2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Gradient-based iterative solutions for general matrix equations

Li Xie, Huizhong Yang, Jie Ding, Feng Ding

Abstract— **In this paper, we study solutions of general matrix equations by using the iterative method and present gradient-based iterative algorithms by applying the hierarchical identification principle. Convergence analysis indicates that the iterative solutions always converge fast to the exact solutions for any initial values and small condition numbers of the associated matrices. Several numerical examples are included.**

I. Introduction

In systems and control, we often encounter the matrix equations of the form,

$$AX+X^{\mathrm{T}}B=F, \tag{1}$$

where $X\in\mathbb{R}^{m\times n}$ is an unknown matrix and A, B and F are constant (coefficient) matrices of appropriate sizes. Such matrix equations play an important role in automatic control.

Just pointed out in [1] that the traditional method of solving such matrix equations using the Kronecker product involves the inversion of associated large matrix $(mn)\times(mn)$ and results in increasing computation and excessive computer memory. The gradient-based iterative method in [1] can be used to solve the matrix equations of the forms,

$$AX+XB=F,$$

and

$$\sum_{i=1}^{p}A_iXB_i=F.$$

But the method there is not suitable for solving the general matrix equations of the form,

$$\sum_{i=1}^{p}A_iXB_i+\sum_{i=1}^{q}C_iX^{\mathrm{T}}D_i=F \tag{2}$$

which including the Lyapunov equations, Sylvester equations as the special cases, e.g., [2]. Recently, iterative approaches for matrix equations have received much attention, e.g., [3]. For example, Dehghan and Hajarian studied the iterative algorithm for the reflexive solutions of the generalized coupled Sylvester matrix equations [4]; Mukaidani et al gave a numerical algorithm for finding solution of cross-coupled algebraic Riccati equations [5]; Wang et al proposed the iterative solutions of coupled discrete Markovian jump Lyapunov equations; and some related contributions include the generalized Sylvester mapping and matrix equations [6] and the condition numbers of the generalized Sylvester equations [7]. Also, Kilicman et al presented the vector least-squares solutions for coupled singular matrix equations; Ding and Chen presented a gradient-based and a least-squares based iterative algorithms for generalized Sylvester matrix equations and general coupled matrix equations by introducing the star ($\star$) product of matrices [8], [9]. Finally, Zhour et al discussed some new connections between matrix products for partitioned and non-partitioned matrices, including the star product [10] and the solutions of other matrix equations can be found in [11], [12].

This paper decomposes the system in (2) into several subsystems by applying the hierarchical identification principle [8], [13], regards the unknown matrix X as the system parameter matrix, and presents a gradient-based iterative algorithm for solving the matrix equation in (2).

The rest of the paper is organized as follows. Section II and III derive iterative algorithms for solving the matrix equations in (1) and (2) and study convergence properties of the algorithms. Section IV provides several examples to illustrate the effectiveness of the proposed algorithms. Finally, we offer some concluding remarks in Section V.

II. The Equation $AX+X^{\mathrm{T}}B=F$

In this section, we apply the hierarchical identification principle to solve matrix equations:

$$AX+X^{\mathrm{T}}B=F, \tag{3}$$

where $A\in\mathbb{R}^{n\times m}$, $B=[b_1,b_2,\cdots,b_n]\in\mathbb{R}^{m\times n}$ and $F=[f_1,f_2,\cdots,f_n]\in\mathbb{R}^{n\times n}$ are given constant matrices, $X\in\mathbb{R}^{m\times n}$ is the unknown matrix to be solved.

Let us introduce some notations first. The symbol I_n stands for an identity matrix of size $n\times n$. For two matrices M and N, $M\otimes N$ is their Kronecker product (called direct product); for an $m\times n$ matrix

$$X=[x_1,x_2,\cdots,x_n]\in\mathbb{R}^{m\times n},\ x_i\in\mathbb{R}^m,$$

col$[X]$ is an mn-dimensional vector formed by the columns of X, i.e.,

$$\mathrm{col}[X]=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\in\mathbb{R}^{mn}.$$

Define

$$S:=I_n\otimes A+\begin{bmatrix}I_n\otimes b_1^{\mathrm{T}}\\I_n\otimes b_2^{\mathrm{T}}\\\vdots\\I_n\otimes b_n^{\mathrm{T}}\end{bmatrix}\in\mathbb{R}^{n^2\times mn}.$$

This work was supported by the National Natural Science Foundation of China (60674092).

The authors are with the School of Communication and Control Engineering, Jiangnan University, Wuxi, P. R. China 214122. xieli2412@126.com (L. Xie); yhz@jiangnan.edu.cn (H.Z. Yang); djvsqxb@163.com (J. Ding); fding@jiangnan.edu.cn (F. Ding).

978-1-4244-4524-0/09/$25.00 ©2009 AACC

Lemma 1: Equation (3) has a unique solution if and only if $\mathrm{rank}\{S, \mathrm{col}[F]\} = \mathrm{rank}[S] = mn$; in this case, the unique solution is given by

$$\mathrm{col}[X] = (S^{\mathrm{T}}S)^{-1}S^{\mathrm{T}}\mathrm{col}[F] \tag{4}$$

and the corresponding homogeneous equation $AX + X^{\mathrm{T}}B = \mathbf{0}$ has a unique solution $X = \mathbf{0}$.

