Filomat 35:9 (2021), 3105–3111
https://doi.org/10.2298/FIL2109105H

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Matrix LSQR Algorithms for Solving Constrained Quadratic Inverse Eigenvalue Problem

Masoud Hajarian[a]

[a]*Department of Applied Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran.*

Abstract. The inverse eigenvalue problem appears in many applications such as control design, seismic tomography, exploration and remote sensing, molecular spectroscopy, particle physics, structural analysis, and mechanical system simulation. This paper investigates the matrix form of LSQR methods for solving the quadratic inverse eigenvalue problem with partially bisymmetric matrices under a prescribed submatrix constraint. In order to illustrate the effectiveness and feasibility of our results, one numerical example is presented.

1. Introduction

Notation: Throughout this paper, we assume that $\mathbf{R}^{m\times n}$, $I_n = (e_1, e_2, ..., e_n)$, $S_n = (e_n, e_{n-1}, ..., e_1)$, $\|A\| = \sqrt{\mathrm{trace}(A^TA)}$ and $D_{p,n} = \{d = (d_1, d_2, ..., d_p) : 1 \leq d_1 < d_2 < ... < d_p \leq n\}$ respectively represent the $m\times n$ real matrix set, the $n\times n$ unit matrix, the $n\times n$ reverse unit matrix, the Frobenius norm of the matrix A and the strictly increasing sequences of p elements from $1, 2, ..., n$. For $s = (s_1, s_2, ..., s_p) \in D_{p,n}$, $t = (t_1, t_2, ..., t_q) \in D_{q,n}$ and $u = (u_1, u_2, ..., u_r) \in D_{r,n}$, we assume that $E_s = (e_{s_1}, e_{s_2}, ..., e_{s_p}) \in \mathbf{R}^{n\times p}$, $E_t = (e_{t_1}, e_{t_2}, ..., e_{t_q}) \in \mathbf{R}^{n\times q}$ and $E_u = (e_{u_1}, e_{u_2}, ..., e_{u_r}) \in \mathbf{R}^{n\times r}$. The symbol $A[s|t]$ exhibits the $p\times q$ submatrix of A determined by rows indexed by s and columns indexed by t. The notation $A[\bar{s}|\bar{t}]$ represents the $(m-p)\times(n-q)$ submatrix of A determined by deleting rows indexed by s and columns indexed by t. A real $n\times n$ matrix $A = (a_{i,j})$ is named a bisymmetric matrix if its elements satisfy the properties $a_{i,j} = a_{j,i}$ and $a_{i,j} = a_{n-j+1,n-i+1}$ for $1 \leq i, j \leq n$. Let $\mathbf{BSR}^{n\times n}$ denote the set of $n\times n$ bisymmetric matrices. It can be verified that a matrix $X \in \mathbf{BSR}^{n\times n}$ if and only if $X = X^T = S_nXS_n$. The bisymmetric matrices including symmetric Toeplitz matrices and persymmetric Hankel matrices as special cases have wide applications in applied sciences [1, 4].
Various types of inverse eigenvalue problem such as

$$\begin{cases} \text{Given } X \in \mathbf{R}^{n\times m}, \text{ and } \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m) \in \mathbf{R}^{m\times m} \\ \text{Find } \; C \in \mathbf{R}^{n\times n} \text{ such that } CX = X\Lambda, \end{cases} \qquad \textit{Inverse eigenvalue problem (IEP)},$$

$$\begin{cases} \text{Given } X \in \mathbf{R}^{n\times m}, \text{ and } \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m) \in \mathbf{R}^{m\times m} \\ \text{Find } \; C, B \in \mathbf{R}^{n\times n} \text{ such that } CX = BX\Lambda, \end{cases} \qquad \textit{Generalized IEP (GIEP)},$$

2020 *Mathematics Subject Classification*. Primary 15A24; 65H10; Secondary 15A69; 65F10
Keywords. Quadratic inverse eigenvalue problem; LSQR method; Partially bisymmetric matrix.
Received: 03 April 2018; Accepted: 10 August 2021
Communicated by Yimin Wei
Email address: m_hajarian@sbu.ac.ir (Masoud Hajarian)

$$\begin{cases} \textit{Given } X \in \mathbf{R}^{n\times m}, \textit{ and } \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m) \in \mathbf{R}^{m\times m} \\ \textit{Find } C, B, A \in \mathbf{R}^{n\times n} \textit{ such that } AX\Lambda^2 + BX\Lambda + CX = 0, \end{cases} \quad \textit{Quadratic IEP (QIEP)},$$

