 CMES, vol.77, no.3, pp.173-182, 2011

An Iterative Method for the Least-Squares Minimum-Norm Symmetric Solution

Minghui Wang[1], Musheng Wei[2] and Shanrui Hu[1]

Abstract: The mapping from the symmetric solution set to its independent parameter space is studied and an iterative method is proposed for the least-squares minimum-norm symmetric solution of $AXB = E$. Numerical results are reported that show the efficiency of the proposed methods.

Keywords: Matrix equation, symmetric matrix, Least squares problem, minimum norm; Iterative algorithm.

1 Introduction

Denoted by $\mathscr{R}^{m\times n}$ and $\mathscr{S}\mathscr{R}^{n\times n}$ the set of $m\times n$ real matrices and the set of $n\times n$ real symmetric matrices, respectively. For any $A\in\mathscr{R}^{m\times n}$, $\mathscr{R}(A), A^T, A^\dagger$, $\|A\|_2$ and $\|A\|_F$ present the range, transpose, Moore-Penrose generalized inverse, Euclid norm and Frobenius norm, respectively. $A\otimes B$ represents the Kronecker product of matrices A and B. The sub-vector consisting of from αth component to βth component of x_i is denoted by $x_{\alpha:\beta,i}$. For any $X\in\mathscr{S}\mathscr{R}^{n\times n}$, we define a following symmetry norm:

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

Let m,n,l be three positive integers, and let $E\in\mathscr{R}^{m\times l}, A\in\mathscr{R}^{m\times n}$ and $B\in\mathscr{R}^{n\times l}$. We consider the least squares problem

$$\min_{X\in\mathscr{S}\mathscr{R}^{n\times n}} \|AXB-E\|_F, \tag{1}$$

and its corresponding linear matrix equation is

$$AXB = E. \tag{2}$$

[1] Department of Mathematics, Qingdao University of Science and Technology, Qingdao 266061, PR China.
[2] Department of Mathematics, Shanghai Normal University, Shanghai 200234, PR China.

Eq.(1) and Eq.(2) have been widely studied due to its some applications in electricity, control theory, processing of digital signals(See Xie, Sheng and Hu (2003)) and the design and analysis of the vibrating structures(See Yuan and Dai (2007)). For solving them, 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 called direct methods. With the increasing dimension of the system, direct methods face many difficulties and become impractical, and here iterative methods play an important role.

In Peng (2005); Peng, Hu and Zhang (2005); Hou, Peng and Zhang (2006); Qiu, Zhang and Lu (2007), 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-Stiefel (1952) and LSQR methodPaige and Saunders (1982) respectively to matrix LS problem $\min_X \|AXB-C\|_F$. 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 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.

Sometimes, it is important to find the minimum-norm symmetric solution of Eq.(1) or Eq.(2). But, the algorithm of Qiu, Zhang and Lu (2007) can't do so. In this paper, on basis of the idea of Qiu, Zhang and Lu (2007), an iterative method for the least-squares minimum-norm symmetric solution will be proposed and relevant results are more perfect.

2 Minimum-norm symmetry constraint matrix

A symmetric matrix is uniquely determined by part of its elements, namely some independent elements. For a matrix $X = (x_1, x_2, \cdots, x_n) \in \mathscr{R}^{n\times n}$, we define

$$\mathrm{vec}(X) = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathscr{R}^{n^2}, \mathrm{vec}_i(X) = \begin{pmatrix} x_{11}/\sqrt{2} \\ x_{2:n,1} \\ x_{22}/\sqrt{2} \\ x_{3:n,2} \\ \vdots \\ x_{nn}/\sqrt{2} \end{pmatrix} \in \mathscr{R}^{N}, \tag{3}$$

where $N \equiv n(n+1)/2$. Obviously, there is an one to one linear mapping from the long-vector space

$$\mathrm{vec}(\mathscr{S}\mathscr{R}^{n\times n}) = \{\mathrm{vec}(X)|X \in \mathscr{S}\mathscr{R}^{n\times n}\}$$

to the independent parameter space

$$\mathrm{vec}_i(\mathscr{S}\mathscr{R}^{n\times n}) = \{\mathrm{vec}_i(X)|X \in \mathscr{S}\mathscr{R}^{n\times n}\}.$$