Lemma 1 is a special case of Lemma 2 and the proof of which can be done in a similar way.

Although (4) can be used to solve (3), it requires excessive computer memory because of computing the inversion of the large matrix $S^{\mathrm{T}}S$ of size $(mn) \times (mn)$ as the dimension of X increases. This motivates us to study the iterative algorithm to solve (3).

According to the hierarchical identification principle, the unknown matrix X is regarded as the parameter matrix, the system in (3) is decomposed into two subsystems and then the parameter matrix of each subsystem is identified by using the gradient search method.

Define two matrices:

$$Q_1 := F - X^{\mathrm{T}}B, \tag{5}$$

$$Q_2 := F - AX. \tag{6}$$

Then, from (3), we obtain two fictitious subsystems:

$$\begin{aligned}\mathrm{Sub1}: &\quad AX = Q_1,\\ \mathrm{Sub2}: &\quad X^{\mathrm{T}}B = Q_2.\end{aligned}$$

Let $X_1(k)$ and $X_2(k)$ be the estimates or iterative solutions of X at iteration k, associated with subsystems Sub1 and Sub2. Then applying the gradient search method [1] or Corollary 3 in [9] to Sub1 and Sub2 leads to the following iterative equations:

$$X_1(k) = X_1(k-1) + \mu A^{\mathrm{T}}[Q_1 - AX_1(k-1)], \tag{7}$$

$$X_2(k) = X_2(k-1) + \mu B[Q_2^{\mathrm{T}} - B^{\mathrm{T}}X_2(k-1)], \tag{8}$$

where $\mu > 0$ is the iterative step size or convergence factor to be given later. Substituting (5) into (7) and (6) into (8) gives

$$X_1(k) = X_1(k-1) + \mu A^{\mathrm{T}}[F - X^{\mathrm{T}}B - AX_1(k-1)], \tag{9}$$

$$X_2(k) = X_2(k-1) + \mu B[F - AX - X_2^{\mathrm{T}}(k-1)B]^{\mathrm{T}}. \tag{10}$$

Because the expressions on the right-hand sides of (9) and (10) contain the unknown matrix X, it is impossible to realize the algorithm in (9) and (10). Our solution is based on the hierarchical identification principle [1], [8], [13]: the unknown variable X in (9) and (10) is replaced with their estimates $X_1(k-1)$ and $X_2(k-1)$ at time $(k-1)$, i.e.,

$$\begin{aligned}X_1(k) = {} & X_1(k-1) + \mu A^{\mathrm{T}}[F - AX_1(k-1)\\ & -X_1^{\mathrm{T}}(k-1)B],\end{aligned} \tag{11}$$

$$\begin{aligned}X_2(k) = {} & X_2(k-1) + \mu B[F - AX_2(k-1)\\ & -X_2^{\mathrm{T}}(k-1)B]^{\mathrm{T}}.\end{aligned} \tag{12}$$

In fact, we need only an iterative solution $X(k)$ rather than two solutions $X_1(k)$ and $X_2(k)$. Taking the average of $X_1(k)$ and $X_2(k)$ as the iterative solution $X(k)$, we obtain a gradient-based iterative (GI) algorithm for the solution of system (3):

$$X(k) = \frac{X_1(k) + X_2(k)}{2}, \tag{13}$$

$$\begin{aligned}X_1(k) = {} & X(k-1) + \mu A^{\mathrm{T}}[F - AX(k-1)\\ & -X^{\mathrm{T}}(k-1)B],\end{aligned} \tag{14}$$

$$\begin{aligned}X_2(k) = {} & X(k-1) + \mu B[F - AX(k-1)\\ & -X^{\mathrm{T}}(k-1)B]^{\mathrm{T}}.\end{aligned} \tag{15}$$

The convergence factor μ can be simply taken to satisfy

$$0 < \mu < \mu_0 := \frac{2}{\lambda_{\max}[AA^{\mathrm{T}}] + \lambda_{\max}[B^{\mathrm{T}}B]}. \tag{16}$$

To initialize the algorithm, we take $X(0) = \mathbf{0}$ or some small real matrix, e.g., $X(0) = 10^{-6}\mathbf{1}_{m\times n}$ with $\mathbf{1}_{m\times n}$ being an $m \times n$ matrix whose elements are all 1.

Theorem 1: If the equation in (3) has a unique solution X, then for any initial value $X(0)$, the iterative solution $X(k)$ given by the algorithm in (13)-(16) converges to the solution X, i.e., $\lim\limits_{k\to\infty} X(k) = X$; or, the error $X(k) - X$ converges to zero.

The proof of Theorem 1 is omitted here but can be given later with the proof of Theorem 2 in the next section.

The convergence rate of the gradient-based iterative algorithm depends on the condition number of the associated system, like the iterative algorithm of the equation $Ax = b$ [1], [14]. Define the error matrix,

$$\tilde{X}(k) := X(k) - X.$$

Using (13)-(15), we have

$$\begin{aligned}\tilde{X}(k) = {} & \tilde{X}(k-1) - \frac{\mu}{2}A^{\mathrm{T}}[A\tilde{X}(k-1) + \tilde{X}^{\mathrm{T}}(k-1)B]\\ & -\frac{\mu}{2}B[A\tilde{X}(k-1) + \tilde{X}^{\mathrm{T}}(k-1)B]^{\mathrm{T}},\end{aligned}$$

which can be equivalently expressed as

$$\mathrm{col}[\tilde{X}(k)] = [I_{mn} - \frac{\mu}{2}\Phi]\mathrm{col}[\tilde{X}(k-1)], \tag{17}$$

where $\Phi := S^{\mathrm{T}}S$.

