J. Appl. Math. & Informatics Vol. **27***(2009), No. 1 - 2, pp. 1 - 12*
Website: http://www.kcam.biz

ITERATIVE ALGORITHMS FOR THE LEAST-SQUARES SYMMETRIC SOLUTION OF $AXB = C$ WITH A SUBMATRIX CONSTRAINT†

MINGHUI WANG* AND YAN FENG

ABSTRACT. Iterative algorithms are proposed for the least-squares symmetric solution of $AXB = E$ with a submatrix constraint. We characterize the linear mappings from their independent element space to the constrained solution sets, study their properities and use these properties to propose two matrix iterative algorithms that can find the minimum and quasi-minimum norm solution based on the classical LSQR algorithm for solving the unconstrained LS problem. Numerical results are provided that show the efficiency of the proposed methods.

AMS Mathematics Subject Classification : 15A09, 65F20.
Key words and phrases : Matrix equation, submatrix constraint, least squares problem, LSQR, iterative algorithm.

1. Introduction

Denoted by $\mathcal{R}^{m\times n}$ and $\mathcal{SR}^{n\times n}$ the set of $m\times n$ real matrices and the set of $n\times n$ real symmetric matrices, respectively. For any $A\in\mathcal{R}^{m\times n}$, $A^T,\mathcal{R}(A),A^{\dagger}$, $\|A\|_2$ and $\|A\|_F$ present the transpose, range, Moore-Penrose generalized inverse, Euclidian norm and Frobenius norm of A, respectively. $A(i:j,k:l)$ represents the submatrix of A containing the intersection of rows i to j and columns k to l. As a special case, $A(:,j)$ is the jth column of A and $A(i,:)$ the ith row of A. For a vector a, $a(i:j)$ is the vector containing the ith to jth elements. I_n denotes the unit matrix of order $n\times n$ and $e_i^{(k)}$ is the ith column of I_k, while the ith column of I_n is simply denoted by e_i. For any $X\in\mathcal{SR}^{n\times n}$, we define a following symmetry norm:

$$\|X\|_S=\sqrt{\sum_{i\geq j}x_{ij}^2}.$$

Received January 4, 2008. Revised March 14, 2008. Accepted March 22, 2008. *Corresponding author. †This work is supported by the National Natural Science Foundation of China(Grant No:10771073).

The well-known linear matrix equation $AXB = C$ has been widely studied. Inevitably, Moore-Penrose generalized inverses and some complicated matrix decompositions such as canonical correlation decomposition (CCD) and general singular value decomposition (GSVD) are involved. All these methods are direct methods. With the increasing dimension of the system, direct methods face many difficulties and become impractical, in which case, iterative methods play an important role.

In [4,5], matrix iteration methods were given for solving $AXB = C$ with the symmetry constraint $X^T = X$. They are matrix-form CGLS method and LSQR method, which can be obtained by applying the classical CGLS method[7] and LSQR method[3] respectively to matrix LS problem $\min_X \|BXA^T - C\|_F$. The matrix-form CGLS method can be easily derived from the classical CGLS method applied on the vector-representation of the matrix LS using Kronecker product. However, as well known, the condition number is squared when normal equation is involved. This may lead to numerical instability. It is not easy to derive the matrix-form LSQR method, which has favorable numerical properties.

Matrix equation peoblems with a submatrix constraint come from the expansion problem of subsystem[1] and are very important. Z. Peng, X. Hu and L. Zhang[5] studied symmetric solutions and bisymmetric solutions of $AX = B$ with a principal submatrix constraint, but did not discuss the least-squares solutions.

In this paper, we will discuss the least-squares symmetric solution of $AXB = C$ with arbitrary submatrix constraint. For simplicity of expression, we only consider the following leading principal submatrix constraint problem.

Given $A \in \mathcal{R}^{m\times n}, B \in \mathcal{R}^{n\times l}, C \in \mathcal{R}^{m\times l}, X_0 \in \mathcal{SR}^{k\times k}$. Find the least-squares solution for

$$||AXB - C||_F = \min, X \in \mathcal{S} \tag{1}$$

with $\mathcal{S} = \{X|X \in \mathcal{SR}^{n\times n}, X(1:k, 1:k) = X_0\}$.

In fact, arbitrary principal submatrix constraint problem can transform into the above problem. Furthermore, arbitrary submatrix constraint problem can be similarly discussed.