Let us denote by $\mathscr{F}(n)$ the matrix that defines linear mapping form $\mathrm{vec}_i(\mathscr{S}\mathscr{R}^{n\times n})$ to $\mathrm{vec}(\mathscr{S}\mathscr{R}^{n\times n})$,

$$X \in \mathscr{S}\mathscr{R}^{n\times n}, \quad \mathrm{vec}(X) = \mathscr{F}(n)\mathrm{vec}_i(X).$$

We call $\mathscr{F}(n) \in \mathscr{R}^{n^2\times N}$ a minimum-norm symmetry constraint matrix of degree n, which will be simply denoted by $\mathscr{F}$ if n can be ignored without misunderstanding. Next we give the representations of $\mathscr{F}(n)$.

Theorem 2.1. *Suppose $\mathscr{F} \in \mathscr{R}^{n^2\times N}$a minimum-norm symmetric constraint matrix of degree n. Then*
(1). $\mathscr{F} = (F_{ij})$ is a block lower triangular matrix with

$$i > j, F_{ij} = e_j e_{i-j+1}^{(n-j+1)^T}; i < j, F_{ij} = 0; F_{ii} = \begin{pmatrix} 0 & 0 \\ \sqrt{2} & 0 \\ 0 & I_{n-i} \end{pmatrix}.$$

(2). $\mathscr{F}^T\mathscr{F} = 2I_N$.
(3). $\mathscr{F}^\dagger = \frac{1}{2}\mathscr{F}^T$.

From the above theorem, we see that the properties and structure of $\mathscr{F}$ are simpler than those of Qiu, Zhang and Lu (2007). On the basis of this theorem, in section 4, we will get a better result(Theorem 4.1).
By simple computation, we can obtain that, for any $X \in \mathscr{S}\mathscr{R}^{n\times n}$,

$$\mathscr{F}^\dagger\mathrm{vec}(X) = \mathrm{vec}_i(X).$$

Furthermore, for any $X \in \mathscr{R}^{n\times n}$,

$$\mathscr{F}^\dagger\mathrm{vec}(X) = \mathscr{F}^\dagger\mathrm{vec}(X^T).$$

So, for any $X \in \mathscr{R}^{n\times n}$,

$$2\mathscr{F}^\dagger\mathrm{vec}(X) = \mathscr{F}^\dagger\mathrm{vec}(X+X^T) = \mathrm{vec}_i(X+X^T).$$

This leads to the following result.

Theorem 2.2. *Suppose $\mathscr{F} \in \mathscr{R}^{n^2 \times N}$ a minimum-norm symmetriy constraint matrix of degree n and $Y \in \mathscr{R}^{n\times n}$. Then*

$$\mathscr{F}^{\dagger}\mathrm{vec}(Y) = \mathrm{vec}_i\left(\frac{Y+Y^T}{2}\right), \mathscr{F}\mathscr{F}^{\dagger}\mathrm{vec}(Y) = \mathrm{vec}\left(\frac{Y+Y^T}{2}\right).$$

3 Algorithm LSQR

In the section, we briefly review the algorithm LSQR prosed by Paige and SaudersPaige and Saunders (1982) for solving the following lease squares problem:

$$\min_{x\in\mathscr{R}^n} \|Mx - f\|_2 \tag{4}$$

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

$$M^T Mx = M^T f. \tag{5}$$

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 Paige and Saunders (1982) 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)$\beta_{i+1}u_{i+1} = Mv_i - \alpha_i u_i$
  (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 $x^* \in \mathscr{R}(M^T)$, then x^* is the unique minimal norm solution of $Mx = f$. So, if Eq.(5) has a solution $x^* \in \mathscr{R}(M^T M) = \mathscr{R}(M^T)$, then x^* is the minimum norm solution of (4). It is obvious that x_k generated by Algorithm LSQR belongs to $\mathscr{R}(M^T)$ and this leads the following result.

