Article ID: 1000-5641(2008)03-0030-07

Iterative algorithm for solving least Frobenius norm problem of an inconsistent matrix equation pair

XU Xiang-jian [1,2], WANG Ming-hui [2], WEI Mu-sheng [2]

(1. *School of Science, Nantong University, Nantong Jiangsu* 226007, *China*; 2. *Department of Mathematics, East China Normal University, Shanghai* 200062, *China*)

Abstract: This paper presented an iterative algorithm for solving the least Frobenius norm problem of inconsistent matrix equation pair $(\boldsymbol{AXB},\boldsymbol{CXD})=(\boldsymbol{E},\boldsymbol{F})$ with a real matrix $\boldsymbol{X}$. By this algorithm, for any (special) initial matrix $\boldsymbol{X}_0$, a solution (the minimal Frobenius norm solution) can be obtained within finite iteration steps in the absence of roundoff errors. The numerical examples verify the efficiency of the algorithm.

Key words: iterative algorithm; Kronecker product; matrix equation pair

CLC number: O24; O151 **Document code**: A

一类不相容矩阵方程对最小 Frobenius 范数问题的迭代算法

徐相建 [1,2], 王明辉 [2], 魏木生 [2]

(1. 南通大学 理学院, 江苏 南通 226007; 2. 华东师范大学 数学系, 上海 200062)

摘要: 提出了关于不相容矩阵方程对$(\boldsymbol{AXB},\boldsymbol{CXD})=(\boldsymbol{E},\boldsymbol{F})$最小 Frobenius 范数问题的一个迭代算法. 对于任意的初始矩阵$\boldsymbol{X}_0$,在没有舍入误差的情况下,运用此算法能在有限步内得到方程对在 Frobenius 范数意义下的最小解. 数值例子表明所提出算法的有效性.

关键词: 迭代算法; Kronecker 积; 矩阵方程对

0 Introduction

In this paper we use the following notation. $\mathbf{R}^{m\times n}$ is the set of $m\times n$ real matrices, $\boldsymbol{A}^{\mathrm{T}}$ is the transpose matrix of a matrix $\boldsymbol{A}$, $\boldsymbol{A}\otimes\boldsymbol{B}$ is the Kronecker product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. In $\mathbf{R}^{m\times n}$, we define the inner product of two matrices $\boldsymbol{A},\boldsymbol{B}\in\mathbf{R}^{m\times n}$ as $\langle\boldsymbol{A},\boldsymbol{B}\rangle=\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{A})$. Then the matrix norm of $\boldsymbol{A}$ induced by this inner product is Frobeniusnorm and

收稿日期:2007-06
基金项目:国家自然科学基金(10771073)
第一作者:徐相建,男,硕士. E-mail:xiangjianxu@hotmail.com.
通讯作者:魏木生,男,教授. E-mail:mwei@math.ecnu.edu.cn.

denoted by $\| A \|$. For all $A,B\in \mathbf{R}^{p\times q}$, if $\langle A,B\rangle = \mathrm{trace}(B^{T}A) = 0$, then we say that A and B are orthogonal.

The author of[1]proposed an iterative algorithm to solve the matrix equation pair $(AXB,CXD) = (E,F)$ when it is consistent. In this paper, we propose a different iterative algorithm to solve the least Frobenius norm problem as follows:

$$\min_{X\in \mathbf{R}^{p\times q}} (\| AXB - E \|^{2} + \| CXD \| - F \|^{2}), \tag{0}$$

where $A\in \mathbf{R}^{m\times p}, B\in \mathbf{R}^{q\times n}$, $C\in \mathbf{R}^{s\times p}, D\in \mathbf{R}^{q\times t}, E\in \mathbf{R}^{m\times n}$ and $F\in \mathbf{R}^{s\times t}$.

It is easy to see that if $(AXB, CXD) = (E,F)$ is a consistent matrix equation pair, then its solutions are also the solutions of problem (0).

This paper is organized as follows. In Section 1, we propose an iterative algorithm to solve problem (0) and prove that the algorithm can obtain (the minimal Frobenius norm) a solution for any (special) initial matrix X_0 in finite steps. In Section 2, two numerical examples are given to show that our iterative algorithm is efficient. Finally, we provide some concluding remarks in Section 3.