Let $\mathcal{S}_{\mathcal{O}} = \{X|X \in \mathcal{SR}^{n\times n}, X(1:k, 1:k) = 0\}$. Then

$$\begin{aligned}\min_{X\in\mathcal{S}} ||AXB - C||_F &= \min_{Y\in\mathcal{S}_{\mathcal{O}}} \left\| A\left(\begin{pmatrix} X_0 & 0 \\ 0 & 0 \end{pmatrix} + Y \right) B - C \right\|_F \\ &= \min_{Y\in\mathcal{S}_{\mathcal{O}}} \left\| AYB - \left(C - A\begin{pmatrix} X_0 & 0 \\ 0 & 0 \end{pmatrix} B \right) \right\|_F .\end{aligned}$$

Let

$$\bar{X} = \begin{pmatrix} X_0 & 0 \\ 0 & 0 \end{pmatrix}, \bar{C} = C - A\bar{X}B. \tag{2}$$

Then the following result is obvious.

Lemma 1. *Suppose that $\bar{X}, \bar{C}$ be denoted by (2). Then the general solution (1) is $X = \bar{X} + Y$, where Y is the general solution*

$$\min_{Y \in \mathcal{S}_O} ||AYB - \bar{C}||_F. \tag{3}$$

Therefore, in this paper, we only consider (1) with $\mathcal{S} = \mathcal{S}_O$. The remaining part of paper is organized as follows. In Section 2, we characterize the symmetric matrices with a leading principal submatrix constraint. In Section 3, we shortly review the algorithm LSQR for $\min_x \|Mx - f\|$, which is numerically very reliable even if M is ill-conditioned. Our matrix iterative algorithms for (1) is proposed in Section 4, based on the classical LSQR method. Numerical examples are provided in Section 5 to show the efficiency of the algorithms.

2. Symmetric matrices with a leading principal submatrix constraint

A symmetric matrix with a leading principal submatrix constraint is uniquely and linearly determined by its partial elements, which are called independent elements. For any $X = (x_1, x_2, \cdots, x_n) \in \mathcal{R}^{n \times n}$, define

$$vec(X) = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathcal{R}^{n^2}, vec_i(X) = \begin{pmatrix} x(k+1:n,1) \\ \vdots \\ x(k+1:n,k+1) \\ x(k+2:n,k+2) \\ \vdots \\ x(n,n) \end{pmatrix} \in \mathcal{R}^N, \tag{4}$$

where $N = (n-k)(n+k+1)/2$.

Obviously, there is a one to one linear mapping from the independent element space

$$vec_i(\mathcal{S}_O) \equiv \{vec_i(X) | X \in \mathcal{S}_O\}$$

to the long-vector space

$$vec(\mathcal{S}_O) \equiv \{vec(X) | X \in \mathcal{S}_O\}.$$

Let $\mathcal{F}(n)$ be this linear mapping:

$$X \in \mathcal{S}_O, \quad vec(X) = \mathcal{F}(n) vec_i(X).$$

We call $\mathcal{F}(n)$ a symmetric constraint matrix of degree n with a leading principal submatrix constraint. If n can be ignored without misunderstanding, $\mathcal{F}(n)$ will be simply denoted by $\mathcal{F}$.

Next we give the representations of $\mathcal{F}(n)$.

Theorem 1. *Suppose that $\mathcal{F} \in \mathcal{R}^{n^2 \times N}$ is a symmetric constraint matrix of degree n with a leading principal submatrix constraint. Then*

(1) $\mathcal{F} = (F_{ij})$ *is a block lower triangular matrix with*

$$i \le k\ , F_{ii} = \begin{pmatrix} 0_{k\times(n-k)} \\ I_{n-k} \end{pmatrix}, , F_{ij} = 0, (i \ne j);$$

$$i > k\ , F_{ii} = \begin{pmatrix} 0_{(i-1)\times(n-i+1)} \\ I_{n-i+1} \end{pmatrix}, \ F_{ij} = 0, (i < j);$$

$$i > k, j \le k+1, i \ne j\ , F_{ij} = e_j e_{i-k}^{(n-k)^T};$$

$$i \ge k+3, j \ge k+2, i \ne j\ , F_{ij} = e_j e_{i-j+1}^{(n-j+1)^T}.$$

(2) $\mathcal{F}^T\mathcal{F} = diag(D_1, \cdots, D_n)$, *where*

$D_i = 2I_{n-k}, (i \le k)$; $D_i = diag(1, 2, \cdots, 2) \in R^{(n-i+1)\times(n-i+1)}, (i > k)$.