Theorem 3.1. *The solution generated by Algorithm LSQR is the minimum norm solution of Eq.(4).*

Remark 3.1. Theoretically, when $\beta_{k+1} = 0$ or $\alpha_{k+1} = 0$ for some $k < \min\{m,n\}$, then recursions will stop. In both cases, x_k is the minimum norm least squares solution to Eq.(4). Also notice that $\|M^T(f - Mx_k)\|_2 = |\alpha_{k+1}\bar{\zeta}_{k+1}c_k| = 0$ is monotonically decreasing when k is increasing. Also notice that, at each step of the LSQR iteration, the main costs of computations are two matrix-vector products.

Remark 3.2. During the iterative processing, because of the round-off error, computed solution $\widehat{x_k}$ may make $\|M^T(f - M\widehat{x_k})\|_2 \neq 0$ even $\|M^T(f - Mx_k)\|_2 = |\alpha_{k+1}\bar{\zeta}_{k+1}c_k| = 0$. Therefore we need to setup the stopping criteria to check the correct k. Paige and Sauders Paige and Saunders (1982) discuss several choices of the stopping criteria. Sometimes we need to use restart strategy to improve the accuracy. In the numerical experiments provided in §5, we use $|\alpha_{k+1}\bar{\zeta}_{k+1}c_k| < \tau = 10^{-11}$ as the stopping criterion. We observe that this stopping criterion works well.

4 The matrix-form LSQR algorithm

In this section, we will propose an iterative method for the minimum-norm symmetric solutions of Eq.(1) and Eq.(2).

Since

$$\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}(X),$$

where $\otimes$ denote the Kronecker product, then we have $\mathrm{vec}(AXB) = (B^T \otimes A)\mathscr{F}\mathrm{vec}_i(X)$ and the problem Eq.(1) is equivalent to

$$\|Mx - f\|_2 = \min, \tag{6}$$

where

$$M = (B^T \otimes A)\mathscr{F} \in \mathscr{R}^{lm\times N}, x = \mathrm{vec}_i(X) \in \mathscr{R}^N, f = \mathrm{vec}(E) \in \mathscr{R}^{lm}.$$

The normal equation of (6) is $M^T Mx = M^T f$, which is equivalent to

$$\mathscr{F}^T(B \otimes A^T)(B^T \otimes A)\mathscr{F}\mathrm{vec}_i(X) = \mathscr{F}^T(B \otimes A^T)\mathrm{vec}(E).$$

From Theorem 2.1 and Theorem 2.2, the above formula leads to the following results.

Lemma 4.1. *The normal equation of Eq.(1) is*

$$A^TAXBB^T + BB^TXA^TA = A^TEB^T + BE^TA.$$

The vector iterations of LSQR will be rewritten into matrix form so that the kronecker product and $\mathscr{F}$ 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 $u = \text{vec}(U) \in \mathscr{R}^{ml}$. Further, we must guarantee that matrix form of M^Tu is symmetric.
For any $v \in \mathscr{R}^N$, let $V \in \mathscr{S}\mathscr{R}^{n\times n}$ satisfy $\text{vec}_i(V) = v$. Then we can obtain that the matrix form of Mv is

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

For deriving the matrix form of M^Tu, we need the following result.

Theorem 4.1. *Suppose that $U \in \mathscr{R}^{m\times l}$ and $Z = B^TUA$. Then we have*

$$M^T\text{vec}(U) = \text{vec}(Z + Z^T).$$

Proof. It follows from Theorem 2.1 and Theorem 2.2 that

$$\begin{aligned}\text{mat}(M^T\text{vec}(U)) &= \text{mat}(\mathscr{F}^T(B\otimes A^T)\text{vec}(U))\\ &= \text{mat}(\mathscr{F}^T\text{vec}(A^TUB^T)) = \text{mat}(\mathscr{F}^T\mathscr{F}\mathscr{F}^\dagger\text{vec}(Z))\\ &= \text{mat}(\mathscr{F}^T\mathscr{F}\text{vec}_i(\frac{Z+Z^T}{2})) = \text{mat}(2I_N\text{vec}_i(\frac{Z+Z^T}{2}))\\ &= Z + Z^T.\square\end{aligned}$$