play a fundamental role in a variety of fields of wireless communications, pole assignment problem, inverse Sturm Liouville problem and quantum mechanics, signal and data processing [5]. The inverse eigenvalue problem attracted a lot of research attention over the last few years due to the growing importance of inverse problems [3, 6–8, 13]. In practice we usually require that the resulting matrix from a specific inverse eigenvalue problem is physically realizable and thus additional structural constraints are imposed [14]. Therefore so far several constrained and specific inverse eigenvalue problems have been studied [15–18]. Wei and Dai proposed two numerical algorithms to solve the inverse eigenvalue problem of Jacobi matrix [12]. In [9], the inverse eigenvalue problem and the associated optimal approximation problem for Hermitian reflexive matrices with respect to a normal $k+1$-potent matrix were studied considered.
The present article deals with the quadratic inverse eigenvalue problem with partially bisymmetric matrices under a prescribed submatrix constraint. We will develop the LSQR methods for a constrained quadratic inverse eigenvalue problem as follows:

Problem 1. *Given* $X \in \mathbf{R}^{n\times m}$, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m) \in \mathbf{R}^{m\times m}$, $s = (s_1, s_2, ..., s_p, n+1-s_p, ..., n+1-s_2, n+1-s_1) \in D_{2p,n}$, $t = (t_1, t_2, ..., t_q, n+1-t_q, ..., n+1-t_2, n+1-t_1) \in D_{2q,n}$, $u = (u_1, u_2, ..., u_r, n+1-u_r, ..., n+1-u_2, n+1-u_1) \in D_{2r,n}$, $A_p \in \mathbf{R}^{2p\times 2p}$, $B_q \in \mathbf{R}^{2q\times 2q}$ *and* $C_r \in \mathbf{R}^{2r\times 2r}$, *let* $S_1 = \{X | X[s,s] = A_p,\ X[\bar{s},\bar{s}] \in \mathbf{BSR}^{(n-2p)\times(n-2p)}\}$, $S_2 = \{X | X[t,t] = B_q,\ X[\bar{t},\bar{t}] \in \mathbf{BSR}^{(n-2q)\times(n-2q)}\}$ *and* $S_3 = \{X | X[u,u] = C_r,\ X[\bar{u},\bar{u}] \in \mathbf{BSR}^{(n-2r)\times(n-2r)}\}$, *find* $A \in S_1$, $B \in S_2$ *and* $C \in S_3$ *such that* $AX\Lambda^2 + BX\Lambda + CX = 0$.

2. Main Results

In this section, first we propose a simplified form of Problem 1. By introducing the sets

$$S_1 = \underbrace{\{X | X[s,s] = A_p,\ X[\bar{p},\bar{p}] = 0\}}_{=\widetilde{S}_1} \oplus \underbrace{\{X | X \in \mathbf{BSR}^{n\times n},\ X[s,s] = 0\}}_{=\widehat{S}_1},$$

$$S_2 = \underbrace{\{X | X[t,t] = B_q,\ X[\bar{q},\bar{q}] = 0\}}_{=\widetilde{S}_2} \oplus \underbrace{\{X | X \in \mathbf{BSR}^{n\times n},\ X[t,t] = 0\}}_{=\widehat{S}_2},$$

$$S_3 = \underbrace{\{X | X[u,u] = C_r,\ X[\bar{u},\bar{u}] = 0\}}_{=\widetilde{S}_3} \oplus \underbrace{\{X | X \in \mathbf{BSR}^{n\times n},\ X[u,u] = 0\}}_{=\widehat{S}_3},$$

Problem 1 can be transformed into the following equivalent problem.

Problem 2. *Given* $X \in \mathbf{R}^{n\times m}$, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_m) \in \mathbf{R}^{m\times m}$, $s = (s_1, s_2, ..., s_p, n+1-s_p, ..., n+1-s_2, n+1-s_1) \in D_{2p,n}$, $t = (t_1, t_2, ..., t_q, n+1-t_q, ..., n+1-t_2, n+1-t_1) \in D_{2q,n}$, $u = (u_1, u_2, ..., u_r, n+1-u_r, ..., n+1-u_2, n+1-u_1) \in D_{2r,n}$, $A_p \in \mathbf{R}^{2p\times 2p}$, $B_q \in \mathbf{R}^{2q\times 2q}$ *and* $C_r \in \mathbf{R}^{2r\times 2r}$, *find* $\widehat{A} \in \widehat{S}_1$, $\widehat{B} \in \widehat{S}_2$ *and* $\widehat{C} \in \widehat{S}_3$ *such that*

$$\widehat{A}X\Lambda^2 + \widehat{B}X\Lambda + \widehat{C}X = \widehat{Z}, \tag{1}$$

where $\widehat{Z} = -\widehat{A}_pX\Lambda^2 - \widehat{B}_qX\Lambda - \widehat{C}_rX$, *in which* $\widehat{A}_p$, $\widehat{B}_q$ *and* $\widehat{C}_r$ *denote the matrices satisfying* $\widehat{A}_p[s,s] = A_p$, $\widehat{B}_q[t,t] = B_q$, $\widehat{C}_r[u,u] = C_r$ *and zeros elsewhere.*