1　An iterative algorithm for solving (0)

In this section, we first establish the following equivalent result of (0)

Iemma 1.1　The least Frobenius norm problem (0) is equivalent to the following consistent matrix equation:

$$A^{T}AXBB^{T} + C^{T}CXDD^{T} = A^{T}EB^{T} + C^{T}FD^{T}. \tag{1}$$

Proof　Because the least Frobenius norm problem (0) is equivalent to

$$\min_{X\in \mathbf{R}^{p\times q}} (\| (B^{T} \otimes A)\mathrm{vec}(X) - \mathrm{vec}(E)^{2} + (D^{T} \otimes C)\mathrm{vec}(X) - \mathrm{vec}(F) \|^{2}), \text{i. e.}$$

$$\min_{X\in R^{p\times q}} \left\| \begin{pmatrix} BT \otimes A \\ D^{T} \otimes C \end{pmatrix} \mathrm{vec}(X) - \begin{pmatrix} vec(E) \\ \mathrm{vec}(F \end{pmatrix} \right\|^{2} \tag{2}$$

Furthermore, the solution set of (2) is also the solution set of the following equation:

$$(BB^{T} \otimes A^{T}A + DD^{T} \otimes C^{T}C)\mathrm{vec}(X) = (B \otimes A^{T})\mathrm{vec}(E) + (D \otimes C^{T})\mathrm{vec}(F) \tag{3}$$

which is equivalent to (1). Therefore the assertion of the lemma holds.

We now state our algorithm.

Algorithm 1.1

(1) Input matrices A,B,C,D,E,F and X_0;

(2) Compute

$$R_0 = A^{T}EB^{T} + C^{T}FD^{T} - A^{T}AX_0BB^{T} - C^{T}CX_0DD^{T},$$
$$P_0 = A^{T}AR_0BB^{T} + C^{T}CR_0DD^{T},$$
$$Q_0 = P_0;\ k = 0;$$

(3) If $R_k=0$, then stop; else $k=k+1$;

(4) Compute

$$X_k = X_{k-1} + \frac{\| R_{k-1} \|^{2}}{\| Q_{k-1} \|^{2}} Q_{k-1};$$

$$R_k = A^T E B^T + C^T F D^T - A^T A X_k B B^T - C^T C X_k D D^T,$$
$$P_k = A^T A R_k B B^T + C^T C R_k D D^T,$$
$$Q_k = P_k - \frac{\text{trace}(P_k^T Q_{k-1})}{\| Q_{k-1} \|^2} Q_{k-1};$$

(5) Return to Step 3.

For Algorithm 1.1, we have the following basic properties.

Lemma1.2 For any initial matrix X_0, the sequences $\{R_i\}$ and $\{Q_i\}$ generated by Algorithm 1.1 satisfy

$$\text{trace}(R_{i+1}{}^T R_j) = \text{trace}(R_i{}^T R_j) - \frac{\| R_i \|^2}{\| Q_i \|^2} \text{trace}(Q_i{}^T P_j),\ i,j = 0,1,2,\cdots, \tag{4}$$

Proof Denote $G = A^T E B^T + C^T F D^T$, then we have

$$\text{trace}(R_{i+1}{}^T R_j) = \text{trace}[(G - A^T A X_{i+1} B B^T - C^T C X_{i+1} D D^T)^T R_j]$$
$$= \text{trace}[(G - A^T A X_i + \frac{\| R_i \|^2}{\| Q_i \|^2} Q_i) B B^T - C^T C (X_i + \frac{\| R_i \|^2}{\| Q_i \|^2} Q_i) D D^T)^T R_j]$$
$$= \text{trace}(R_i{}^T R_j) - \frac{\| R_i \|^2}{\| Q_i \|^2} [\text{trace}(B B^T Q_i{}^T A^T A R_j) + \text{trace}(D D^T Q_i{}^T C^T C R_j)]$$
$$= \text{trace}(R_i{}^T R_j) - \frac{\| R_i \|^2}{\| Q_i \|^2} \text{trace}(Q_i{}^T P_j).$$

The following lemma shows that if $R_i \neq 0$, then $Q_i \neq 0 (i=0,1,2,\cdots)$.