For any $u \in \mathscr{R}^{lm}$, let $U \in \mathscr{R}^{l\times m}$ satisfy $u = \text{vec}(U)$ and define $Z = B^TUA$ Then we have

$$\text{mat}(M^Tu) = \text{mat}(M^T\text{vec}(U)) = Z + Z^T.$$

Now we can give the following algorithm.
Algorithm LSQR-M-S
(1)Initialization.
$X_0 = 0(\in \mathscr{R}^{n\times n}),\ \beta_1 = \|E\|_F,\ U_1 = E/\beta_1,$

$Z_1 = A^T U_1 B^T$, $\bar{V}_1 = Z_1 + Z_1^T$, $\alpha_1 = \|\bar{V}_1\|_S$, $V_1 = \bar{V}_1/\alpha_1$,
$H_1 = V_1$, $\bar{\zeta}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$.
(2)Iteration. For $i = 1, 2, \cdots$
$\bar{U}_{i+1} = AV_iB - \alpha_i U_i$, $\beta_{i+1} = \|\bar{U}_{i+1}\|_F$, $U_{i+1} = \bar{U}_{i+1}/\beta_{i+1}$,
$Z_{i+1} = A^T U_{i+1} B^T$, $\bar{V}_{i+1} = Z_{i+1} + Z_{i+1}^T - \beta_{i+1} V_i$,
$\alpha_i = \|\bar{V}_{i+1}\|_S$, $V_{i+1} = \bar{V}_{i+1}/\alpha_{i+1}$,
$\rho_i = \sqrt{\bar{\rho}_i^2 + \beta_{i+1}^2}$, $c_i = \bar{\rho}_i/\rho_i$, $s_i = \beta_{i+1}/\rho_i$, $\theta_{i+1} = s_i \alpha_{i+1}$,
$\bar{\rho}_{i+1} = -c_i \alpha_{i+1}$, $\zeta_i = c_i \bar{\zeta}_i$, $\bar{\zeta}_{i+1} = s_i \bar{\zeta}_i$,
$X_i = X_{i-1} + \zeta_i/\rho_i H_i$,
$H_{i+1} = V_{i+1} - \theta_i/\rho_i H_i$,
(3)check convergence.

Remark 4.1. Algorithm LSQR-M-S can compute X with minimal $\|\mathrm{vec}_i(X)\|_2$. Because $\|X\|_F^2 = 2\|\mathrm{vec}_i(X)\|_2^2$, X computed by Algorithm LSQR-M-S is the minimum Frobenius norm solution of Eq.(1) or Eq.(2).

5 Numerical examples

In Qiu, Zhang and Lu (2007), the authors have pointed out that the matrix-form LSQR method needs less flops than the matrix-form CGLS method in each iteration. In this section, we will compare our matrix-form LSQR algorithm and the matrix-form CGLS algorithm considered in Peng (2005); Peng, Hu and Zhang (2005), and show that our algorithm can find the minimum-norm solution more efficiently.

Example 5.1. The example given in Peng, Hu and Zhang (2005) is that

$$A = \begin{pmatrix} 1 & 3 & -5 & 7 & -5 \\ 3 & 0 & 4 & 1 & -1 \\ 0 & -2 & 9 & 6 & 8 \\ 11 & 6 & 2 & 17 & -13 \\ -5 & 5 & -22 & -1 & -11 \\ 9 & 4 & -6 & -9 & -19 \end{pmatrix}, B = \begin{pmatrix} 4 & 0 & 4 & -5 & 4 \\ -1 & 5 & 0 & -2 & 3 \\ 3 & -1 & 0 & 3 & 5 \\ 0 & 3 & 9 & 2 & -6 \\ -2 & 7 & -8 & 1 & 11 \end{pmatrix},$$

$$E = \begin{pmatrix} -279 & 242 & -554 & 132 & 238 \\ 28 & -130 & -179 & 8 & 105 \\ -87 & 176 & -58 & 244 & 60 \\ -474 & 94 & -1645 & 288 & 791 \\ -248 & 326 & -138 & -128 & -32 \\ 258 & -742 & -421 & -464 & 195 \end{pmatrix}.$$