It is obvious that $\widehat{A}^*$, $\widehat{B}^*$ and $\widehat{C}^*$ are the solutions of Problem 2 iff $\widehat{A}^* + \widehat{A}_p$, $\widehat{B}^* + \widehat{B}_p$ and $\widehat{C}^* + \widehat{C}_p$ are the solutions of Problem 1. Now we can solve Problem 2 more easier than Problem 1.
By using the Golub-Kahan bidiagonalization process, two types of the LSQR method were constructed in [10] to compute an approximation solution of the linear systems $Ax = b$ and unconstrained least-squares problem $\min_x \|Ax - b\|$. Two types of the LSQR method can be summarized as follows [10, 11].

Algorithm 1. ***Type 1 of LSQR method***
$\tau(0) = 1$; $\xi(0) = -1$; $\omega(0) = 0$; $w(0) = 0$; $z(0) = 0$; $\beta(1) = \|b\|$; $\beta(1)u(1) = b$; $\alpha(1) = \|A^T u(1)\|$; $\alpha(1)v(1) = A^T u(1)$;
For $i = 1, 2, \ldots$, until convergence, do:
$\xi(i) = -\xi(i-1)\beta(i)/\alpha(i)$; $z(i) = z(i-1)+\xi(i)v(i)$; $w(i) = (\tau(i-1)-\beta(i)w(i-1))/\alpha(i)$; $\omega(i) = \omega(i-1)+w(i)v(i)$; $\beta(i+1) = \|Av(i)-\alpha(i)u(i)\|$; $\beta(i+1)u(i+1) = Av(i)-\alpha(i)u(i)$; $\tau(i) = -\tau(i-1)\alpha(i)/\beta(i+1)$; $\alpha(i+1) = \|A^T u(i+1)-\beta(i+1)v(i)\|$; $\alpha(i+1)v(i+1) = A^T u(i+1) - \beta(i+1)v(i)$; $\gamma(i) = \beta(i+1)\xi(i)/(\beta(i+1)w(i) - \tau(i))$; $x(i) = z(i) - \gamma(i)\omega(i)$.

Algorithm 2. ***Type 2 of LSQR method***
$\theta(1) = \|A^T b\|$; $\theta(1)v(1) = A^T b$; $\rho(1) = \|Av(1)\|$; $\rho(1)p(1) = Av(1)$; $\omega(1) = v(1)/\rho(1)$; $\xi(1) = \theta(1)/\rho(1)$; $x(1) = \xi(1)\omega(1)$;
For $i = 1, 2, \ldots$, until convergence, do:
$\theta(i+1) = \|A^T p(i) - \rho(i)v(i)\|$; $\theta(i+1)v(i+1) = A^T p(i) - \rho(i)v(i)$; $\rho(i+1) = \|Av(i+1) - \theta(i+1)p(i)\|$; $\rho(i+1)p(i+1) = Av(i+1) - \theta(i+1)p(i)$; $\omega(i+1) = (v(i+1) - \theta(i+1)\omega(i))/\rho(i+1)$; $\xi(i+1) = -\xi(i)\theta(i+1)/\rho(i+1)$; $x(i+1) = x(i) + \xi(i+1)\omega(i+1)$.

Theorem 1. *[11] LSQR algorithms return the minimum-norm solution.*

In the above algorithms, the scalars $\alpha(i) \geq 0$, $\beta(i) \geq 0$, $\rho(i) \geq 0$ and $\theta(i) \geq 0$ are chosen to make $\|u(i)\| = 1$ and $\|v(i)\| = 1$, respectively [11].
Now we propose the matrix form of the above algorithms for solving Problem 2. With the aid of Kronecker product and vectorization operator, it is easy to see that solvability of Problem 2 is equivalent to the following system:

$$\underbrace{\begin{pmatrix} (X\Lambda^2)^T \otimes I_n & (X\Lambda)^T \otimes I_n & X^T \otimes I_n \\ I_n \otimes (X\Lambda^2)^T & I_n \otimes (X\Lambda)^T & I_n \otimes X^T \\ (X\Lambda^2)^T S_n \otimes S_n & (X\Lambda)^T S_n \otimes S_n & X^T S_n \otimes S_n \\ S_n \otimes (X\Lambda^2)^T S_n & S_n \otimes (X\Lambda)^T S_n & S_n \otimes X^T S_n \\ E_s^T \otimes E_s^T & 0 & 0 \\ 0 & E_t^T \otimes E_t^T & 0 \\ 0 & 0 & E_u^T \otimes E_u^T \\ E_s^T S_n \otimes E_s^T S_n & 0 & 0 \\ 0 & E_t^T S_n \otimes E_t^T S_n & 0 \\ 0 & 0 & E_u^T S_n \otimes E_u^T S_n \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \mathrm{vec}(\widehat{A}) \\ \mathrm{vec}(\widehat{B}) \\ \mathrm{vec}(\widehat{C}) \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} \mathrm{vec}(\widehat{Z}) \\ \mathrm{vec}(\widehat{Z}^T) \\ \mathrm{vec}(\widehat{Z}) \\ \mathrm{vec}(\widehat{Z}^T) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{b}. \tag{2}$$

If we substitute the above system into Algorithms 1 and 2 then we obtain the following matrix LSQR algorithms for solving Problem 2.

Algorithm 3. ***Type 1 of matrix LSQR method***
$\tau(0) = 1$; $\xi(0) = -1$; $\Omega_1(0) = 0$; $\Omega_2(0) = 0$; $\Omega_3(0) = 0$; $w(0) = 0$; $Z_1(0) = 0$; $Z_2(0) = 0$; $Z_3(0) = 0$;
$\beta(1) = 2\|\widehat{Z}\|$;
$\beta(1)U_1(1) = \widehat{Z}$; $U_2(1) = 0$; $U_3(1) = 0$; $U_4(1) = 0$;
$\alpha(1) = \Big(\|U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T + (U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T)^T + S_n(U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T + (U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T)^T)S_n\|^2 + \|U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T + (U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T)^T + S_n(U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T + (U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T)^T)S_n\|^2 + \|U_1(1)X^T + E_u U_4(1)E_u^T + (U_1(1)X^T + E_u U_4(1)E_u^T)^T + S_n(U_1(1)X^T + E_u U_4(1)E_u^T + (U_1(1)X^T + E_u U_4(1)E_u^T)^T)S_n\|^2\Big)^{1/2}$;
$\alpha(1)V_1(1) = U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T + (U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T)^T + S_n(U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T + (U_1(1)(X\Lambda^2)^T + E_s U_2(1)E_s^T)^T)S_n$;
$\alpha(1)V_2(1) = U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T + (U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T)^T + S_n(U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T + (U_1(1)(X\Lambda)^T + E_t U_3(1)E_t^T)^T)S_n$;