From (17), the closer the eigenvalues of $\frac{\mu}{2}\Phi$ are to 1, the closer the eigenvalues of $I_{mn} - \frac{\mu}{2}\Phi$ tend to be zero, and hence, the faster the error $\mathrm{col}[\tilde{X}(k)]$ or $\tilde{X}(k)$ converges to zero. In other words, the gradient-based iterative algorithm in (13)-(16) has a fast convergence rate for small condition numbers of Φ – see the examples later.

III. The general matrix equation

In this section, we will extend the iterative method to solution of a general matrix equation:

$$\sum_{i=1}^{p} A_iXB_i + \sum_{i=1}^{q} C_iX^{\mathrm{T}}D_i = F, \tag{18}$$

where $A_i \in \mathbb{R}^{r\times m}$, $B_i \in \mathbb{R}^{n\times s}$, $C_i \in \mathbb{R}^{r\times n}$, $D_i \in \mathbb{R}^{m\times s}$ and $F = [f_1, f_2, \cdots, f_s] \in \mathbb{R}^{r\times s}$ are given constant matrices, $X \in \mathbb{R}^{m\times n}$ is the unknown matrix to be solved.

Let $B_i=[b_i(j,k)]\in\mathbb{R}^{n\times s}$, $D_i=[d_{i1},d_{i2},\cdots,d_{is}]$, $d_{ij}\in\mathbb{R}^m$, $X=[x_1,x_2,\cdots,x_n]$ and

$$S_1:=\sum_{i=1}^{p}B_i^{\mathrm{T}}\otimes A_i+\sum_{i=1}^{q}\begin{bmatrix}C_i\otimes d_{i1}^{\mathrm{T}}\\ C_i\otimes d_{i2}^{\mathrm{T}}\\ \vdots\\ C_i\otimes d_{is}^{\mathrm{T}}\end{bmatrix}\in\mathbb{R}^{(rs)\times(mn)}. \qquad (19)$$

Lemma 2: The equation in (18) has a unique solution if and only if $\mathrm{rank}\{S_1,\mathrm{col}[F]\}=\mathrm{rank}[S_1]=mn$; in this case, the unique solution is given by

$$\mathrm{col}[X]=(S_1^{\mathrm{T}}S_1)^{-1}S_1^{\mathrm{T}}\mathrm{col}[F], \qquad (20)$$

and the corresponding homogeneous matrix equation in (18) with $F=\mathbf{0}$ has a unique solution $X=\mathbf{0}$.

Proof Equation (18) can be written as

$$\sum_{i=1}^{p}A_i[x_1,x_2,\cdots,x_n]\begin{bmatrix}b_i(1,1)&b_i(1,2)&\cdots&b_i(1,s)\\ b_i(2,1)&b_i(2,2)&\cdots&b_i(2,s)\\ \vdots&\vdots&&\vdots\\ b_i(n,1)&b_i(n,2)&\cdots&b_i(n,s)\end{bmatrix}$$
$$+\sum_{i=1}^{q}C_i\begin{bmatrix}x_1^{\mathrm{T}}\\ x_2^{\mathrm{T}}\\ \vdots\\ x_n^{\mathrm{T}}\end{bmatrix}[d_{i1},d_{i2},\cdots,d_{is}]=[f_1,f_2,\cdots,f_s].$$

Or

$$\sum_{i=1}^{p}A_i[x_1,x_2,\cdots,x_n]\begin{bmatrix}b_i(1,1)&b_i(1,2)&\cdots&b_i(1,s)\\ b_i(2,1)&b_i(2,2)&\cdots&b_i(2,s)\\ \vdots&\vdots&&\vdots\\ b_i(n,1)&b_i(n,2)&\cdots&b_i(n,s)\end{bmatrix}$$
$$+\sum_{i=1}^{q}C_i\begin{bmatrix}x_1^{\mathrm{T}}d_{i1}&x_1^{\mathrm{T}}d_{i2}&\cdots&x_1^{\mathrm{T}}d_{is}\\ x_2^{\mathrm{T}}d_{i1}&x_2^{\mathrm{T}}d_{i2}&\cdots&x_2^{\mathrm{T}}d_{is}\\ \vdots&\vdots&&\vdots\\ x_n^{\mathrm{T}}d_{i1}&x_n^{\mathrm{T}}d_{i2}&\cdots&x_n^{\mathrm{T}}d_{is}\end{bmatrix}=[f_1,f_2,\cdots,f_s].$$