Proof. (1). For any $X \in \mathcal{S}_{\mathcal{O}}$, when $i \le k$,

$$x_i = \begin{pmatrix} 0_k \\ x(k+1:n, i) \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & I_{n-k} & 0 & \cdots & 0 \end{pmatrix} vec_i(X).$$

when $i > k$,

$$x_i = \begin{pmatrix} x(i,1) \\ x(i,2) \\ \vdots \\ x(i,k) \\ x(i,k+1) \\ x(i,k+2) \\ x(i,k+3) \\ \vdots \\ x(i,i-1) \\ x(i:n,i) \end{pmatrix} = \begin{pmatrix} e_{i-k}^{(n-k)^T} x(k+1:n,1) \\ e_{i-k}^{(n-k)^T} x(k+1:n,2) \\ \vdots \\ e_{i-k}^{(n-k)^T} x(k+1:n,k) \\ e_{i-k}^{(n-k)^T} x(k+1:n,k+1) \\ e_{i-k-1}^{(n-k-1)^T} x(k+2:n,k+2) \\ e_{i-k-2}^{(n-k-2)^T} x(k+3:n,k+3) \\ \vdots \\ e_2^{(n-i+2)^T} x(i-1:n,i-1) \\ I_{n-i+1} x(i:n,i) \end{pmatrix}$$

$$= diag(e_{i-k}^{(n-k)^T}, \cdots, e_{i-k}^{(n-k)^T}, e_{i-k-1}^{(n-k-1)^T}, \cdots, e_2^{(n-i+2)^T}, I_{n-i+1}, 0) vec_i(X)$$

$$= (F_{i1}, F_{i2}, \cdots, F_{ii}, 0) vec_i(X).$$

(2). Notice that $\mathcal{F}$ is column orthogonal. By simple computation, we know that (2) is correct. □

From the properties and structure of $\mathcal{F}$, by simple computation, we can obtain that, for any $X \in \mathcal{SR}^{n\times n}$,

$$\mathcal{F}^\dagger vec(X) = vec_i(X).$$

Furthermore, for any $X \in \mathcal{R}^{n\times n}$, $\mathcal{F}^\dagger vec(X) = \mathcal{F}^\dagger vec(X^T)$. Therefore, for any $X \in \mathcal{R}^{n\times n}$,

$$2\mathcal{F}^\dagger vec(X) = \mathcal{F}^\dagger vec(X + X^T) = vec_i(X + X^T).$$

This leads to the following result.

Theorem 2. *Suppose that $\mathcal{F} \in \mathcal{R}^{n^2\times N}$ is a symmetric constraint matrix of degree n with a leading principal submatrix constraint and $Y \in \mathcal{R}^{n\times n}$. Then*

$$\mathcal{F}^{\dagger}vec(Y) = vec_i\left(\frac{Y+Y^T}{2}\right).$$

For any $X \in \mathcal{R}^{n\times n}$, we define

$$v\bar{e}c_i(X) = Pvec_i(X),$$

where $P = diag(p_1, \cdots, p_N)$,

$$p_i = \begin{cases} \sqrt{2}/2, & i = (n-k)k+1, (n-k)k+(n-k)+1, \\ & \quad (n-k)k+(n-k)+(n-k-1)+1, \cdots, N \\ 1, & \text{otherwise} \end{cases};$$

that is , giving every independent element x_{ii} a weight $\sqrt{2}/2$. Define $\bar{\mathcal{F}}(n)$ as a linear mapping from $v\bar{e}c_i(\mathcal{S}_{\mathcal{O}})$ to $vec(\mathcal{S}_{\mathcal{O}})$:

$$X \in \mathcal{S}_{\mathcal{O}}, \quad vec(X) = \bar{\mathcal{F}}(n)v\bar{e}c_i(X).$$

we call $\bar{\mathcal{F}}(n)$ a symmetric constraint matrix of degree n with a leading principal submatrix and minimum norm constraint. If n can be ignored without misunderstanding, $\bar{\mathcal{F}}(n)$ will be simply denoted by $\bar{\mathcal{F}}$.