$\alpha(1)V_3(1) = U_1(1)X^T+E_uU_4(1)E_u^T+(U_1(1)X^T+E_uU_4(1)E_u^T)^T+S_n(U_1(1)X^T+E_uU_4(1)E_u^T+(U_1(1)X^T+E_uU_4(1)E_u^T)^T)S_n;$
For $i = 1, 2, ...,$ until convergence, do:
$\xi(i) = -\xi(i-1)\beta(i)/\alpha(i);$
$Z_1(i) = Z_1(i-1) + \xi(i)V_1(i); Z_2(i) = Z_2(i-1) + \xi(i)V_2(i); Z_3(i) = Z_3(i-1) + \xi(i)V_3(i);$
$w(i) = (\tau(i-1) - \beta(i)w(i-1))/\alpha(i);$
$\Omega_1(i) = \Omega_1(i-1) + w(i)V_1(i); \Omega_2(i) = \Omega_2(i-1) + w(i)V_2(i); \Omega_3(i) = \Omega_3(i-1) + w(i)V_3(i);$
$\beta(i+1) = 2\Big(\|V_1(i)X\Lambda^2 + V_2(i)X\Lambda + V_3(i)X - \alpha(i)U_1(i)\|^2+\|E_s^TV_1(i)E_s - \alpha(i)U_2(i)\|^2 + \|E_t^TV_2(i)E_t - \alpha(i)U_3(i)\|^2 + \|E_u^TV_3(i)E_u - \alpha(i)U_4(i)\|^2\Big)^{1/2};$
$\beta(i+1)U_1(i+1) = V_1(i)X\Lambda^2 + V_2(i)X\Lambda + V_3(i)X - \alpha(i)U_1(i);$
$\beta(i+1)U_2(i+1) = E_s^TV_1(i)E_s - \alpha(i)U_2(i);$
$\beta(i+1)U_3(i+1) = E_t^TV_2(i)E_t - \alpha(i)U_3(i);$
$\beta(i+1)U_4(i+1) = E_u^TV_3(i)E_u - \alpha(i)U_4(i);$
$\tau(i) = -\tau(i-1)\alpha(i)/\beta(i+1);$
$\alpha(i+1) = \Big(\|U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T + (U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T)^T + S_n(U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T + (U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T)^T)S_n - \beta(i+1)V_1(i)\|^2 + \|U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T + (U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T)^T + S_n(U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T + (U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T)^T)S_n - \beta(i+1)V_2(i)\|^2 + \|U_1(i+1)X^T + E_uU_4(i+1)E_u^T + (U_1(i+1)X^T + E_uU_4(i+1)E_u^T)^T + S_n(U_1(i+1)X^T + E_uU_4(i+1)E_u^T + (U_1(i+1)X^T + E_uU_4(i+1)E_u^T)^T)S_n - \beta(i+1)V_3(i)\|^2\Big)^{1/2};$
$\alpha(i+1)V_1(i+1) = U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T + (U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T)^T + S_n(U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T + (U_1(i+1)(X\Lambda^2)^T + E_sU_2(i+1)E_s^T)^T)S_n - \beta(i+1)V_1(i);$
$\alpha(i+1)V_2(i+1) = U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T + (U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T)^T + S_n(U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T + (U_1(i+1)(X\Lambda)^T + E_tU_3(i+1)E_t^T)^T)S_n - \beta(i+1)V_2(i);$
$\alpha(i+1)V_3(i+1) = U_1(i+1)X^T + E_uU_4(i+1)E_u^T + (U_1(i+1)X^T + E_uU_4(i+1)E_u^T)^T + S_n(U_1(i+1)X^T + E_uU_4(i+1)E_u^T + (U_1(i+1)X^T + E_uU_4(i+1)E_u^T)^T)S_n - \beta(i+1)V_3(i);$
$\gamma(i) = \beta(i+1)\xi(i)/(\beta(i+1)w(i) - \tau(i));$
$\widehat{A}(i) = Z_1(i) - \gamma(i)\Omega_1(i);$
$\widehat{B}(i) = Z_2(i) - \gamma(i)\Omega_2(i);$
$\widehat{C}(i) = Z_3(i) - \gamma(i)\Omega_3(i).$

Algorithm 4. ***Type 2 of matrix LSQR method***
$\theta(1) = \Big(\|\widehat{Z}(X\Lambda^2)^T + (\widehat{Z}(X\Lambda^2)^T)^T + S_n(\widehat{Z}(X\Lambda^2)^T + (\widehat{Z}(X\Lambda^2)^T)^T)S_n\|^2 + \|\widehat{Z}(X\Lambda)^T + (\widehat{Z}(X\Lambda)^T)^T + S_n(\widehat{Z}(X\Lambda)^T + (\widehat{Z}(X\Lambda)^T)^T)S_n\|^2+\|\widehat{Z}X^T+(\widehat{Z}X^T)^T+S_n(\widehat{Z}X^T+(\widehat{Z}X^T)^T)S_n\|^2\Big)^{1/2}; \theta(1)V_1(1) = \widehat{Z}(X\Lambda^2)^T+(\widehat{Z}(X\Lambda^2)^T)^T+S_n(\widehat{Z}(X\Lambda^2)^T+(\widehat{Z}(X\Lambda^2)^T)^T)S_n;$
$\theta(1)V_2(1) = \widehat{Z}(X\Lambda)^T+(\widehat{Z}(X\Lambda)^T)^T+S_n(\widehat{Z}(X\Lambda)^T+(\widehat{Z}(X\Lambda)^T)^T)S_n; \theta(1)V_3(1) = \widehat{Z}X^T+(\widehat{Z}X^T)^T+S_n(\widehat{Z}X^T+(\widehat{Z}X^T)^T)S_n;$
$\rho(1) = 2\Big(\|V_1(1)X\Lambda^2 + V_2(1)X\Lambda + V_3(1)X\|^2 + \|E_s^TV_1(1)E_s\|^2 + \|E_t^TV_2(1)E_t\|^2 + \|E_u^TV_3(1)E_u\|^2\Big)^{1/2};$
$\rho(1)P_1(1) = V_1(1)X\Lambda^2 + V_2(1)X\Lambda + V_3(1)X;$
$\rho(1)P_2(1) = E_s^TV_1(1)E_s; \rho(1)P_3(1) = E_t^TV_2(1)E_t; \rho(1)P_4(1) = E_u^TV_3(1)E_u;$
$\Omega_1(1) = V_1(1)/\rho(1); \Omega_2(1) = V_2(1)/\rho(1); \Omega_3(1) = V_3(1)/\rho(1);$
$\xi(1) = \theta(1)/\rho(1); \widehat{A}(1) = \xi(1)\Omega_1(1); \widehat{B}(1) = \xi(1)\Omega_2(1); \widehat{C}(1) = \xi(1)\Omega_3(1);$
For $i = 1, 2, ...,$ until convergence, do:
$\theta(i+1) = \Big(\|P_1(i)(X\Lambda^2)^T + E_sP_2(i)E_s^T + (P_1(i)(X\Lambda^2)^T + E_sP_2(i)E_s^T)^T + S_n(P_1(i)(X\Lambda^2)^T + E_sP_2(i)E_s^T + (P_1(i)(X\Lambda^2)^T + E_sP_2(i)E_s^T)^T)S_n - \rho(i)V_1(i)\|^2 + \|P_1(i)(X\Lambda)^T + E_tP_3(i)E_t^T + (P_1(i)(X\Lambda)^T + E_tP_3(i)E_t^T)^T + S_n(P_1(i)(X\Lambda)^T + E_tP_3(i)E_t^T + (P_1(i)(X\Lambda)^T+E_tP_3(i)E_t^T)^T)S_n-\rho(i)V_2(i)\|^2+\|P_1(i)X^T+E_uP_4(i)E_u^T+(P_1(i)X^T+E_uP_4(i)E_u^T)^T+S_n(P_1(i)X^T+E_uP_4(i)E_u^T + (P_1(i)X^T + E_uP_4(i)E_u^T)^T)S_n - \rho(i)V_3(i)\|^2\Big)^{1/2};$
$\theta(i+1)V_1(i+1) = P_1(i)(X\Lambda^2)^T+E_sP_2(i)E_s^T+(P_1(i)(X\Lambda^2)^T+E_sP_2(i)E_s^T)^T+S_n(P_1(i)(X\Lambda^2)^T+E_sP_2(i)E_s^T+(P_1(i)(X\Lambda^2)^T + E_sP_2(i)E_s^T)^T)S_n - \rho(i)V_1(i);$

