Mathematical and Computational Applications, Vol. 18, No. 1, pp. 12-18, 2013

A NEW METHOD FOR SOLVING MATRIX EQUATION $AXB+CX^TD=E$

Minghui Wang

Department of Mathematics
Qingdao University of Science and Technology, 266061, Qingdao, Shandong, PR China
mhwang@yeah.net

Abstract- In this paper, we propose a new iterative algorithm to solve the matrix equation $AXB+CX^TD=E$. The algorithm can obtain the minimal Frobenius norm solution or the least-squares solution with minimal Frobenius norm. Our algorithm is better than Algorithm II of the paper [M. Wang, etc., Iterative algorithms for solving the matrix equation $AXB+CX^TD=E$, *Appl. Math. Comput.* **187**, 622-629, 2007]

Key Words- Iterative algorithm, Kronecker product, LSQR, Matrix equation, Least Squares

1. INTRODUCTION

Let $\mathbf{R}^{m\times n}$ denote the set of $m\times n$ real matrices. For a matrix $A\in\mathbf{R}^{m\times n}$, we denote its transpose by A^T. $A\otimes B$ is the Kronecker product of two matrices A and B.

In this paper, we will discuss the following matrix equation

$$AXB+CX^TD=E, \tag{1}$$

where $A\in\mathbf{R}^{m\times l}$, $B\in\mathbf{R}^{n\times p}$, $C\in\mathbf{R}^{m\times n}$, $D\in\mathbf{R}^{l\times p}, E\in\mathbf{R}^{m\times p}$ are given matrices, and $X\in\mathbf{R}^{l\times n}$ is unknown matrix to be found.

When Eq.(1) is not consistent, we will consider the least square problem

$$\min_X \| AXB+CX^TD-E \|. \tag{2}$$

Eq.(1) has an important role in system theory and control theory [1, 2], for example, eigenstructure assignment [3], observer design [4], system control with constraint input [5] and fault detection [6]

It is very difficult to solve Eq.(1) and Eq.(2) by means of matrix decomposition. By such way, we can only solve some special cases. In [7, 8, 9, 10], $A^TXB+B^TX^TA=D$ was studied by means of GSVD, CCD and SVD, respectively. By similar methods, $A^TX\pm X^TA=B$ and $AXA^T+AZB^T+BZ^TA^T=D$ were discussed in [11] and [12], respectively.

In [13], the authors have studied Eq.(1) and Eq.(2), and proposed two iterative algorithms, which come from the famous conjugate gradient (CG) method. In this paper, we will propose a new iterative algorithm, which is better than those in [13].

2. A NEW ITERATIVE ALGORITHM

In this section, we present a new iterative algorithm for Eq.(1) and Eq.(2). First, we combine Theorem 4.3.8 and Corollary 4.3.10 of [14] into the following result.

Lemma 1. *Let*

$$P(l,k) = \sum_{i=1}^{l}\sum_{j=1}^{k} E_{ij} \otimes E_{ij}^T .$$

Then, for arbitrary $A \in \mathbf{R}^{m\times n}, B \in \mathbf{R}^{p\times q}$, *we have*

$$\mathrm{v}ec(A^T) = P(m,n)\mathrm{v}ec(A) \tag{3}$$

$$B \otimes A = P(m,p)^T (A \otimes B) P(n,q), \tag{4}$$

where $P(i,j)$ *is a permutation matrix and*

$$P(i,j) = P(j,i)^T = P(j,i)^{-1}.$$

From Lemma 1, it is easy to know that Eq.(1) and Eq.(2) are equivalent to

$$[B^T \otimes A + (D^T \otimes C)P(l,n)]\mathrm{v}ec(X) = \mathrm{v}ec(E) \tag{5}$$

and

$$\min_X \| [B^T \otimes A + (D^T \otimes C)P(l,n)]\mathrm{v}ec(X) - \mathrm{v}ec(E) \|_2, \tag{6}$$

respectively.