From Theorem 1 and 2, we can easily obtain the following results.

Theorem 3. *Suppose that $\bar{\mathcal{F}}$ is a symmetric constraint matrix of degree n with a leading principal submatrix and minimum norm constraint and $\mathcal{F}$ is defined by Theorem 1. Then*

(1) $\bar{\mathcal{F}} = \mathcal{F}P^{-1}$; (2) $\bar{\mathcal{F}}^T\bar{\mathcal{F}} = 2I_N$.

Theorem 4. *Suppose that $\bar{\mathcal{F}}$ is a symmetric constraint matrix of degree n with a leading principal submatrix and minimum norm constraint and $Y \in \mathcal{R}^{n\times n}$. Then*

$$\bar{\mathcal{F}}^{\dagger}vec(Y) = v\bar{e}c_i\left(\frac{Y+Y^T}{2}\right).$$

3. Algorithm LSQR

In the section, we briefly review the algorithm LSQR proposed by Paige and Sauders[3] for solving the following lease squares problem:

$$\min_{x\in\mathcal{R}^n} \|Mx - f\|_2 \tag{5}$$

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

$$M^TMx = M^Tf. \tag{6}$$

The LSQR algorithm is based on the bidiagonalization procedure of Golub and Kahan[2],

$$MV_k = U_{k+1}\bar{N}_k, \quad M^TU_k = V_kN_k^T, \tag{7}$$

where $V_k = (v_1, \cdots, v_k)$ and $U_{k+1} = (u_1, \cdots, u_{k+1})$ are orthogonal, and

$$\bar{N}_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} = \begin{pmatrix} N_k \\ (0, \cdots, 0, \beta_{k+1}) \end{pmatrix} \in R^{(k+1)\times n}$$

is a lower-bidiagonal matrix. When u_1 is preset, v_1 and the recursions that generate $v_{i+1},\ u_{i+1},\ i = 1, 2, \cdots$ are obtained with

$$\begin{aligned} &\bar{v}_1 = M^T u_1, \quad \alpha_1 = \|\bar{v}_1\|_2, \quad v_1 = \bar{v}_1/\alpha_1, \\ &\bar{u}_{i+1} = M v_i - \alpha_i u_i, \quad \beta_{i+1} = \|\bar{u}_{i+1}\|_2, \quad u_{i+1} = \bar{u}_{i+1}/\beta_{i+1}, \\ &\bar{v}_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i, \quad \alpha_{i+1} = \|\bar{v}_{i+1}\|_2, \quad v_{i+1} = \bar{v}_{i+1}/\alpha_{i+1}. \end{aligned}$$

It is easy to verify that

$$u_i \in \kappa_i(MM^T, u_1), \quad v_i \in \kappa_i(M^T M, M^T u_1).$$

If we take $u_1 = \bar{r}_0/\|\bar{r}_0\|_2$ with $\bar{r}_0 = f - Mx_0$, then

$$\kappa_k = \kappa_k(M^T M, M^T \bar{r}_0) = span\{v_1, \cdots, v_k\}.$$

We can write $x_k = x_0 + V_k y_k, y_k \in \mathcal{R}^k$. So setting

$$y_k = \arg\min_{y \in R^k} \|f - M(x_0 + V_k y)\|_2$$

gives the same sequence $\{x_k\}$ as CGLS. Let $\beta_1 = \|\bar{r}_0\|_2$, then it follows from (7) that

$$\|f - M(x_0 + V_k y)\|_2 = \|U_{k+1}(\beta_1 e_1 - \bar{N}_k y)\|_2 = \|\beta_1 e_1 - \bar{N}_k y\|_2. \tag{8}$$