Since $x_j^{\mathrm{T}}d_{ik}=d_{ik}^{\mathrm{T}}x_j$, we have

$$\sum_{i=1}^{p}A_i[x_1,x_2,\cdots,x_n]\begin{bmatrix}b_i(1,1)&b_i(1,2)&\cdots&b_i(1,s)\\ b_i(2,1)&b_i(2,2)&\cdots&b_i(2,s)\\ \vdots&\vdots&&\vdots\\ b_i(n,1)&b_i(n,2)&\cdots&b_i(n,s)\end{bmatrix}$$
$$+\sum_{i=1}^{q}C_i\begin{bmatrix}d_{i1}^{\mathrm{T}}x_1&d_{i2}^{\mathrm{T}}x_1&\cdots&d_{is}^{\mathrm{T}}x_1\\ d_{i1}^{\mathrm{T}}x_2&d_{i2}^{\mathrm{T}}x_2&\cdots&d_{is}^{\mathrm{T}}x_2\\ \vdots&\vdots&&\vdots\\ d_{i1}^{\mathrm{T}}x_n&d_{i2}^{\mathrm{T}}x_n&\cdots&d_{is}^{\mathrm{T}}x_n\end{bmatrix}=[f_1,f_2,\cdots,f_s].$$

Expanding gives

$$\sum_{i=1}^{p}[A_ib_i(1,k)x_1+A_ib_i(2,k)x_2+\cdots+A_ib_i(n,k)x_n]$$
$$+\sum_{i=1}^{q}C_i\begin{bmatrix}d_{ik}^{\mathrm{T}}&&&\\ &d_{ik}^{\mathrm{T}}&&\\ &&\ddots&\\ &&&d_{ik}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}$$
$$=f_k,\ k=1,2,\cdots,s.$$

According to the definition of S_1, we have

$$S_1\mathrm{col}[X]=\mathrm{col}[F].$$

Since $\mathrm{rank}\{S_1,\mathrm{col}[F]\}=\mathrm{rank}[S_1]=mn$, this proves Lemma 2 according to the theory of linear equations. □

Equation (20) can give the solution of (18) but also requires excessive computer memory as the dimension of X increases. So we seek for an iterative algorithm to solve (18) by means of the hierarchical identification principle, similar to the last section. The details are as follows.

Define following matrices:

$$Q_j:=F-\sum_{i=1,i\neq j}^{p}A_iXB_i-\sum_{i=1}^{q}C_iX^{\mathrm{T}}D_i,\ j=1,2,\cdots,p. \qquad (21)$$

$$Q_{p+l}:=F-\sum_{i=1}^{p}A_iXB_i-\sum_{i=1,i\neq l}^{q}C_iX^{\mathrm{T}}D_i,\ l=1,2,\cdots,q. \qquad (22)$$

Then from (18), we obtain $p+q$ fictitious subsystems

$$\begin{aligned}\text{Subsystem } j\ &:\quad A_jXB_j=Q_j,\ j=1,2,\cdots,p.\\ \text{Subsystem } p+l&:\quad C_lX^{\mathrm{T}}D_l=Q_{p+l},\ l=1,2,\cdots,q.\end{aligned}$$

Let $X_i(k)$ be the estimate or iterative solution of X at iteration k, associated with ith subsystem. Applying the gradient search method [1] or Corollary 3 in [9] to Subsystem i, $i=1,2,\cdots,p+q$, we can obtain the iterative algorithms:

$$X_j(k)=X_j(k-1)+\mu A_j^{\mathrm{T}}[Q_j-A_jX_j(k-1)B_j]B_j^{\mathrm{T}},$$
$$j=1,2,\cdots,p. \qquad (23)$$

$$X_{p+l}(k)=X_{p+l}(k-1)+\mu D_l[Q_{p+l}^{\mathrm{T}}$$
$$-D_l^{\mathrm{T}}X_{p+l}(k-1)C_l^{\mathrm{T}}]C_l,\ l=1,2,\cdots,q. \qquad (24)$$

The convergence factor $\mu>0$ will be given later. Substituting (21) and (22) into (23) and (24), then replacing the unknown variable X with its estimate $X_j(k-1)$ and $X_{p+l}(k-1)$ respectively by means of the hierarchical identification principle [8], [9], [13] gives

$$X_j(k)=X_j(k-1)+\mu A_j^{\mathrm{T}}\Big[F-\sum_{i=1}^{p}A_iX_j(k-1)B_i$$
$$-\sum_{i=1}^{q}C_iX_j^{\mathrm{T}}(k-1)D_i\Big]B_j^{\mathrm{T}}, \qquad (25)$$

$$X_{p+l}(k)=X_{p+l}(k-1)+\mu D_l\Big[F-\sum_{i=1}^{p}A_iX_{p+l}(k-1)B_i$$
$$-\sum_{i=1}^{q}C_iX_{p+l}^{\mathrm{T}}(k-1)D_i\Big]^{\mathrm{T}}C_l. \qquad (26)$$

In fact, we need only an iterative solution $X(k)$ rather than $p+q$ solutions $X_i(k)$: $i=1,2,\cdots,p+q$. Taking the average of the $p+q$ solutions as the iterative solution $X(k)$ of X, we obtain a gradient-based iterative (GI) algorithm for the the general matrix equation in (18):

$$X(k)=\frac{1}{p+q}\Big[\sum_{j=1}^{p}X_j(k)+\sum_{l=1}^{q}X_{p+l}(k)\Big], \qquad (27)$$

$$X_j(k)=X(k-1)+\mu A_j^{\mathrm{T}}\Big[F-\sum_{i=1}^{p}A_iX(k-1)B_i$$
$$-\sum_{i=1}^{q}C_iX^{\mathrm{T}}(k-1)D_i\Big]B_j^{\mathrm{T}}, \qquad (28)$$

$$\begin{aligned}X_{p+l}(k)=X(k-1)+\mu D_l\Big[F-\sum_{i=1}^{p}A_iX(k-1)B_i\\-\sum_{i=1}^{q}C_iX^{\rm T}(k-1)D_i\Big]^{\rm T}C_l.\end{aligned}\qquad(29)$$