$\theta(i+1)V_2(i+1) = P_1(i)(X\Lambda)^T + E_tP_3(i)E_t^T + (P_1(i)(X\Lambda)^T + E_tP_3(i)E_t^T)^T + S_n(P_1(i)(X\Lambda)^T + E_tP_3(i)E_t^T + (P_1(i)(X\Lambda)^T + E_tP_3(i)E_t^T)^T)S_n - \rho(i)V_2(i)$; $\theta(i+1)V_3(i+1) = P_1(i)X^T + E_uP_4(i)E_u^T + (P_1(i)X^T + E_uP_4(i)E_u^T)^T + S_n(P_1(i)X^T + E_uP_4(i)E_u^T + (P_1(i)X^T + E_uP_4(i)E_u^T)^T)S_n - \rho(i)V_3(i)$;
$\rho(i+1) = 2\Big(\|V_1(i+1)X\Lambda^2 + V_2(i+1)X\Lambda + V_3(i+1)X - \theta(i+1)P_1(i)\|^2 + \|E_s^TV_1(i+1)E_s - \theta(i+1)P_2(i)\|^2 + \|E_t^TV_2(i+1)E_t - \theta(i+1)P_3(i)\|^2 + \|E_u^TV_3(i+1)E_u - \theta(i+1)P_4(i)\|^2\Big)^{1/2}$; $\rho(i+1)P_1(i+1) = V_1(i+1)X\Lambda^2 + V_2(i+1)X\Lambda + V_3(i+1)X - \theta(i+1)P_1(i)$; $\rho(i+1)P_2(i+1) = E_s^TV_1(i+1)E_s - \theta(i+1)P_2(i)$;
$\rho(i+1)P_3(i+1) = E_t^TV_2(i+1)E_t - \theta(i+1)P_3(i)$;
$\rho(i+1)P_4(i+1) = E_u^TV_3(i+1)E_u - \theta(i+1)P_4(i)$;
$\Omega_1(i+1) = (V_1(i+1) - \theta(i+1)\Omega_1(i))/\rho(i+1)$;
$\Omega_2(i+1) = (V_2(i+1) - \theta(i+1)\Omega_2(i))/\rho(i+1)$;
$\Omega_3(i+1) = (V_3(i+1) - \theta(i+1)\Omega_3(i))/\rho(i+1)$;
$\xi(i+1) = -\xi(i)\theta(i+1)/\rho(i+1)$;
$\widehat{A}(i+1) = \xi(i+1)\Omega_1(i+1)$;
$\widehat{B}(i+1) = \xi(i+1)\Omega_2(i+1)$;
$\widehat{C}(i+1) = \xi(i+1)\Omega_3(i+1)$.