Therefore,

$$y_k = \arg\min_{y \in R^k} \|\beta_1 e_1 - \bar{N}_k y\|_2.$$

The LSQR method solves the optimization problem above and constructs $V_k y_k$ iteratively, using the QR decomposition of the lower bidiagonal matrix $\bar{N}_k$ and simultaneously transforming $\beta_0 e_1$,

$$Q_k \bar{N}_k = \begin{pmatrix} \rho_1 & \theta_2 & & \\ & \ddots & \ddots & \\ & & \rho_{k-1} & \theta_k \\ & & & \rho_k \\ & & & 0 \end{pmatrix} = \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \quad Q_k(\beta_1 e_1) = \begin{pmatrix} \zeta_1 \\ \vdots \\ \zeta_k \\ \bar{\zeta}_{k+1} \end{pmatrix}$$

$$= \begin{pmatrix} z_k \\ \bar{\zeta}_{k+1} \end{pmatrix}.$$

This transformation can be easily done by Givens rotations as follows. Set $\bar{\rho}_1 = \alpha_1, \bar{\zeta}_1 = \beta_1$, and for $i = 1, \cdots, k$, construct Givens rotation $G_i = \begin{pmatrix} c_i & s_i \\ s_i & -c_i \end{pmatrix}$ such that

$$\begin{pmatrix} c_i & s_i \\ s_i & -c_i \end{pmatrix} \begin{pmatrix} \bar{\rho}_i & 0 & \bar{\zeta}_i \\ \beta_{i+1} & \alpha_{i+1} & 0 \end{pmatrix} = \begin{pmatrix} \rho_i & \theta_{i+1} & \zeta_i \\ 0 & \bar{\rho}_{i+1} & \bar{\zeta}_{i+1} \end{pmatrix}.$$

Clearly, the optimal solution is given by $y_k = R_k^{-1} z_k$ and

$$f - Mx_k = U_{k+1} Q_k^T (\bar{\zeta}_{k+1} e_{k+1}^{(k+1)}).$$

Together with (7) and (8), we have

$$\|f - Mx_k\|_2 = |\bar{\zeta}_{k+1}|, \quad \|M^T(f - Mx_k)\|_2 = |\alpha_{k+1}\bar{\zeta}_{k+1} c_k|. \tag{9}$$

We can choose $\|M^T(f - Mx_k)\|_2 = |\alpha_{k+1}\bar{\zeta}_{k+1} c_k| < \tau$ as stop criteria, where $\tau > 0$ is a small tolerance.

We rewrite the recursion form

$$x_k = x_0 + V_k y_k = x_0 + V_k R_k^{-1} z_k \equiv G_k z_k,$$

where $G_k = V_k R_k^{-1} = (G_{k-1}, g_k)$. Then $g_1 = v_1/\rho_1$ and $g_k = (v_k - \theta_k g_{k-1})/\rho_k$ for $k > 1$. Thus

$$x_k = x_0 + (G_{k-1}, g_k) \begin{pmatrix} z_{k-1} \\ \zeta_k \end{pmatrix} = x_{k-1} + \zeta_k g_k.$$

Define $h_k = \rho_k g_k$. Then we have from the above formulas, $h_1 = \rho_1 g_1 = v_1$,

$$x_k = x_{k-1} + \frac{\zeta_k}{\rho_k} h_k, \quad h_{k+1} = \rho_{k+1}(v_{k+1} - \theta_{k+1} g_k)/\rho_{k+1} = v_{k+1} - \frac{\theta_{k+1}}{\rho_k} h_k.$$

Theoretically, LSQR converges within at most n iterations if exact arithmetic could be performed, where n is the length of x. In practice the iteration number of LSQR may be larger than n because of the computational errors. It was shown in [3] that LSQR is numerically more reliable even if M is ill-conditioned.

We summarize the LSQR algorithm as follows.

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 $x^* \in \mathcal{R}(M^T)$, then x^* is the unique minimal norm solution of $Mx = f$. So, if (6) has a solution $x^* \in \mathcal{R}(M^TM) = \mathcal{R}(M^T)$, then x^* is the minimum norm solution of (5). It is obvious that x_k generated by Algorithm LSQR belongs to $\mathcal{R}(M^T)$ and this leads the following result.

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

4. The quasi-minimum norm solution and minimum norm solution for (1)

Since $vec(AXB) = (B^T \otimes A)vec(X)$, where $\otimes$ denote the Kronecker product, then we have $vec(AXB) = (B^T \otimes A)\mathcal{F}x$ and the problem (1) is equivalent to

$$\|Mx - f\|_2 = \min, \tag{10}$$

where

$$M = (B^T \otimes A)\mathcal{F} \in \mathcal{R}^{lm\times N}, \quad f = vec(C) \in \mathcal{R}^{lm}.$$

The vector iterations of LSQR will be rewritten into matrix form so that the Kronecker product and $\mathcal{F}$ can be released. To this end, it is required to transform the matrix-vector products of Mv and M^Tu back to a matrix-matrix form for variant vectors v in the independent element space and $u = vec(U) \in \mathcal{R}^{ml}$. Further, we must guarantee that matrix form of M^Tu belongs to $\mathcal{S}_\mathcal{O}$.