A conservative choice of the convergence factor μ is

$$\begin{aligned}0<\mu\ <2\Big\{\sum_{j=1}^{p}\lambda_{\max}[A_jA_j^{\rm T}]\lambda_{\max}[B_j^{\rm T}B_j]\\+\sum_{l=1}^{q}\lambda_{\max}[C_lC_l^{\rm T}]\lambda_{\max}[D_l^{\rm T}D_l]\Big\}^{-1}=:\mu_0.\end{aligned}\qquad(30)$$

To initialize the algorithm, we take $X(0)=\mathbf{0}$ or some small real matrix, e.g., $X(0)=10^{-6}\mathbf{1}_{m\times n}$ with $\mathbf{1}_{m\times n}$ being an $m\times n$ matrix whose elements are all 1.

Theorem 2: If the equation in (18) has a unique solution X, then the iterative solution $X(k)$ given by the algorithm in (27)-(30) converges to X, i.e., $\lim\limits_{k\to\infty}X(k)=X$; or, the error $X(k)-X$ converges to zero for any initial value $X(0)$.

Proof Define the estimation error matrices:

$$\begin{aligned}\tilde{X}_i(k)&:=X_i(k)-X,\\ \tilde{X}(k)&:=X(k)-X\\ &=\frac{1}{p+q}\Big[\sum_{j=1}^{p}\tilde{X}_j(k)+\sum_{l=1}^{q}\tilde{X}_{p+l}(k)\Big],\end{aligned}\qquad(31)$$

and

$$\xi_i(k):=A_i\tilde{X}(k-1)B_i,\quad \eta_i(k):=D_i^{\rm T}\tilde{X}(k-1)C_i^{\rm T}.\qquad(32)$$

Using (18), (28)-(29) and (32), it is easy to get

$$\begin{aligned}\tilde{X}_j(k)&=X_j(k)-X\\ &=X(k-1)-X+\mu A_j^{\rm T}\Big[F-\sum_{i=1}^{p}A_iX(k-1)B_i\\ &\quad-\sum_{i=1}^{q}C_iX^{\rm T}(k-1)D_i\Big]B_j^{\rm T}\\ &=\tilde{X}(k-1)-\mu A_j^{\rm T}\Big[\sum_{i=1}^{p}A_i(X(k-1)-X)B_i\\ &\quad+\sum_{i=1}^{q}C_i(X^{\rm T}(k-1)-X^{\rm T})D_i\Big]B_j^{\rm T}\\ &=\tilde{X}(k-1)-\mu A_j^{\rm T}\Big[\sum_{i=1}^{p}A_i\tilde{X}(k-1)B_i\\ &\quad+\sum_{i=1}^{q}C_i\tilde{X}^{\rm T}(k-1)D_i\Big]B_j^{\rm T}\\ &=\tilde{X}(k-1)-\mu A_j^{\rm T}\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]B_j^{\rm T}.\end{aligned}\qquad(33)$$

Similarly,

$$\tilde{X}_{p+l}(k)=\tilde{X}(k-1)-\mu D_l\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]^{\rm T}C_l.\qquad(34)$$

Taking the norm of both sides of the above equations and using formula: ${\rm tr}[AB]={\rm tr}[BA]$ and ${\rm tr}[A^{\rm T}]={\rm tr}[A]$ give

$$\begin{aligned}\|\tilde{X}_j(k)\|^2&={\rm tr}[\tilde{X}_j^{\rm T}(k)\tilde{X}_j(k)]\\ &=\Big\|\tilde{X}(k-1)-\mu A_j^{\rm T}\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]B_j^{\rm T}\Big\|^2\\ &=\|\tilde{X}(k-1)\|^2\\ &\quad-\mu{\rm tr}\Big\{\tilde{X}^{\rm T}(k-1)A_j^{\rm T}\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]B_j^{\rm T}\Big\}\\ &\quad-\mu{\rm tr}\Big\{B_j\Big[\sum_{i=1}^{p}\xi_i^{\rm T}(k)+\sum_{i=1}^{q}\eta_i(k)\Big]A_j\tilde{X}(k-1)\Big\}\\ &\quad+\mu^2\Big\|A_j^{\rm T}\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]B_j^{\rm T}\Big\|^2\\ &\le\|\tilde{X}(k-1)\|^2-2\mu{\rm tr}\Big\{\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]\xi_j^{\rm T}(k)\Big\}\\ &\quad+\mu^2\lambda_{\max}[A_jA_j^{\rm T}]\lambda_{\max}[B_j^{\rm T}B_j]\Big\|\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big\|^2.\end{aligned}\qquad(35)$$

Similarly,

$$\begin{aligned}&\|\tilde{X}_{p+l}(k)\|^2\\ &\le\|\tilde{X}(k-1)\|^2-2\mu{\rm tr}\Big\{\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]\eta_l(k)\Big\}\\ &\quad+\mu^2\lambda_{\max}[C_lC_l^{\rm T}]\lambda_{\max}[D_l^{\rm T}D_l]\Big\|\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big\|^2.\end{aligned}\qquad(36)$$