Lemma 1.3 Assume X^* is a solution of (1), then the sequences $\{R_i\}$ and $\{Q_i\}$ generated by Algorithm 1.1 satisfy

$$\text{trace}[(X^* - X_k) Q_k{}^T] = \| R_k \|^2, \quad k = 0,1,2,\cdots. \tag{5}$$

Proof We prove the conclusion by induction. When $k=0$, we have

$$\text{trace}[(X^* - X_0) Q_0^T] = \text{trace}[(X^* - X_0)(B B^T R_0^T A^T A + D D^T R_0^T C^T C)]$$
$$= \text{trace}(X^* B B^T R_0^T A^T A + X^* D D^T R_0^T C^T C) - \text{trace}(X_0 B B^T R_0^T A^T A + X_0 D D^T R_0^T C^T C)$$
$$= \text{trace}[R_0{}^T (A^T A X^* B B^T + C^T C X^* D D^T)] - \text{trace}[R_0^T (A^T A X_0 B B^T + C^T C X_0 D D^T)]$$
$$= \text{trace}[R_0^T G] - \text{trace}[R_0{}^T (G - R_0)]$$
$$= \text{trace}(R_0^T R_0)$$
$$= \| R_0 \|^2.$$

Assume that equality in (5) holds for $1 \leqslant k \leqslant n$, then we have

$$\text{trace}[(X^* - X_{n+1}) Q_n^T] = \text{trace}[(X^* - X_n - \frac{\| R_n \|^2}{\| Q_n \|^2} Q_n) Q_n{}^T]$$
$$= \text{trace}[(X^* - X_n) Q_n^T] - \frac{\| R_n \|^2}{\| Q_n \|^2} \text{trace}(Q_n Q_n{}^T)$$
$$= \| R_n \|^2 - \| R_n \|^2 = 0.$$

So

$$\text{trace}[(X^* - X_{n+1}) Q_{n+1}{}^T] = \text{trace}[(X^* - X_{n+1})(P_{n+1} - \frac{\text{trace}(P_{n+1}{}^T Q_n)}{\| Q_n \|^2} Q_n)^T]$$
$$= \text{trace}[(X^* - X_{n+1}) P_{n+1}{}^T]$$

$$= \mathrm{trace}[(\boldsymbol{X}^{*} - \boldsymbol{X}_{n+1})(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{R}_{n-1}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{R}_{n+1}\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}})^{\mathrm{T}}]$$

$$= \mathrm{trace}\left(\boldsymbol{X}^{*}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{R}_{n+1}{}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{X}^{*}\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{R}_{n+1}{}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}\right)$$

$$- \mathrm{trace}[\boldsymbol{X}_{n-1}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{R}_{n-1}{}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{X}_{n-1}\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{R}_{n+1}{}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}]$$

$$= \mathrm{trace}(\boldsymbol{R}_{n+1}{}^{\mathrm{T}}\boldsymbol{G}) - \mathrm{trace}[\boldsymbol{R}_{n-1}{}^{\mathrm{T}}(\boldsymbol{G} - \boldsymbol{R}_{n+1})]$$

$$= \|\boldsymbol{R}_{n-1}\|^{2}.$$

By the induction principle, the conclusion holds.

Lemma 1.4 Assume that the sequences $\{\boldsymbol{R}_i\}$ *and* $\{\boldsymbol{Q}_i\}$ are generated by Algorithm 1.1, where $\boldsymbol{R}_i \neq 0 (i=0,1,2,\cdots,k)$, then we have

$$\mathrm{trace}(\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{R}_j) = 0,\quad \mathrm{trace}(\boldsymbol{Q}_i^{\mathrm{T}}\boldsymbol{Q}_j) = 0,\quad (i,j = 0,1,2,\cdots,k,\ i \neq j). \tag{6}$$