For any $v \in \mathcal{R}^N$, let $V \in \mathcal{S}_\mathcal{O}$ satisfy $vec_i(V) = v$. Then we have

$$mat(Mv) = mat((B^T \otimes A)\mathcal{F}v) = mat((B^T \otimes A)vec(V)) = AVB.$$

For any $u \in \mathcal{R}^{lm}$, let $U \in \mathcal{R}^{l\times m}$ satisfy $u = vec(U)$ and define

$$G = A^TUB^T, \quad H = \frac{G + G^T}{2},$$

$$Z = \begin{pmatrix} 0 & H(1:k, k+1:n) \\ H(k+1:n, 1:k) & H(k+1:n, k+1:n) \end{pmatrix}.$$

Then we have

$$\begin{aligned} mat(M^Tu) &= mat(\mathcal{F}^T(B \otimes A^T)vec(U)) \\ &= mat(\mathcal{F}^T vec(A^TUB^T)) = mat(\mathcal{F}^T\mathcal{F}\mathcal{F}^\dagger vec(G)) \\ &= mat(\mathcal{F}^T\mathcal{F}vec_i(H)) = mat(\mathcal{F}^T\mathcal{F}vec_i(Z)) \\ &= 2Z - diag(Z). \end{aligned}$$

Now we can give the following algorithm.

Algorithm LSQR-M-BS

(1) Initialization.

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

$G_1 = A^T U_1 B^T, \quad T_1 = (G_1 + G_1^T)/2,$

$Z_1 = \begin{pmatrix} 0 & T(1:k, k+1:n) \\ T(k+1:n, 1:k) & T(k+1:n, k+1:n) \end{pmatrix},$

$\bar{V}_1 = 2Z_1 - diag(Z_1), \quad \alpha_1 = \|\bar{V}_1\|_S, \quad V_1 = \bar{V}_1/\alpha_1,$

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

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

$\bar{U}_{i+1} = AV_iB - \alpha_i U_i,$

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

$G_{i+1} = A^T U_{i+1} B^T, \quad T_{i+1} = (G_{i+1} + G_{i+1}^T)/2,$

$Z_{i+1} = \begin{pmatrix} 0 & T_{i+1}(1:k, k+1:n) \\ T_{i+1}(k+1:n, 1:k) & T_{i+1}(k+1:n, k+1:n) \end{pmatrix},$

$\bar{V}_{i+1} = 2Z_{i+1} - diag(Z_{i+1}) - \beta_{i+1}V_i,$

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

$\rho_i = \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2}, \quad c_i = \bar{\rho}_i/\rho_i, \quad s_i = \beta_{i+1}/\rho_i, \quad \theta_{i+1} = s_i\alpha_{i+1},$

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

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

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

check convergence.

Remark 1. The solution obtained by LSQR-M-BS has minimum symmetry norm, and therefor is called *quasi-minimum norm solution.*

Next, we discussed the method for minimum Frobenius norm solution. The problem (1) is equivalent to

$$\|Mx - f\|_2 = \min, \tag{11}$$

where

$$M = (B^T \otimes A)\bar{\mathcal{F}} \in \mathcal{R}^{lm\times N}, \quad f = vec(E) \in \mathcal{R}^{lm}.$$

Notice that x comes from $v\bar{e}c_i(X)$.

For any $v \in \mathcal{R}^N$, let $V \in \mathcal{S}_\mathcal{O}$ satisfy $vec_i(V) = v$ and define

$$\tilde{V} = V + (\sqrt{2} - 1)diag(V).$$

Then we have

$$\begin{aligned} mat(Mv) &= mat((B^T \otimes A)\bar{\mathcal{F}}vec_i(V)) = mat((B^T \otimes A)\mathcal{F}P^{-1}vec_i(V)) \\ &= mat((B^T \otimes A)\mathcal{F}vec_i(\tilde{V})) = mat((B^T \otimes A)vec(\tilde{V})) \\ &= A\tilde{V}B. \end{aligned}$$