Hence, using (35) and (36) and from (31), we have

$$\begin{aligned}\|\tilde{X}(k)\|^2&=\frac{1}{(p+q)^2}\Big\|\sum_{j=1}^{p}\tilde{X}_j(k)+\sum_{l=1}^{q}\tilde{X}_{p+l}(k)\Big\|^2\\ &\le\frac{1}{p+q}\Big(\sum_{j=1}^{p}\|\tilde{X}_j(k)\|^2+\sum_{l=1}^{q}\|\tilde{X}_{p+l}(k)\|^2\Big)\\ &\le\frac{1}{p+q}\Big\{p\|\tilde{X}(k-1)\|^2\\ &\quad-2\mu{\rm tr}\Big\{\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]\Big[\sum_{j=1}^{p}\xi_j^{\rm T}(k)\Big]\Big\}\\ &\quad+\mu^2\Big(\sum_{j=1}^{p}\lambda_{\max}[A_jA_j^{\rm T}]\lambda_{\max}[B_j^{\rm T}B_j]\Big)\\ &\quad\Big\|\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big\|^2+q\|\tilde{X}(k-1)\|^2\\ &\quad-2\mu{\rm tr}\Big\{\Big[\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big]\Big[\sum_{l=1}^{q}\eta_l(k)\Big]\Big\}\\ &\quad+\mu^2\Big(\sum_{l=1}^{q}\lambda_{\max}[C_lC_l^{\rm T}]\lambda_{\max}[D_l^{\rm T}D_l]\Big)\\ &\quad\Big\|\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big\|^2\Big\}\\ &=\|\tilde{X}(k-1)\|^2\\ &\quad-\frac{1}{p+q}\Big\{2\mu-\mu^2\Big(\sum_{j=1}^{p}\lambda_{\max}[A_jA_j^{\rm T}]\lambda_{\max}[B_j^{\rm T}B_j]\\ &\quad+\sum_{l=1}^{q}\lambda_{\max}[C_lC_l^{\rm T}]\lambda_{\max}[D_l^{\rm T}D_l]\Big)\Big\}\Big\|\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\rm T}(k)\Big\|^2\\ &\le\|\tilde{X}(0)\|^2\end{aligned}$$

$$-\frac{\mu}{p+q}\Big\{2-\mu\Big(\sum_{j=1}^{p}\lambda_{\max}[A_jA_j^{\mathrm T}]\lambda_{\max}[B_j^{\mathrm T}B_j] + \sum_{l=1}^{q}\lambda_{\max}[C_lC_l^{\mathrm T}]\lambda_{\max}[D_l^{\mathrm T}D_l]\Big)\Big\} \Big(\sum_{j=1}^{k}\Big\|\sum_{i=1}^{p}\xi_i(j)+\sum_{i=1}^{q}\eta_i^{\mathrm T}(j)\Big\|^2\Big).$$

If the convergence factor μ is chosen to satisfy

$$0<\mu \ <2\Big\{\sum_{j=1}^{p}\lambda_{\max}[A_jA_j^{\mathrm T}]\lambda_{\max}[B_j^{\mathrm T}B_j] + \sum_{l=1}^{q}\lambda_{\max}[C_lC_l^{\mathrm T}]\lambda_{\max}[D_l^{\mathrm T}D_l]\Big\}^{-1},$$

then we have

$$\sum_{j=1}^{k}\Big\|\sum_{i=1}^{p}\xi_i(j)+\sum_{i=1}^{q}\eta_i^{\mathrm T}(j)\Big\|^2<\infty.$$

For the necessary condition of the series convergence, when $k\to\infty$, we have

$$\Big\|\sum_{i=1}^{p}\xi_i(k)+\sum_{i=1}^{q}\eta_i^{\mathrm T}(k)\Big\|^2\to 0,$$

or

$$\Big\|\sum_{i=1}^{p}A_i\tilde X(k-1)B_i+\sum_{i=1}^{q}C_i\tilde X^{\mathrm T}(k-1)D_i\Big\|^2\to 0.$$

According to Lemma 2, we can get $\tilde X(k-1)\to\mathbf{0}$ as $k\to\infty$. This proves Theorem 2. □

Next, we show that the convergence rate of the gradient-based iterative algorithm in (27)-(30) depends on the condition number of the associated system. From (31)-(34), we can get an error equation

$$\mathrm{col}[\tilde X(k)]=\Big(I_{mn}-\frac{\mu}{p+q}\Phi\Big)\mathrm{col}[\tilde X(k-1)], \qquad (37)$$

where $\Phi:=S_1^{\mathrm T}S_1$.

From (37), we can see that the closer the eigenvalues of $\frac{\mu}{p+q}\Phi$ are to 1, the closer the eigenvalues of $I_{mn}-\frac{\mu}{p+q}\Phi$ tend to be zero, and hence, the faster the error $\mathrm{col}[\tilde X(k)]$ or $\tilde X(k)$ converges to zero. In other words, the gradient-based iterative algorithm in (27)-(30) has a fast convergence rate for small condition numbers of Φ.

IV. Example

This section gives three examples to illustrate the performances of the proposed algorithms.