Proof By Lemma 1.2, when $k=1$, we have

$$\mathrm{trace}(\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{R}_0) = \mathrm{trace}(\boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{R}_0) - \frac{\|\boldsymbol{R}_0\|^2}{\|\boldsymbol{Q}_0\|^2}\mathrm{trace}(\boldsymbol{Q}_0^{\mathrm{T}}\boldsymbol{P}_0)$$

$$= \|\boldsymbol{R}_0\|^2 - \frac{\|\boldsymbol{R}_0\|^2}{\boldsymbol{Q}_0\|^2}\|\mathrm{trace}(\boldsymbol{Q}_0^{\mathrm{T}}\boldsymbol{Q}_0) = 0,$$

$$\mathrm{trace}(\boldsymbol{Q}_1^{\mathrm{T}}\boldsymbol{Q}_0) = \mathrm{trace}[(\boldsymbol{P}_1 - \frac{\mathrm{trace}(\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{Q}_0)}{\|\boldsymbol{Q}_0\|^2}\boldsymbol{Q}_0)^{\mathrm{T}}\boldsymbol{Q}_0]$$

$$= \mathrm{trace}(\boldsymbol{P}_1{}^{\mathrm{T}}\boldsymbol{Q}_0) - \frac{\mathrm{trace}(\boldsymbol{P}_1{}^{\mathrm{T}}\boldsymbol{Q}_0)}{\|\boldsymbol{Q}_0\|^2}\mathrm{trace}(\boldsymbol{Q}_0^{\mathrm{T}}\boldsymbol{Q}_0) = 0.$$

Assume that equality in (6) holds for $1 \leqslant k \leqslant n$, then we have

$$\mathrm{trace}(\boldsymbol{R}_{n+1}{}^{\mathrm{T}}\boldsymbol{R}_j) = \mathrm{trace}(\boldsymbol{R}_n{}^{\mathrm{T}}\boldsymbol{R}_j) - \frac{\|\boldsymbol{R}_n\|^2}{\|\boldsymbol{Q}_n\|^2}\mathrm{trace}(\boldsymbol{Q}_n^{\mathrm{T}}\boldsymbol{P}_j)$$

$$= \mathrm{trace}(\boldsymbol{R}_n^{\mathrm{T}}\boldsymbol{R}_j) - \frac{\|\boldsymbol{R}_n\|^2}{\|\boldsymbol{Q}_n\|^2}\mathrm{trace}[\boldsymbol{Q}_n^{\mathrm{T}}(\boldsymbol{Q}_j - \frac{\mathrm{trace}(\boldsymbol{P}_j^{\mathrm{T}}\boldsymbol{Q}_{j-1})}{\|\boldsymbol{Q}_{j-1}\|^2}\boldsymbol{Q}_{j-1})]$$

$$= 0, (j = 0,1,2,\cdots,n-1).$$

When $j=n$,

$$\mathrm{trace}(\boldsymbol{R}_{n+1}^{\mathrm{T}}\boldsymbol{R}_n) = \mathrm{trace}(\boldsymbol{R}_n^{\mathrm{T}}\boldsymbol{R}_n) - \frac{\|\boldsymbol{R}_n\|^2}{\|\boldsymbol{Q}_n\|^2}\mathrm{trace}(\boldsymbol{Q}_n^{\mathrm{T}}\boldsymbol{P}_n)$$

$$= \|\boldsymbol{R}_n\|^2 - \frac{\|\boldsymbol{R}_n\|^2}{\|\boldsymbol{Q}_n\|^2}\mathrm{trace}[\boldsymbol{Q}_n^{\mathrm{T}}(\boldsymbol{Q}_n - \frac{\mathrm{trace}(\boldsymbol{P}_n^{\mathrm{T}}\boldsymbol{Q}_{n-1})}{\|\boldsymbol{Q}_{n-1}\|^2}\boldsymbol{Q}_{n-1})]$$

$$= 0.$$

$$\mathrm{trace}(\boldsymbol{Q}_{n-1}^{\mathrm{T}}\boldsymbol{Q}_j) = \mathrm{trace}[(\boldsymbol{P}_{n-1} - \frac{\mathrm{trace}(\boldsymbol{P}_{n+1}{}^{\mathrm{T}}\boldsymbol{Q}_n)}{\|\boldsymbol{Q}_n\|^2}\boldsymbol{Q}_n)^{\mathrm{T}}\boldsymbol{Q}_j]$$

$$= \mathrm{trace}(\boldsymbol{P}_{n+1}^{\mathrm{T}}\boldsymbol{Q}_j) - \frac{\mathrm{trace}(\boldsymbol{P}_{n+1}{}^{\mathrm{T}}\boldsymbol{Q}_n)}{\|\boldsymbol{Q}_n\|^2}\mathrm{trace}(\boldsymbol{Q}_n^{\mathrm{T}}\boldsymbol{Q}_j). \tag{7}$$

When $j=n$, the right side of above equality (7) equal to 0.