Matrix equation $AXB = E$ is consistent and has infinite symmetric solution. By our algorithm, we can obtain the minimum-norm symmetric solution as follows

$$X_{12} = \begin{pmatrix} 0.2947 & -1.9916 & 1.1226 & -5.1217 & -0.0858 \\ -1.9916 & 1.6254 & -2.9704 & 1.4939 & -0.3997 \\ 1.1226 & -2.9704 & 0.1950 & 0.1664 & -0.2461 \\ -5.1217 & 1.4939 & 0.1664 & -0.1228 & 3.8844 \\ -0.0858 & -0.3997 & -0.2461 & 3.8844 & 1.1435 \end{pmatrix}$$

with the residual error $\xi_{12} = \|R_{12}\|_F = \|AX_{12}B - E\|_F = 3.1918e-012$.

Using the algorithm of Peng, Hu and Zhang (2005) and iterating 15 steps, the authors got the unique minimum-norm solution with the residual error $\|R_{12}\|_F = \|AX_{15}B - E\|_F = 4.6978e-012$. This solution is the same as the above X_{12} when four decimals are contained.

As I said early, our algorithm needs less flops.

Example 5.2. The example given in Peng (2005) is that

$$A = \begin{pmatrix} 4 & 3 & -1 & 3 & 1 & -3 & 2 \\ 3 & -2 & 3 & -4 & 3 & 2 & 1 \\ 4 & 3 & -1 & 3 & 1 & -3 & 2 \\ 3 & -1 & 3 & -1 & 3 & 2 & 1 \\ 4 & 3 & -1 & 3 & 1 & -3 & 2 \\ 3 & -1 & 3 & -1 & 3 & 2 & 1 \end{pmatrix}, B = \begin{pmatrix} -3 & 4 & -3 & -3 & 4 & 4 \\ 5 & -3 & 5 & 5 & -3 & -3 \\ -6 & 2 & -6 & -6 & 2 & 2 \\ -8 & 4 & -8 & -8 & 4 & 4 \\ 4 & -5 & 4 & 3 & -2 & -7 \\ -3 & 2 & -3 & -3 & 2 & 2 \\ -1 & -2 & -1 & -1 & -2 & -2 \end{pmatrix},$$

$$E = \begin{pmatrix} 43 & -54 & 73 & -54 & 51 & -54 \\ -31 & 37 & -61 & 37 & -53 & 37 \\ 43 & -54 & 73 & -54 & 51 & -54 \\ -31 & 37 & -61 & 37 & -53 & 37 \\ 47 & -54 & 73 & -54 & 21 & -54 \\ -31 & 27 & -61 & 27 & -53 & 27 \end{pmatrix}.$$

Our algorithm is very efficient for this example and, using 22 iterations, computes the minimum-norm solution

$$X_{17} = \begin{pmatrix} 1.0650 & 0.2510 & -0.9062 & 0.6469 & 0.6130 & -1.8154 & 0.5729 \\ 0.2510 & -0.6516 & -0.0189 & 0.4239 & 1.8937 & 0.8660 & -1.3207 \\ -0.9062 & -0.0189 & 1.9641 & 0.3755 & -2.2609 & 0.4210 & 2.2353 \\ 0.6469 & 0.4239 & 0.3755 & -0.3307 & -0.2146 & -0.4136 & 1.0401 \\ 0.6130 & 1.8937 & -2.2609 & -0.2146 & -2.6651 & -4.3017 & 2.3216 \\ -1.8154 & 0.8660 & 0.4210 & -0.4136 & -4.3017 & -1.0648 & 2.4271 \\ 0.5729 & -1.3207 & 2.2353 & 1.0401 & 2.3216 & 2.4271 & -0.4410 \end{pmatrix}$$

with the normal equation error

$$\eta_{17} = \|A^T E B^T + B E^T A - A^T A X_{17} B B^T - B B^T X_{17} A^T A\|_F = 4.0136e-012$$

and the residual error $\|R_{17}\|_F = \|AX_{17}B - E\|_F = 179.0445$.