Next, we review the LSQR algorithm proposed by Paige and Sauders [15] for solving the following least squares problem:

$$\min_{x\in R^n} \| Mx - f \|_2 \tag{7}$$

with given $M \in R^{m\times n}$ and $f \in R^m$, whose normal equation is

$$M^T M x = M^T f. \tag{8}$$

Algorithm 1 (Algorithm LSQR)

(1)Initialization.

$\beta_1 u_1 = f, \alpha_1 v_1 = M^T u_1, h_1 = v_1, x_0 = 0, \bar{\zeta}_1 = \beta_1, \bar{\rho}_1 = \alpha_1.$

(2)Iteration. For $i = 1, 2, \cdots$

(i) bidiagonalization

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

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

(ii)construct and use Givens rotation

$\rho_i = \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2}$

$c_i = \bar{\rho}_i / \rho_i, s_i = \beta_{i+1} / \rho_i, \theta_{i+1} = s_i \alpha_{i+1}$

$\bar{\rho}_{i+1} = -c_i \alpha_{i+1}, \zeta_i = c_i \bar{\zeta}_i, \bar{\zeta}_{i+1} = s_i \bar{\zeta}_i$

(iii) update x and h

$x_i = x_{i-1} + (\zeta_i / \rho_i) h_i$

$h_{i+1} = v_{i+1} - (\theta_{i+1} / \rho_i) h_i$

(iv) check convergence.

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 minimal norm solution of $Mx = f$. So, if Eq.(8) has a solution $x^* \in R(M^T M) = R(M^T)$, then x^* is the minimum norm solution of Eq.(7). It is obvious that x_k generated by Algorithm LSQR belongs to $R(M^T)$ and this leads to the following result.

Theorem 2. *The solution generated by Algorithm LSQR is the minimum norm solution of Eq.(7) or consistent equation* $Mx = f$.

Now, we can derive our new algorithm, which is based on the above Algorithm LSQR. We have known that Eq.(2) is equivalent to Eq.(6), namely

$$\min_{x \in \mathbf{R}^n} \| Mx - f \|_2 \tag{9}$$

where

$$M = B^T \otimes A + (D^T \otimes C) P(l,n) \in \mathbf{R}^{mp \times nl}, \quad f = \mathrm{v}ec(E) \in \mathbf{R}^{mp}.$$

Next, we will apply ALgorithm LSQR to Eq.(9). The vector iteration of LSQR will be rewritten into matrix form so that the Kronecker product can be released. To this end, it is required to transform the matrix-vector products of Mv and $M^T u$ back to a matrix-matrix form. Notice that we do not want to construct the matrix M explicitly.

Let $\mathrm{m}at(a)$ represent the matrix form of a vector a, or the inverse operation of operator $\mathrm{v}ec$. For $v \in \mathbf{R}^{nl}, u \in \mathbf{R}^{mp}$, let $V = \mathrm{m}at(v) \in \mathbf{R}^{l \times n}, U = \mathrm{m}at(u) \in \mathbf{R}^{m \times p}$, then we have

$$\begin{aligned}
\mathrm{m}at(Mv) &= \mathrm{m}at([B^T \otimes A + (D^T \otimes C) P(l,n)] v) \\
&= \mathrm{m}at((B^T \otimes A) \mathrm{v}ec(V) + (D^T \otimes C) P(l,n) \mathrm{v}ec(V)) \\
&= AVB + CV^T D. \\
\mathrm{m}at(M^T u) &= \mathrm{m}at([B \otimes A^T + P(l,n)^T (D \otimes C^T)] u) \\
&= \mathrm{m}at((B \otimes A^T) \mathrm{v}ec(U) + (P(l,n)^T (D \otimes C^T) P(p,m)) P(m,p) \mathrm{v}ec(U)) \\
&= \mathrm{m}at((B \otimes A^T) \mathrm{v}ec(U) + (C^T \otimes D) \mathrm{v}ec(U^T)) \\
&= A^T U B^T + D U^T C
\end{aligned}$$

By combining the above equalities and the LSQR algorithm, we now propose the following matrix-form LSQR algorithm to evaluate the minimum-norm solution of Eq.(2) and consistent equation Eq.(1).