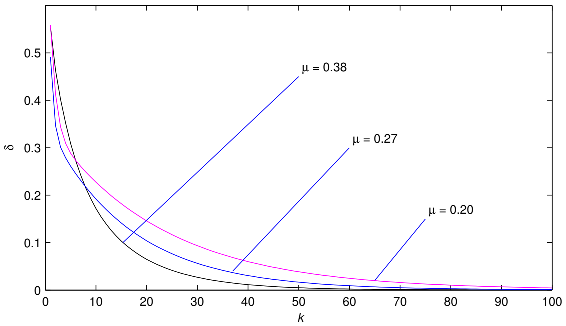
Example 1 Suppose that $AX+X^{\mathrm T}B=F$, where

$$A=\begin{bmatrix}1&1\\2&-1\end{bmatrix},\quad B=\begin{bmatrix}1&-1\\1&1\end{bmatrix},\quad F=\begin{bmatrix}8&8\\5&2\end{bmatrix}.$$

From (4), we can obtain the solution of this matrix equation, which is

$$X=\begin{bmatrix}x_{11}&x_{12}\\x_{21}&x_{22}\end{bmatrix}=\begin{bmatrix}1&2\\3&4\end{bmatrix}.$$

Take $X(0)=10^{-6}\mathbf{1}_{2\times 2}$. Applying the GI algorithm in (13)-(16) to compute $X(k)$, the iterative errors $\delta:=\|X(k)-X\|/\|X\|$ versus k are shown in Fig. 1.



$\mu_0=2\{\lambda_{\max}[AA^{\mathrm T}]+\lambda_{\max}[B^{\mathrm T}B]\}^{-1}=0.27)$

Fig. 1. The errors δ versus k

From Fig. 1, it is clear that the errors δ are becoming smaller and go to zero as k increases. The effect of changing the convergence factor μ is illustrated in Fig. 1. We can see that for $\mu=0.20$, 0.27 and 0.38, the larger the convergence factor μ is, the faster the convergence rate is. However, if we keep enlarging μ, the algorithm will diverge. How to choose a best convergence factor is still a project to be studied.

Example 2 Suppose that $AX+X^{\mathrm T}B=F$, where A, B and F are 20×20 matrices ($m=n=20$) and produced randomly in Matlab functions, with the simulation program given below (since dimensions of matrices are too large to be given here),

```
m = 20; n = 20; Im = eye(m); In = eye(n);
rand('state',34);
A = triu(rand(m, m), 1) + diag(α + diag(rand(m)));
B = triu(rand(m, m), 1) + 2∗diag(α + diag(rand(m)));
C = rand(m, m);
S1 = kron(eye(m), B(:, 1)′);
for i = 2:20
    S1 = [S1; kron(eye(m), B(:, i)′)];
end
S = kron(eye(m), A) + S1;
TX = reshape(S\reshape(C, m ∗ m, 1), m, m);
X = ones(m,m) ∗ 1E-6;
mu = 2/(max(eig(A ∗ A′)) + max(eig(B ∗ B′)));
Phi = S′ ∗ S;
q1 = eig(Phi);
q2 = [max(q1), min(q1), max(q1)/min(q1)];
LL = 100;
fm = norm(TX, ’fro’);
for i = 1:LL
    Ct = C-A ∗ X-X′ ∗ B;
    X = X+ mu ∗ (A′ ∗ Ct + B ∗ Ct′)/2;
    E(i) = norm(X-TX, ’fro’)/fm;
end
```

This program contains a variable α. For different α values ($\alpha=1$, 2, 3, 6, and so on) and $\mu=\mu_0$ given by (16), the iterative errors δ versus k are shown in Fig. 2 and the corresponding condition numbers of Φ are shown in Table I, where $\lambda_{\max}[\Phi]$ and $\lambda_{\min}[\Phi]$ represent the biggest and smallest eigenvalue of Φ, respectively, and cond$[\Phi]$ denotes the condition number of Φ.

From Fig. 2 and Table I, we see that the bigger the α is, the faster the convergence rate of the GI algorithm is, i.e., the convergence rate becomes faster as the condition number cond$[\Phi]$ of Φ is decreasing. As $\alpha=1$, the condition number

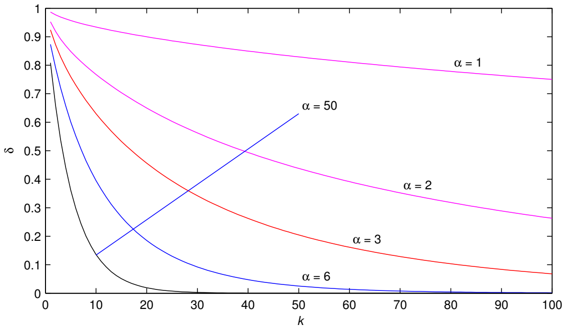


Fig. 2. The errors δ versus k

TABLE I

THE CONDITION NUMBERS OF Φ WITH DIFFERENT α

α	μ_0	$\lambda_{\max}[\Phi]$	$\lambda_{\min}[\Phi]$	cond$[\Phi]$
1	0.01470	232.15	0.02	12447.00
2	0.01087	327.50	0.78	418.83
3	0.00824	441.10	3.20	138.11
6	0.00416	890.33	22.02	40.43
50	0.00014	26094.00	2364.30	11.04

cond$[\Phi] = 12447$, too large a condition number implies an ill-conditioned equation. However, if we keep on increasing iterative step k, the algorithm will still converge.