When $j=0,1,2,\cdots,n-1$, by equality (7), we have

$$\mathrm{trace}(\boldsymbol{Q}_{n+1}^{\mathrm{T}}\boldsymbol{Q}_j) = \mathrm{trace}(\boldsymbol{P}_{n+1}^{\mathrm{T}}\boldsymbol{Q}_j) = \mathrm{trace}(\boldsymbol{Q}_j^{\mathrm{T}}\boldsymbol{P}_{n+1})$$

$$= \frac{\|\boldsymbol{Q}_j\|^2}{\|\boldsymbol{R}_j\|^2}[\mathrm{trace}(\boldsymbol{R}_j^{\mathrm{T}}\boldsymbol{R}_{n+1}) - \mathrm{trace}(\boldsymbol{R}_{j+1}{}^{\mathrm{T}}\boldsymbol{R}_{n+1})] = 0, \tag{8}$$

where the equality (8) is because of Lemma 1.2.

By the induction principle, the conclusion of the lemma holds.

Theorem 1.5 For any initial matrix $\boldsymbol{X}_0\in\mathbf{R}^{p\times q}$, the sequence $\{\boldsymbol{X}_k\}$ generated by Algorithm 1.1 converges to an exact solution of (1.1) in at most pq iteration steps.

Proof Assume that $\boldsymbol{R}_i\neq 0$ for $i=0,1,2,\cdots,pq$, then by Lemma 1.3, $\boldsymbol{Q}_i\neq 0$ for $i=0,1,2,\cdots,pq$, so $\boldsymbol{X}_{pq},\boldsymbol{R}_{pq}$ can be obtained by Algorithm 1.1, then by Lemma 1.4, $\{\mathbf{R}_0,\mathbf{R}_1,\cdots,\mathbf{R}_{pq}\}$ is an orthogonal sequence in $\mathbf{R}^{p\times q}$, this is impossible, because that for $k=0,1,2,\cdots,pq$, we have

$$\mathrm{vec}(\mathbf{R}_k)=\mathrm{vec}(\boldsymbol{G})-(\boldsymbol{BB}^{\mathrm{T}}\otimes\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{DD}^{\mathrm{T}}\otimes\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})\mathrm{vec}(\boldsymbol{X}_k)\in\mathbf{R}^{pq}.$$

Therefore $\boldsymbol{R}_{pq}=0$ and $\boldsymbol{X}_{pq}$ is an exact solution of the matrix equation (1).

It is known that if the consistent system of linear equation $\boldsymbol{A}x=b$ has a solution $x^*\in\boldsymbol{R}(\boldsymbol{A}^{\mathrm{T}})$, where $\boldsymbol{A}\in\mathbf{R}^{m\times n}$, $b\in\mathbf{R}^m$, then x^* is the unique minimal Frobenius norm solution of $\boldsymbol{A}x=b$. In Algorithm 1.1, if we choose the initial matrix $\boldsymbol{X}_0=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{AHBB}^{\mathrm{T}}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{CHDD}^{\mathrm{T}})$, *where* $\boldsymbol{H}\in\mathbf{R}^{p\times q}$ is an arbitrary matrix, then

$$\begin{aligned}\mathrm{vec}(\boldsymbol{X}_0)&=(\boldsymbol{BB}^{\mathrm{T}}\otimes\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\mathbf{DD}^{\mathrm{T}}\otimes\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})\mathrm{vec}(\boldsymbol{H})\in\boldsymbol{R}(\boldsymbol{BB}^{\mathrm{T}}\otimes\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{DD}^{\mathrm{T}}\otimes\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})\\&=\boldsymbol{R}[(\boldsymbol{BB}^{\mathrm{T}}\otimes\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{DD}^{\mathrm{T}}\otimes\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})^{\mathrm{T}}].\end{aligned}$$

It is not difficult to verify that all $\boldsymbol{X}_k$ generated by Algorithm 1.1 satisfy $\mathrm{vec}(\boldsymbol{X}_k)\in\mathbf{R}[(\boldsymbol{BB}^{\mathrm{T}}\otimes\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\boldsymbol{DD}^{T}\otimes\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})^{\mathrm{T}}]$. Therefore, with initial matrix $\boldsymbol{X}_0$, then solution $\boldsymbol{X}^*$ obtained by Algorithm 1.1 is the minimal Frobenius norm solution.