Using Algorithm 2.1 of Peng (2005) and iterating 58 steps, the author got the unique minimum-norm solution with the normal equation error $\eta_{58} = 2572e-011$ and the residual error $\|R_{58}\|_F = 179.0445$.

Figure 1 plots the functions of $\log 10(\xi_k)$ and $\log 10(\eta_k)$ in Example 1 and Example 2.

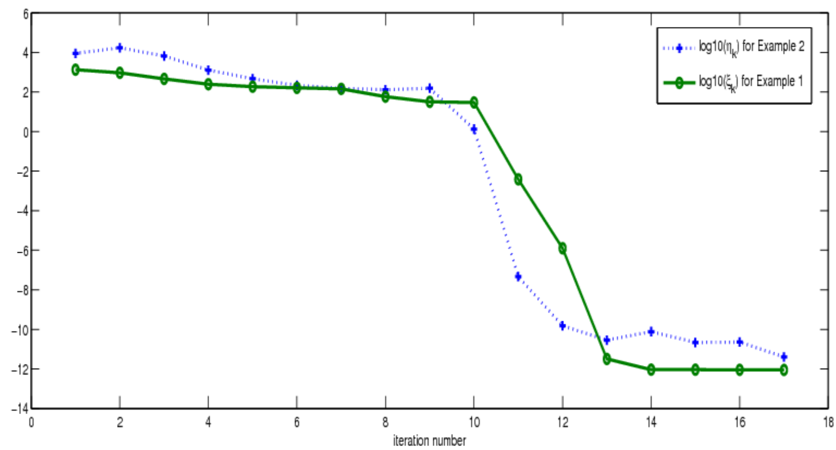

Figure 1: Error and iteration number for Example 1 and Example 2

Acknowledgement: The authors are grateful to two anonymous referees for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China(Grant No:11001144) and the Science and Technology Program of Shandong Universities of China (J10LA06,J11LA04).

References

Xie, D.; Sheng, Y.; Hu, X. (2003): The least-squares solution of inconsistent matrix equation over symmetric and antisymmetric matrices. *Appl. Math. Letters*, vol. 16, pp. 589–598.

Yuan, Y.; Dai, H. (2007): Inverse problems for symmetric matrices with a submatrix constraint. *Appl. Numer. Math.*, vol. 57, pp. 646–656.

Hou, J.; Peng, Z.; Zhang, X. (2006): An iterative method for the least squares symmetric solution of matrix equation $AXB = C$. *ANumer. Algor.*, vol. 42, pp. 181–192.

Peng, Z. (2005): An iterative method for the least squares symmetric solution of the linear matrix equation $AXB = C$. *Appl. Math. Comput.*, vol. 170, pp. 711–723.

Peng, Y.; Hu, X.; Zhang, L. (2005): An iteration method for the symmetric solutions and the optimal approximation solution of the matrix equation $AXB = C$. *Appl. Math. Comput.*, vol. 160, pp. 763–777.

Qiu, Y.; Zhang. Z.; Lu, J. (2007): Matrix iterative solutions to the least squares problem of $BXA^T = F$ with some linear constraints. *Appl. Math. Comput.*, vol. 185, pp. 284–300.

Stiefel, E. (1952): Ausgleichung ohne Aufstellung der Gausschen Normalgleichungen. *Wiss. Z. Tech. Hochsch. Dresden*, vol. 2, pp. 441–442.

Paige, C.; Saunders, A. (1982): LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, vol. 8, pp. 43–71.