Algorithm 2 (LSQR-M)

(1). Initialization.

$X_0 = 0(\in \mathbf{R}^{l\times n}), \quad \beta_1 = \| E \|_F, \quad U_1 = E / \beta_1,$

$\bar{V}_1 = A^T U_1 B^T + DU^T C, \quad \alpha_1 = \| \bar{V}_1 \|_F, \quad V_1 = \bar{V}_1 / \alpha_1,$

$H_1 = V_1, \quad \bar{\zeta}_1 = \beta_1, \quad \bar{\rho}_1 = \alpha_1.$

(2). Iteration. For $i = 1, 2, \cdots$

$\bar{U}_{i+1} = AV_i B + CV_i^T D - \alpha_i U_i,$

$\beta_{i+1} = \| \bar{U}_{i+1} \|_F, \quad U_{i+1} = \bar{U}_{i+1} / \beta_{i+1}$

$\bar{V}_{i+1} = A^T U_{i+1} B^T + DU_{i+1}^T C - \beta_{i+1} V_i,$

$\alpha_{i+1} = \| \bar{V}_{i+1} \|_F, \quad V_{i+1} = \bar{V}_{i+1} / \alpha_{i+1},$

$\rho_i = \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2},$

$c_i = \bar{\rho}_i / \rho_i, \quad s_i = \beta_{i+1} / \rho_i, \quad \theta_{i+1} = s_i \alpha_{i+1},$

$\bar{\rho}_{i+1} = -c_i \alpha_{i+1}, \quad \zeta_i = c_i \bar{\zeta}_i, \quad \bar{\zeta}_{i+1} = s_i \bar{\zeta}_i,$

$X_i = X_{i-1} + (\zeta_i / \rho_i) H_i,$

$H_{i+1} = V_{i+1} - (\theta_{i+1} / \rho_i) H_i,$

(3). check convergence.

Algorithm II(CG) of [13] need eight matrix-matrix products per iterative step, our algorithm need only four matrix-matrix products. Therefore, calculated amount of our algorithm should be less than that of Algorithm II. Furthermore, when solving bad-condition system, our algorithm work more reliably and more stably.

3. NUMERICAL EXAMPLES

In this paper, we will use some numerical examples to illustrate the efficiency of our algorithm. The computations are carried out at a PC computer, with software MATLAB 7.0. The machine precision is $2^{-52} \sim 2^{-16}$.

Example 1. Consider inconsistent matrix equation

$$AX + X^T D = E, \tag{10}$$

where

$$A=\begin{pmatrix}-10 & 7 & 0 & 6\\ 13 & -9 & 8 & 23\\ 0 & -1 & 24 & 8\\ -7 & 10 & 6 & 0\\ 19 & 0 & -9 & -12\end{pmatrix}, = \begin{pmatrix}9 & -14 & 5 & 0 & 3\\ 8 & 0 & 14 & 9 & -1\\ -9 & 18 & 6 & -17 & 0\\ 0 & -28 & -17 & 14 & 7\end{pmatrix},$$

$$E=\begin{pmatrix}-1 & -21 & 11 & 9 & 12\\ 3 & 11 & 43 & 4 & 44\\ 39 & 17 & -9 & 37 & 40\\ 17 & -15 & 17 & 1 & 18\\ 6 & -26 & 61 & 4 & 7\end{pmatrix}.$$