Restate the above observations as the following theorem.

Theorem 1.6 If we choose the initial matrix $\boldsymbol{X}_0=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{AHBB}^{\mathrm{T}}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{CHDD}^{\mathrm{T}})$, then the solution $\boldsymbol{X}^*$ obtained by Algorithm 1.1 is the minimal Frobenius norm solution of (1), where $\boldsymbol{H}\in\mathbf{R}^{p\times q}$ is arbitrary, specially, we can choose $\boldsymbol{X}_0=0\in\mathbf{R}^{p\times q}$.

2 Numerical example

This section gives two examples to illustrate our results. All the following tests are performed by MATLAB 7.0, and the initial iterative matrix is chosen as $\boldsymbol{X}_0=0$.

Example 2.1 Given matrices $\boldsymbol{A},\boldsymbol{B},\boldsymbol{E}$ and $\boldsymbol{C},\boldsymbol{D},\boldsymbol{F}$ as follows:

$$\boldsymbol{A}=\begin{pmatrix}-10&7&0&6\\13&-9&8&23\\0&-1&24&8\\-7&10&6&0\\19&0&-9&-12\end{pmatrix},\quad\boldsymbol{B}=\begin{pmatrix}9&8&-9\\-14&0&18\\5&14&6\\0&9&-17\\3&-1&0\end{pmatrix},\quad\boldsymbol{E}=\begin{pmatrix}9&80&-6\\100&1\,020&-75\\93&920&-62\\25&250&-18\\-5&-60&4\end{pmatrix},$$

$$\boldsymbol{C}=\begin{pmatrix}-1&-21&11&9\\3&11&43&4\\39&17&-9&37\end{pmatrix},\quad\boldsymbol{D}=\begin{pmatrix}17&6\\-15&-26\\17&61\\1&4\\18&7\end{pmatrix},\quad\boldsymbol{F}=\begin{pmatrix}-75&-50\\2370&3020\\310&4010\end{pmatrix}.$$

It can be verified that the matrix equations $\boldsymbol{AXB}=\boldsymbol{E}$ and $\boldsymbol{CXD}=\boldsymbol{F}$ are all inconsistent. By Lemma 1.1, we can obtain an approximation of the unique minimal Frobenius norm solution,

$$\tilde{\boldsymbol{X}}^{*}=\begin{pmatrix} 1.0481477e+00 & 5.7050135e-01 & 1.0322990e+00 & 5.3747750e-01 \\ 9.4932555e-01 & 8.4110781e-01 & 9.2974062e-01 & 8.3745073e-01 \\ 1.0397843e+00 & 1.1411831e+00 & 9.4812920e-01 & 1.1070043e+00 \\ 9.8881412e-01 & 4.9749231e-01 & 1.0429709e+00 & 5.0045105e-01 \end{pmatrix}.$$

We apply Algorithm 1.1 to compute $\boldsymbol{X}_k$. The result is provided in Fig. 1, where $r_k=\| \boldsymbol{G}\text{-}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{AX}_k\boldsymbol{BB}^{\mathrm{T}}-\boldsymbol{C}^{\mathrm{T}}\boldsymbol{CX}_k\boldsymbol{DD}^{\mathrm{T}} \|$ and

$$\boldsymbol{X}_{150}=\begin{pmatrix} 1.0481 & 0.5705 & 1.0323 & 0.5375 & -1.2356 \\ 0.9493 & 0.8411 & 0.9297 & 0.8375 & 0.4727 \\ 1.0398 & 1.1412 & 0.9481 & 1.1070 & 1.6399 \\ 0.9888 & 0.4975 & 1.0430 & 0.5005 & -1.4584 \end{pmatrix},$$

$r_1=\| \boldsymbol{E}-\boldsymbol{AXB} \| =6.9431$,

$r_2=\| \boldsymbol{F}-\boldsymbol{CXD} \| =1.8644\mathrm{e}-009$.

By the way, our algorithm is also applicable for the consistent matrix equation pair.