Stopping criterion. To check convergence of Algorithms 3 and 4, we use the stopping criterion

$$\sqrt{\|\widehat{A}(i)X\Lambda^2 + \widehat{B}(i)X\Lambda + \widehat{C}(i)X - \widehat{Z}\|^2 + \|E_s^T\widehat{A}(i)E_s\|^2 + \|E_t^T\widehat{B}(i)E_t\|^2 + \|E_u^T\widehat{C}(i)E_u\|^2} \leq \texttt{tol},$$

where `tol` is a chosen fixed threshold.

3. Numerical results

In this section, to illustrate the effeteness of Algorithms 3 and 4, we give an example. The following example is taken from [2]. We consider the constrained generalized inverse eigenvalue problem $CX = BX\Lambda$ with the following parameters:

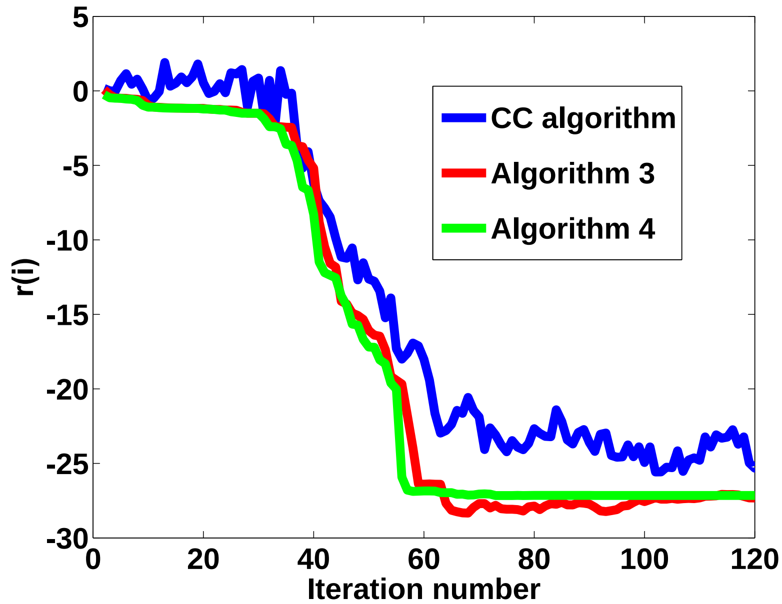
$$X = \begin{pmatrix} 0.1034 & 0.0869 & -0.4414 & 0.7860 & -0.9775 & -0.9975 \\ 0.3922 & 0.1724 & 0.6877 & -1.0000 & 1.0000 & -0.1738 \\ 0.9541 & 0.8176 & -0.8511 & 0.4781 & 0.4923 & 0.8610 \\ -1.0000 & 1.0000 & -1.0000 & -0.4625 & -0.4971 & 1.0000 \\ -0.3853 & 0.1767 & 0.7691 & 0.9927 & -0.9974 & -0.5748 \\ -0.1200 & 0.1535 & -0.6144 & -0.7733 & 0.9719 & -0.5570 \end{pmatrix},$$

$$\Lambda = \begin{pmatrix} 7.5462 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6732 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1659 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0719 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0221 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0368 \end{pmatrix}.$$

Considering $s = \{3,4\}$, $t = \{2,5\}$, $C_p = \begin{pmatrix} 0.05 & 0.15 \\ 0.2 & 0.1 \end{pmatrix}$, $B_q = \begin{pmatrix} 0.25 & -0.05 \\ -0.25 & 0.25 \end{pmatrix}$, gives us

$$\widehat{C}_p = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0.15 & 0 & 0 \\ 0 & 0 & 0.20 & 0.10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \widehat{B}_q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & -0.50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.25 & 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Figure 1: Residual plots.



Now we apply the iterative methods proposed in [2] (CC algorithm) with zero initial matrices and Algorithms 3 and 4 for solving this problem. The results plot in Figure 1 where

$$r(i) = \log \sqrt{\| - \widehat{B}(i)X\Lambda + \widehat{C}(i)X - \widehat{Z}\|^2 + \|E_t^T\widehat{B}(i)E_t\|^2 + \|E_u^T\widehat{C}(i)E_u\|^2}.$$