For any $u \in \mathcal{R}^{lm}$, let $U \in \mathcal{R}^{l\times m}$ satisfy $u = vec(U)$ and define

$$G = A^TUB^T, H = \frac{G + G^T}{2}, Z = \begin{pmatrix} 0 & H(1:k, k+1:n) \\ H(k+1:n, 1:k) & H(k+1:n, k+1:n) \end{pmatrix}.$$

Then we have

$$\begin{aligned} mat(M^T u) &= mat(\bar{\mathcal{F}}^T (B \otimes A^T) vec(U)) \\ &= mat(\bar{\mathcal{F}}^T vec(A^T U B^T)) = mat(\bar{\mathcal{F}}^T \bar{\mathcal{F}} \bar{\mathcal{F}}^\dagger vec(G)) \\ &= mat(2 v\bar{e}c_i(H)) = mat(2 v\bar{e}c_i(Z)) = mat(2 P vec_i(Z)) \\ &= 2Z - (2 - \sqrt{2}) diag(Z). \end{aligned}$$

Now we can give the following algorithm.

Algorithm LSQRmin-M-BS

(1) Initialization.

$Y_0 = 0 (\in \mathcal{R}^{n\times n}), \quad \beta_1 = \|E\|_F, \quad U_1 = E/\beta_1,$

$G_1 = A^T U_1 B^T, \quad T_1 = (G_1 + G_1^T)/2,$

$$Z_1 = \begin{pmatrix} 0 & T(1:k, k+1:n) \\ T(k+1:n, 1:k) & T(k+1:n, k+1:n) \end{pmatrix},$$

$\bar{V}_1 = Z_1 + Z_1^T - (2-\sqrt{2}) diag(Z), \quad \alpha_1 = \|\bar{V}_1\|_S, \quad V_1 = \bar{V}_1/\alpha_1,$

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

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

$\tilde{V}_i = V_i + (\sqrt{2}-1) diag(V_i), \quad \bar{U}_{i+1} = A\tilde{V}_i B - \alpha_i U_i,$

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

$G_{i+1} = A^T U_{i+1} B^T, \quad T_{i+1} = (G_{i+1} + G_{i+1}^T)/2,$

$$Z_{i+1} = \begin{pmatrix} 0 & T_{i+1}(1:k, k+1:n) \\ T_{i+1}(k+1:n, 1:k) & T_{i+1}(k+1:n, k+1:n) \end{pmatrix},$$

$\bar{V}_{i+1} = Z_{i+1} + Z_{i+1} - (2-\sqrt{2}) diag(Z_{i+1}) - \beta_{i+1} V_i,$

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

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

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

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

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

$X_i = Y_i + (\sqrt{2}-1) diag(Y_i),$

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

check convergence.

Remark 2. From the transforming process from (1) to (11), Y_i obtained by Algorithm LSQRmin-M-BS and approximate solution X_i of (1) satisfy $vec_i(Y_i) = v\bar{e}c_i(X_i)$. So we have $X_i = Y_i + (\sqrt{2}-1) diag(Y_i)$.

Remark 3. Algorithm LSQRmin-M-BS can compute Y with minimal $\|vec_i(Y)\|_2$, and hence X with minimal $\|v\bar{e}c_i(X)\|_2$. Because $\|X\|_F^2 = 2\|v\bar{e}c_i(X)\|_2^2$, Algorithm LSQRmin-M-BS can compute the minimum Frobenius norm solution of (1).

5. Numerical examples

In this section, we present a numerical example to illustrate the efficiency of Algorithm LSQRmin-M-BS.

Example 1. *Let*

$$A = \begin{pmatrix} 3 & -20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 \\ 1 & 13 & 0 & 0 & -21 \\ 0 & 2 & 0 & 0 & 17 \end{pmatrix}, B = \begin{pmatrix} -31 & 70 & 1 \\ -51 & 11 & 3 \\ 0 & 0 & 0 \\ 4 & 0 & -17 \\ 9 & 23 & -19 \end{pmatrix},$$

$$C = \begin{pmatrix} 0 & 1 & -4 \\ 3 & -4 & 0 \\ 5 & 1 & -1 \\ -7 & 0 & 0 \end{pmatrix}, X_0 = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 0 & 3 \\ -1 & 3 & -2 \end{pmatrix},$$

where $N = 9, rank(M) = 7$.