Example 2.2 Choose matrices $\boldsymbol{A},\boldsymbol{B}$ and $\boldsymbol{C},\boldsymbol{D}$ as same in Example 2.1 and

$$\boldsymbol{E}=\begin{pmatrix} 9 & 90 & -6 \\ 105 & 1050 & -70 \\ 93 & 930 & -62 \\ 27 & 270 & -18 \\ -6 & -60 & 4 \end{pmatrix},\quad \boldsymbol{F}=\begin{pmatrix} -76 & -104 \\ 2318 & 3172 \\ 3192 & 4368 \end{pmatrix}.$$

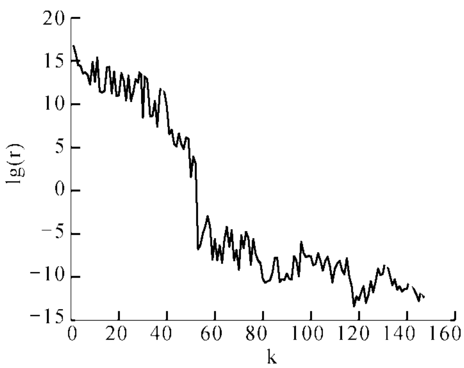

Fig. 1 The residual in logarithm

It can be verified that the matrix equations $\boldsymbol{AXB}=\boldsymbol{E}$ and $\boldsymbol{CXD}=\boldsymbol{F}$ are all consistent and has a solution of 4×5 matrix with all elements equal to one. After applying Algorithm 1.1, we obtain

$$\boldsymbol{X}_{150}=\begin{pmatrix} 1.0327 & 1.0309 & 0.9812 & 1.0088 & 1.0776 \\ 0.9881 & 0.9888 & 1.0068 & 0.9968 & 0.9718 \\ 1.0034 & 1.0032 & 0.9981 & 1.0009 & 1.0080 \\ 0.9718 & 0.9733 & 1.0162 & 0.9924 & 0.9331 \end{pmatrix},$$

$r_1 = \| \boldsymbol{E} - \boldsymbol{AXB} \| = 3.519\ 4\text{e}-009$,

$r_2 = \| \boldsymbol{F} - \boldsymbol{CXD} \| = 1.180\ 7\text{e}-009$.

3 Concluding remarks

In this paper, we have proposed an iterative algorithm for solving the least Frobenius norm problem (0). We proved that, with any initial matrix $\boldsymbol{X}_0$, the iterative sequence $\{\boldsymbol{X}_k\}$ generated by our algorithm converges to a solution of (0), within finite steps in the absence of roundoff errors. Numerical examples we tested using MATLAB verify our theoretical results.

[References]

[1] SHENG X, CHEN G. A finite iterative method for solving a pair of linear matrix equations $(\boldsymbol{AXB}, \boldsymbol{CXD}) = (\boldsymbol{E}, \boldsymbol{F})$ [J]. Appl Math Comput, 2007, 189: 1350-1358.

[2] GOLUB G H, VAN LOAN C F. Matrix Computations[M]. 3rd Edition. Baltimore: Johns Hopkins Univ Press, 1996.

[3] WEI M. The Theory and Computation of the Generalized Least Squares Problem[M]. Beijing: Science Press, 2006.

[4] PENG Y. An iterative method for the least squares symmetric solution of the linear matrix equation $\boldsymbol{AXB} = \boldsymbol{C}$ [J]. Appl Math Comput, 2005, 170: 711-723.

[5] PENG Y, HU X, ZHANG L. An iteration method for the symmetric solutions and the optimal approximation solution of the matrix equation $\boldsymbol{AXB} = \boldsymbol{C}$ [J]. Appl Math Comput, 2005, 160(3): 763-777.

(上接第 29 页)

[4] VASIL'VA A B, BUTUZOV V F. Asymptotic Expansions of Solutions of Singularly Perturbed Equations[M]. Nauka: Moscow, 1973.

[5] HORN R A, JOHSON C R. Topics in Matrix Analysis[M]. Cambridge: Cambridge University Press, 1991: 459-489.

[6] ESIPOVA V A. Asymptotic properties of general boundary value problems for singularly perturbed conditionally stable systems of ordinary differential equations[J]. Differential Equations, 1975, 11(11): 1 457-1465.