The normal equation of Eq.(10) is

$$A^T AX + A^T X^T D + DX^T A^T + DD^T X = A^T E + DE^T. \tag{11}$$

By Algorithm II(CG) of [13], after 75 iterative steps, we have

$$X_{76}=\begin{pmatrix}0.8919 & -2.1007 & 0.3799 & 2.3249 & 1.8777\\ 1.0846 & 2.6603 & 1.2421 & 1.0505 & 0.4321\\ 0.2809 & 0.1802 & -1.8048 & 1.6891 & 2.1657\\ 0.2263 & -6.6782 & -3.7120 & 1.7252 & 2.5619\end{pmatrix},$$

which satisfy

$$\| R_{76} \| = \| A^T E + DE^T - A^T AX_{76} - A^T X_{76}^T D - DX_{76}^T A^T - DD^T X_{76} \| = 6.94422\times 10^{-11}$$

and

$$\| E - AX_{76} - X_{76}^T D \| = 35.4543.$$

By our algorithm, after 24 iterative steps, we can obtain the same solution X_{24} as the above X_{76}, and $\| R_{24} \| = 1.5630\times 10^{-11}$.

Figure 1 gives an intuitive comparison about $log10(\| R_k \|)$ s of two algorithms.

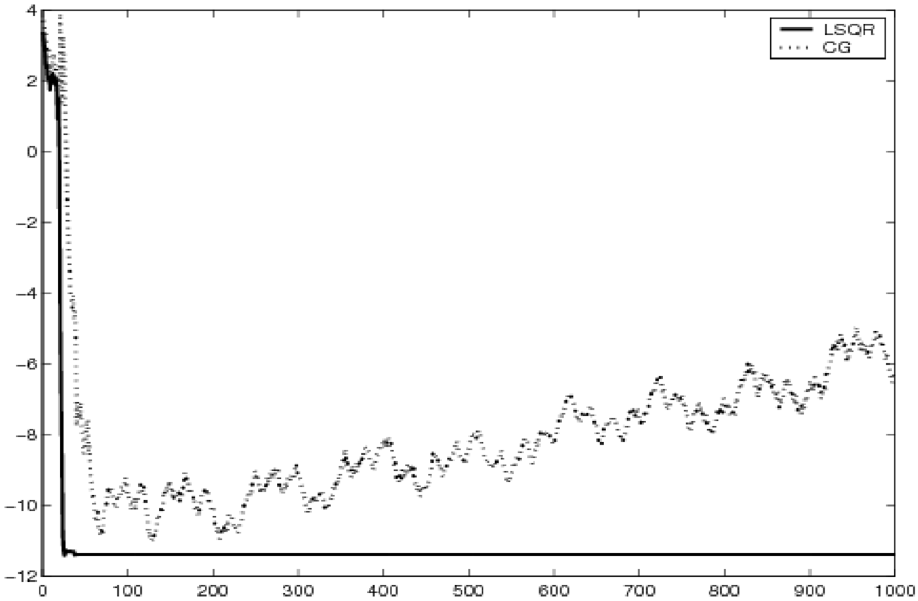

Figure 1. Comparison about $log10(\| R_k \|)$s of two algorithms in Example 1.

From Figure 1, we can see that our algorithm is more reliable and more stable than Algorithm II(CG) in [13].

Example 2. Let $A = Hilb(8), D = pascal(8), E = ones(8,8)$, where *Hilb, pascal* and *ones* are functions in Matlab. Then inconsistent equation Eq.(10) is bad-conditioned.

Figure 2 gives an intuitive comparison about $log10(\| R_k \|)$s of two algorithms.

