_i\|_F.$$

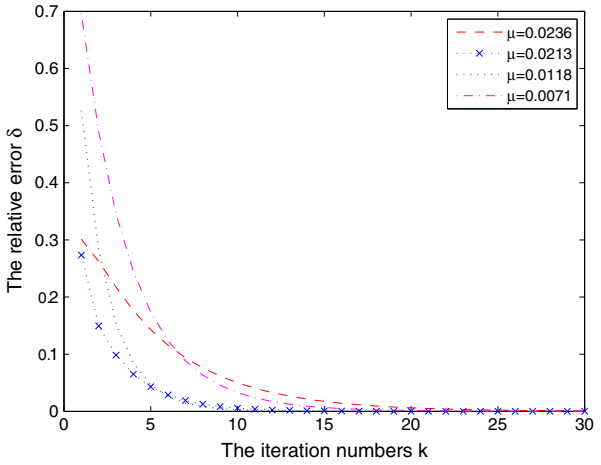

Fig. 2 The relative errors δ versus k of the Algorithm 1

Example 6.3 Consider the matrix equations (6.1) which have been appeared in Example 6.2, where A_i and B_i, $i = 1, 2$ are 256×256 matrices and generated in MATLAB as follows:

$$A_1 = (round((5 + 2 * randn(256, 256)). * 100))/100,$$
$$B_1 = (round((2 + 2 * randn(256, 256)). * 100))/100,$$
$$A_2 = (round((6 + randn(256, 256)). * 100))/100,$$
$$B_2 = (round((4 + randn(256, 256)). * 100))/100,$$
$$F_1 = unifrnd(-4e4, 5e4, 256),$$
$$F_2 = unifrnd(-2e4, 4e4, 256).$$

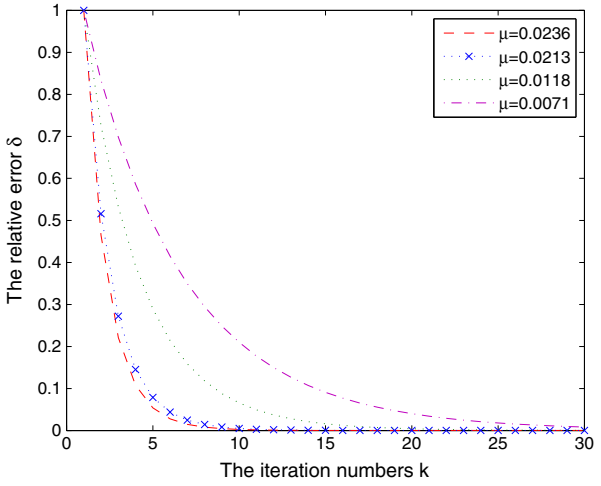

Fig. 3 The relative errors δ versus k of the Algorithm 3

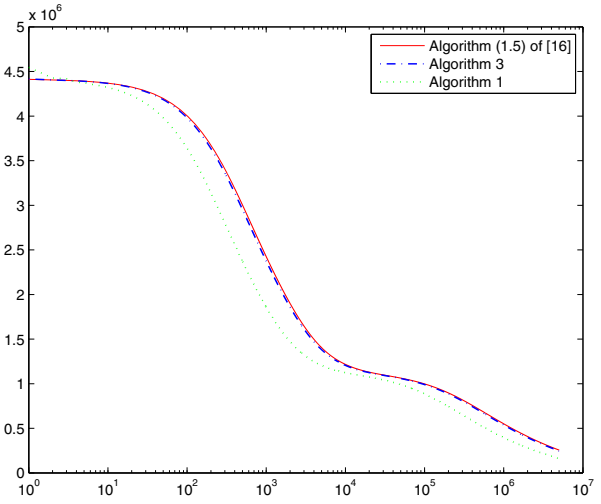

Fig. 4 Comparison of convergence curves

It is time consume to solve these matrix equations by Lemma 2.3 directly since it involved the matrix inverse. So the best way is to choose an iterative method. It can be seen from Fig. 4 that when the three algorithms choose the largest converge factor μ itself, the Algorithm 1 converges the fastest. The Algorithm 3 and Algorithm (1.5) of [16] perform nearly the same.

7 Conclusion

Solving the linear matrix equations is an active area of research. In this paper, we have proposed two efficient iterative algorithms for solving the matrix equations (1.1). According to Theorems 4.1 and 5.1, the convergence region of the Algorithms 1 and 3 are larger than the algorithm (1.5) of [16]. Thus we explain their experimental results. Furthermore, how to make use of our proposed methods to solve the linear matrix equations (1.1) over symmetric, Skew-symmetric, bisymmetric and generalized centro-symmetric matrices is the work undertaken.

Acknowledgments The authors would like to thank the editor and referees for their constructive comments and helpful suggestions, which greatly improved the original manuscript of this paper.

This research was supported by EU FP7-IRSES projects EYE2E(269118) and LIVCODE(295151). And this work was also supported partly by the National Natural Science Foundations of China (11131006, 60970149) and the Natural Science Foundations of Jiangxi Province (2009GZS0021, CA201107114) and the Youth Science Funds of The Education Department of Jiangxi Province (GJJ12141).