With 15 iterations, Algorithm LSQRmin-M-BS obtains the minimum Frobenius norm solution

$$X_{15} = \begin{pmatrix} 1.00000000 & 2.00000000 & -1.00000000 & -6.453694647911 & 5.942629102890 \\ 2.00000000 & 0 & 3.00000000 & 5.583496558026 & -4.373544972661 \\ -1.00000000 & 3.00000000 & -2.00000000 & 0.00000000000001 & 0.000000000000 \\ -6.453694647911 & 5.583496558026 & 0.00000000000001 & -18.131131672281 & 16.837529191766 \\ 5.942629102890 & -4.373544972661 & 0.000000000000 & 16.837529191766 & -15.189156071512 \end{pmatrix}$$

with

$$||AX_{15}B - C|| = 1.627240099172723e + 003.$$

These results are the same as those obtained by direct method.

In Figure 1, we plot convergence curve of normal equation error

$$\eta_k = \|M^T(M\bar{vec}_i(X_k) - f)\|_2.$$

Above example, as well as more other examples, show that our algorithms are efficient.

References

1. P. Deift and T. Nanda, *On the determination of a tridiagonal matrix from its spectrum and a submatrix*, Linear Algebra Appl., **60** (1984), 43-55.
2. G.H. Golub and W. Kahan, *Calculating the singular values and pseudoinverse of a matrix*, SIAM J. Numer. Anal., **2** (1965), 205-224.
3. C.C. Paige and A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, **8** (1982), 43-71.

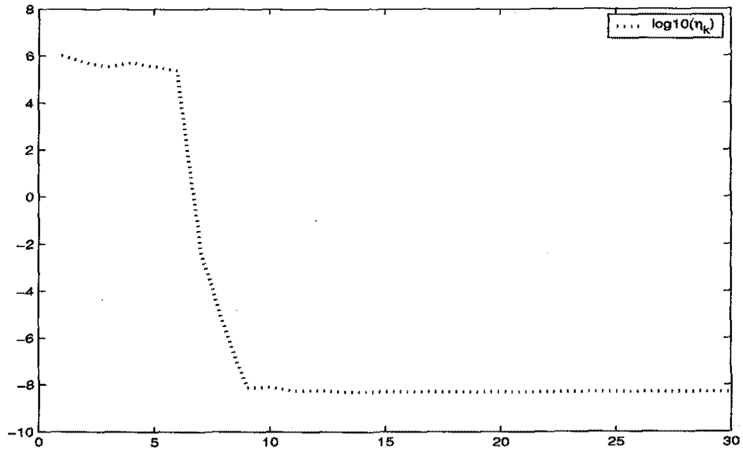

FIGURE 1. Error and iteration number

4. Z. Peng, *An iterative method for the least squares symmetric solution of the linear matrix equation* $AXB = C$, Appl. Math. Comput., **170** (2005), 711-723.
5. Z. Peng, X. Hu and L. Zhang, *The inverse problem of bisymmetric matrices with a submatrix constraint*, Numer. Linear Algebra Appl., **11** (2004), 59-73.
6. Y. Qiu Z. Zhang and J. Lu, *Matrix iterative solutions to the least squares problem of* $BXA^T = F$ *with some linear constraints*, Appl. Math. Comput., **185** (2007), 284-300.
7. E. Stiefel, *Ausgleichung ohne Aufstellung der Gausschen Normalgleichungen*, Wiss. Z. Tech. Hochsch. Dresden **2** (1952-1953), 441-442.

Minghui Wang received his B.E. degree from Qufu normal University in 2000 and M.E. degree in Numerical Algebra in 2003 from East China Normal University. Currently, He is pursuing his Ph.D. degree in East China Normal University. His research interests include matrix equations and generalized inverse, etc.

School of Mathematical Sciences, Qufu Normal University, Qufu Shandong 273165, China
e-mail: mhwang@yeah.net

Yan Feng received her B.E. degree from Qufu normal University in 1999, M.E. degree in Control Systems Theory in 2004 from Qufu normal University and Ph.D. degrees in computer science in 2007 from Shanghai University. Her research interests include image process and virtual reality, etc.

College of Information Science & Technology, Qingdao University of Science & Technology, Qingdao Shandong 266061, China
e-mail: fywmh@163.com