The solutions of Problem 1 can be computed as

$$\widehat{C}^* + \widehat{C}_p = \begin{pmatrix} -0.50 & -0.30 & -0.10 & 0.10 & 0.90 & 0.70 \\ -0.30 & -0.50 & 0.15 & 0.25 & 0.70 & 0.90 \\ -0.10 & 0.15 & 0.05 & 0.15 & 0.25 & 0.10 \\ 0.10 & 0.25 & 0.20 & 0.10 & 0.15 & -0.10 \\ 0.90 & 0.70 & 0.25 & 0.15 & -0.50 & -0.30 \\ 0.70 & 0.90 & 0.10 & -0.10 & -0.30 & -0.50 \end{pmatrix},$$

$$\widehat{B}^* + \widehat{B}_q = \begin{pmatrix} -0.00 & 1.50 & -0.00 & -2.00 & 4.50 & 8.00 \\ 1.50 & 0.25 & 0.00 & 0.00 & -0.50 & 4.50 \\ -0.00 & 0.00 & 0.25 & 0.50 & 0.00 & -2.00 \\ -2.00 & 0.00 & 0.50 & 0.25 & 0.00 & -0.00 \\ 4.50 & -0.25 & 0.00 & 0.00 & 0.25 & 1.50 \\ 8.00 & 4.50 & -2.00 & -0.00 & 1.50 & -0.00 \end{pmatrix}.$$

It is found from Figure 1 that Algorithms 3 and 4 are more efficient than the CC algorithm [2].

Acknowledgments

The author is grateful to Professor Jing Cai for sharing the M-file of [2].

References

[1] J. Bunch. Stability of methods for solving Toeplitz systems of equations. *SIAM Journal on Scientific and Statistical Computing*, 6:349–364, 1985.

[2] J. Cai and J. Chen. Iterative solutions of generalized inverse eigenvalue problem for partially bisymmetric matrices. *Linear and Multilinear Algebra*, 65:1643–1654, 2017.

[3] J. Cai and J. Chen. Least-squares solutions of generalized inverse eigenvalue problem over Hermitian-Hamiltonian matrices with a submatrix constraint. *Computational and Applied Mathematics*, 37:593–603, 2018.

[4] A. Cantoni and P. Butler. Properties of the eigenvectors of persymmetric matrices with applications to communication theory. *IEEE Transactions on Communications*, 24:804–809, 1976.

[5] M. Chu and G. Golub. *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*. Oxford University Press, New York, 2005.

[6] Z. Dalvand and M. Hajarian. Solving generalized inverse eigenvalue problems via L-BFGS-B method. *Inverse Problems in Science and Engineering*, 28:1719–1746, 2020.

[7] Z. Dalvand and M. Hajarian. Newton-like and inexact Newton-like methods for a parameterized generalized inverse eigenvalue problem. *Mathematical Methods in the Applied Sciences*, 44:4217–4234, 2021.

[8] Z. Dalvand, M. Hajarian, and J. E. Roman. An extension of the Cayley transform method for a parameterized generalized inverse eigenvalue problem. *Numerical Linear Algebra with Applications*, 27:e2327, 2020.

[9] S. Gigola, L. Lebtahi, and N. Thome. The inverse eigenvalue problem for a Hermitian reflexive matrix and the optimization problem. *Journal of Computational and Applied Mathematics*, 291:449–457, 2016.

[10] C. Paige. Bidiagonalization of matrices and solution of linear equation. *SIAM Journal on Numerical Analysis*, 11:197–209, 1974.

[11] C. Paige and M. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8:43–47, 1982.

[12] Y. Wei and H. Dai. An inverse eigenvalue problem for Jacobi matrix. *Applied Mathematics and Computation*, 251:633–642, 2015.

[13] J. Wu. On the realizability of open nonnegative inverse eigenvalue problems. *Applied Mathematics Letters*, 25:907–913, 2012.

[14] K. Yang. *Numerical Algorithms for Inverse Eigenvalue Problems arising in Control and Nonnegative Matrices*. PhD thesis, The Australian National University, Canberra, Australia, 2006.

[15] F. Yin and G.-X. Huang. Left and right inverse eigenvalue problem of (R, S)-symmetric matrices and its optimal approximation problem. *Applied Mathematics and Computation*, 219:9261–9269, 2013.

[16] S.-F. Yuan, Q.-W. Wang, and Z.-P. Xiong. Linear parameterized inverse eigenvalue problem of bisymmetric matrices. *Linear Algebra and its Applications*, 439:1990–2007, 2013.

[17] Y. Yuan and H. Dai. Solutions to an inverse monic quadratic eigenvalue problem. *Linear Algebra and its Applications*, 434:2367–2381, 2011.

[18] L. Zhao, X. Hu, and L. Zhang. Inverse eigenvalue problems for bisymmetric matrices under a central principal submatrix constraint. *Linear and Multilinear Algebra*, 59:117–128, 2011.