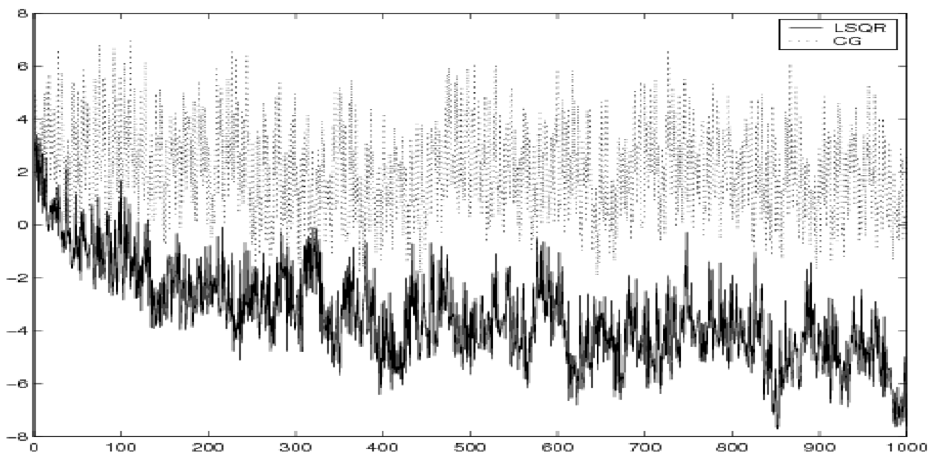

Figure 2. Comparison about $log10(\| R_k \|)$s of two algorithms in Example 2.

Although our algorithm can obtain better results than Algorithm II(CG) in [13], its stability is not good. For very ill-conditioned system, our algorithm cannot also work very well and therefore, preconditioning techniques must be considered.

Acknowledgments-This work is supported by the National Natural Science Foundation of China (Grant No:11001144), the Science and Technology Program of Shandong Universities of China (J11LA04, J10LA06) and the Research Award Fund for outstanding young scientists of Shandong Province in China(BS2012DX009)

5. REFERENCES

1. H. Fujioka, State covariance assignment problem with measurement noise: a unified approach based on a symmetric matrix equation, *Linear Algebra and its Applications* **124**, 102-114, 1994.
2. K. Yasuda and R. Skelton, Assigning controllability and observability gramians in feedback control, *Journal of Guidance Control* **14**, 878-C885, 1990.
3. L. Fletcher, J. Kuatsky and N. Nichols, Eigenstructure assignment in descriptor systems, *IEEE Trans. Auto. Control* **31**, 1138-1141, 1986.
4. L. Dai, *Singular control systems*, Verlag, 1989.
5. G.R. Duan, The solution to the matrix equation $AV+BW=EVJ+R$, *Applied Mathematics Letters* **17**, 1197-1202, 2004.
6. P.M. Frank, Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy- a survey and some new results, *Automatica* **26**, 459-474, 1990.
7. Y. Yuan and H. Dai, The least squares solution with the minimum norm of matrix equation $A^T XB+B^T X^T A=D$, *Applied Mathematics-A Journal of Chinese Universities* **27**, 232-238, 2005.
8. Y. Yuan, The least-squares solutions of a matrix equation, *Applied Mathematics-A Journal of Chinese Universities* **23**, 324-329, 2001.
9. Y. Yuan, The minimum norm solutions of two classes of matrix equations, *Applied Mathematics-A Journal of Chinese Universities* **24**, 127-134, 2002
10. K. Chu, Symmetric solutions of linear matrix equation by matrix decomposition, *Linear Algebra and its Applications* **119**, 35-50, 1989.
11. H. Braden, The equations $A^T X \pm X^T A = B$, *SIAM Journal on Matrix Analysis and Applications* **20**, 295-302, 1998.
12. Y. Yuan and H. Liu, The minimum norm solutions of matrix equations $AXA^T+AYB^T+AZB^T=D$ and $AXA^T+AZB^T BZ^T A^T=D$, *Journal of Nanjing University Mathematical Biquarterly* **23**, 79-87, 2006.
13. M. Wang, X. Cheng and M. Wei, Iterative algorithms for solving the matrix equation $AXB+CX^T D=E$, *Applied Mathematics and Computation* **187**, 622-629, 2007.
14. R. Horn, C. Johnson, *Topics in matrix analysis*, Cambridge University Press, Cambridge, UK, 1991.
15. C. Paige and A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software* **8**, 43-71, 1982.