Example 3 Suppose that $A_1XB_1 + A_2XB_2 + C_1X^{\mathrm{T}}D_1 + C_2X^{\mathrm{T}}D_2 = F$, where

$$A_1 = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix},\ A_2 = \begin{bmatrix} 0 & 1 \\ 3 & -1 \end{bmatrix},\ B_1 = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix},\ C_1 = \begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix},\ C_2 = \begin{bmatrix} -1 & 3 \\ -1 & 2 \end{bmatrix},$$

$$D_1 = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix},\ D_2 = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix},\ F = \begin{bmatrix} 35 & 9 \\ 20 & 7 \end{bmatrix}.$$

From (20), the solution is

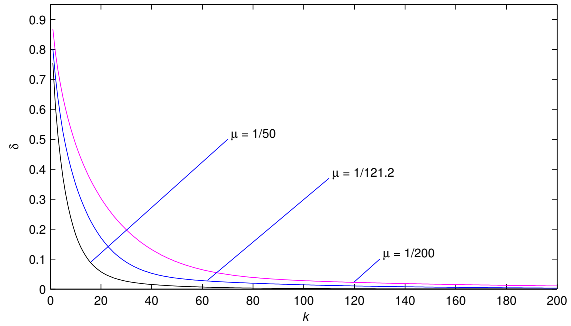
$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}.$$

Taking $X(0) = 10^{-6}\mathbf{1}_{2\times 2}$, we apply the algorithm in (27)-(30) to compute $X(k)$. the errors $\delta := \|X(k) - X\|/\|X\|$ versus k are shown in Fig. 3.

From Fig. 3, it is clear that the errors δ are becoming smaller and converges to zero as k increases. The effect of changing the convergence factor μ is illustrated in Fig. 3 with $\mu = 1/200$, 1/121.2 and 1/50, and a larger μ leads to a faster convergence rate.

V. Conclusions

The gradient-based iterative algorithms of solving general matrix equations are studied by using the hierarchical identification principle. The analysis indicates that the iterative solutions given by the proposed algorithms converge fast to their true solutions for any initial values and small condition numbers.



$$(\mu_0 = 2\Big\{\sum_{j=1}^{2}\lambda_{\max}[A_jA_j^{\mathrm{T}}]\lambda_{\max}[B_j^{\mathrm{T}}B_j] + \sum_{l=1}^{2}\lambda_{\max}[C_lC_l^{\mathrm{T}}]\lambda_{\max}[D_l^{\mathrm{T}}D_l]\Big\}^{-1} = 1/121.2)$$

Fig. 3. The errors δ versus k

References

[1] F. Ding and T. Chen, “Gradient based iterative algorithms for solving a class of matrix equations,” *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216-1221, 2005.

[2] Z.L. Tian and C.Q. Gu, “A numerical algorithm for Lyapunov equations,” *Applied Mathematics and Computation*, vol. 202, no. 1, pp. 44-53, 2008.

[3] A. Kilicman, Z. Abdel and A.A. Zhour, “Vector least-squares solutions for coupled singular matrix equations,” *Journal of Computational and Applied Mathematics*, vol. 206, no. 2, pp. 1051-1069, 2007.

[4] M. Dehghan and M. Hajarian, “An iterative algorithm for the reflexive solutions of the generalized coupled Sylvester matrix equations and its optimal approximation,” *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 571-588, 2008.

[5] H. Mukaidani, S. Yaniamoto and T. Yamamoto, “A numerical algorithm for finding solution of cross-coupled algebraic Riccati equations,” *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E91A, no. 2, pp. 682-685, 2008.

[6] B. Zhou and G.R. Duan, “On the generalized Sylvester mapping and matrix equations,” *Systems & Control Letters*, vol. 57, no. 3, pp. 200-208, 2008.

[7] Y.Q. Lin and Y.M. Wei, “Condition numbers of the generalized Sylvester equation,” *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2380-2385, 2007.

[8] F. Ding and T. Chen, “Iterative least squares solutions of coupled Sylvester matrix equations,” *Systems & Control Letters*, vol. 54, no. 2, pp. 95-107, 2005.

[9] F. Ding and T. Chen, “On iterative solutions of general coupled matrix equations,” *SIAM Journal on Control and Optimization*, vol.44, no. 6, pp. 2269-2284, 2006.

[10] Z. Al Zhour and A. Kilicman, “Some new connections between matrix products for partitioned and non-partitioned matrices,” *Computers & Mathematics with Applications*, vol. 54, no. 6, pp. 763-784, 2007.

[11] B. Zhou and G.R. Duan, “A new solution to the generalized Sylvester matrix equation $AV - EVF = BW$,” *Systems & Control Letters*, vol. 55, no. 3, pp. 193-198, 2006.

[12] B. Zhou and G.R. Duan, “Parametric solutions to the generalized Sylvester matrix equation $AX - XF = BY$ and the regulator equation $AX - XF = BY + R$,” *Asian Journal of Control*, vol. 9, no. 4, pp. 475-483, 2007.

[13] F. Ding and T. Chen, “Hierarchical least squares identification methods for multivariable systems,” *IEEE Transactions on Automatic Control*, vol, 50, no. 3, pp. 397-402, 2005.

[14] G.H. Golub and C.F. Van Loan, Matrix Computations, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.