References

1. Datta, B.N.: Numerical Methods for Linear Control Systems Design and Analysis. Elseiver, Amsterdam (2003)

2. Ding, F., Chen, T.W.: Gradient based iterative algorithms for solving a class of matrix equations. IEEE Trans. Automat. Control. **50**, 1216–1221 (2005)
3. Ding, F., Chen, T.W.: On iterative solutions of general coupled matrix equations, SIAM. J. Control Optim. **44**, 2269–2284 (2006)
4. Ding, F., Chen, T.W.: Iterative least squares solutions of coupled Sylvester matrix equations. Syst. Control Lett. **54**, 95–107 (2005)
5. Ding, F., Chen, T.W.: Hierarchical gradient-based identification of multivariable discrete-time systems. Automation **41**, 95–107 (2005)
6. Ding, F., Chen, T.W.: Hierarchical least squares identification methods for multivable systems. IEEE Trans. Automat. Control. **50**, 397–402 (2005)
7. Ding, F., Chen, T.W.: Hierarchical identification of lifted state-space models for general dual-rate systems. IEEE Trans. Circ. Syst. **52**, 1179–1187 (2005)
8. Ding, F., Liu, P.X., Ding, J.: Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle. Appl. Math. Comput. **197**, 41–50 (2008)
9. Dehghan, M., Hajarian, M.: On the relexive solutions of the matrix equation $AXB+CYD = E$. Bull. Korean Math. Soc. **46**(3), 511–519 (2009)
10. Xie, L., Liu, Y.J., Yang, H.: Gradient based and least squares based iterative algorithms for matrix equations $AXB + CX^TD = F$. Appl. Math. Comput. **217**, 2191–2199 (2010)
11. Hajarian, M., Dehghan, M.: The generalized centro-symmetric and least squares generalized centro-symmetric solutions of the matrix equation $AYB + CY^TD = E$. Math. Meth. Appl. Sci. **34**, 1562–1579 (2011)
12. Liang, K.F., Liu, J.: Iterative algorithms for the minimum-norm solution and the least-squares solution of the linear matrix equations. Appl. Math. Comput. **218**, 3166–3175 (2011)
13. On Hermitian and skew-Hermitian splitting iteration methods for continous Sylvester equations. J. Comput. Math. **2**, 185–198 (2011)
14. Wang, X., Dai, L., Liao, D.: A modified gradient based algorithm for solving Sylvester equations. Appl. Math. Comput. **218**, 5620–5628 (2012)
15. Hansen, P.C., Nagy, J.M., O'Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. SIAM, Philadelphia (2006)
16. Ding, J., Liu, Y.J., Ding, F.: Iterative solutions to matrix equations of the form $A_iXB_i = F_i$. Comput. Math. Appl. **59**, 3500–3507 (2010)
17. Bertsekas, D.P.: A new class of incremental gradient methods for least squares problems. SIAM J. Optimiz. **7**(4), 913–926 (1997)
18. Nedić, A., Bertsekas, D.P.: Incremental subgradient methods for nondifferentiable optimization. SIAM J. Optimiz. **12**, 109–138 (2001)
19. Dos Santos, L.T.: A parallel subgradient projections method for the convex feasibility problem. J. Comput. Appl. Math. **18**(3), 307–320 (1987)
20. Combettes, P.L., Puh, H.: Iterations of parallel convex projections in Hilbert spaces. Numer. Funct. Anal. Optimiz. **15**, 225–243 (1994)
21. Huang, G.X., Yin, F., Guo, K.: An iterative method for the skew-symmetric solution and the optimal approximate solution of the matrix equation $AXB = C$. J. Comput. Appl. Math. **212**, 231–244 (2008)
22. Liang, M.L., You, C.H., Dai, L.F.: An efficient algorithm for the generalized centro-symmetric solution of matrix equation $AXB = C$. Numer. Algor. **44**, 173–184 (2007)
23. Peng, Z.Y.: New matrix iterative methods for constraint solutions of the matrix equation $AXB = C$. J. Comput. Appl. Math. **235**(3), 726–735 (2010)
24. Wang, Q.W.: A system of matrix equations and a linear matrix equation over arbitrary regular rings with identity. Linear Algebra Appl. **384**, 43–54 (2004)
25. Sheng, X.P., Chen, G.L.: A finite iterative method for solving a pair of linear matrix equations $(AXB, CXD) = (E, F)$. Appl. Math. Comput. **189**(2), 1350–1358 (2007)
26. Cai, J., Chen, G.L.: An iterative algorithm for the least squares bisymmetric solutions of the matrix equations $A_1XB_1 = C_1$, $A_2XB_2 = C_2$. Math. Comput. Model. **50**, 1237–1244 (2009)
27. Dehghan, M., Hajarian, M.: An iterative algorithm for solving a pair of matrix equations $AYB = E, CYD = F$ over generalized centro-symmetric matrices. Comput. Math. Appl. **56**, 3246–3260 (2008)
28. Ruszxzynski, A.: Nonlinear Optimization. Princeton University Press, New Jersey (2006)
29. Polyak, B.T.: Introduction to Optimization. Optimization Softwarse, New York (1987)

30. Su, M., Xu, H.K.: Remarks on the gradient-projection algorithm. J. Nonlinear Anal. Opt. **1**(1), 35–43 (2010)
31. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert spaces. Springer, New York (2011)
32. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: Bauschke, H.H., Burachik, R., Combettes, P.L., Elser, V., Luke, D.R., Wolkowicz, H. (eds.) Fixed-Point Algorithms for Inverse Problems in Science and Engineering. Springer, New York (2009)
33. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. **4**, 1168–1200 (2005)
34. Byrne, C.: A unified treatment of some iterative algorithms in signal processing and image reconstruction. Inverse Problems **20**, 103–120 (2004)