ISSN: 1017-060X (Print)

ISSN: 1735-8515 (Online)

Bulletin of the Iranian Mathematical Society

Vol. 40 (2014), No. 1, pp. 41–53

Title:

A matrix LSQR algorithm for solving constrained linear operator equations

Author(s):

M. Hajarian

Published by Iranian Mathematical Society
http://bims.ims.ir

Bull. Iranian Math. Soc.
Vol. 40 (2014), No. 1, pp. 41–53
Online ISSN: 1735-8515

A MATRIX LSQR ALGORITHM FOR SOLVING CONSTRAINED LINEAR OPERATOR EQUATIONS

M. HAJARIAN

(Communicated by Abbas Salemi)

Abstract. In this work, an iterative method based on a matrix form of LSQR algorithm is constructed for solving the linear operator equation $\mathcal{A}(X) = B$ and the minimum Frobenius norm residual problem $||\mathcal{A}(X) - B||_F$ where $X \in \mathcal{S} := \{X \in \mathsf{R}^{n\times n} \mid X = \mathcal{G}(X)\}$, $\mathcal{F}$ is the linear operator from $\mathsf{R}^{n\times n}$ onto $\mathsf{R}^{r\times s}$, $\mathcal{G}$ is a linear self-conjugate involution operator and $B \in \mathsf{R}^{r\times s}$. Numerical examples are given to verify the efficiency of the constructed method.

Keywords: Iterative method, LSQR method, linear operator equation.
MSC(2010): Primary: 15A06, 15A24; Secondary: 65F10 65F30, 65F10.

1. Introduction

Throughout this paper, we use A^T, $\mathrm{tr}(A)$ and $R(A)$ to denote the transpose, the trace and the column space of the matrix A, respectively. We define an inner product as $\langle A, B\rangle = \mathrm{tr}(B^T A)$, then the norm of a matrix A generated by this inner product is Frobenius norm and is denoted by $\langle A, A\rangle = ||A||_F^2$. If u is a vector, its euclidean norm is $||u||_2 = \sqrt{\langle u, u\rangle} = \sqrt{u^T u}$. For $A = (a_1, a_2, ..., a_n) \in \mathsf{R}^{m\times n}$, where a_i denotes the i-th column of A, we represent by $\mathrm{vec}(A) = (a_1^T, a_2^T, ..., a_n^T)^T$ vector expanded by columns of A. $\mathcal{I}$ stands for the identity operator on $\mathsf{R}^{n\times n}$. Once an inner product is defined, then for any linear operator $\mathcal{F}$ from $\mathsf{R}^{n\times n}$ onto $\mathsf{R}^{r\times s}$, there is another operator called the adjoint of $\mathcal{F}$, written $\mathcal{F}^{adj}$. What defines the adjoint is that for any two matrices

Article electronically published on February 25, 2014.
Received: 9 July 2012, Accepted: 6 December 2012.

$X \in \mathsf{R}^{n\times n}$ and $Y \in \mathsf{R}^{r\times s}$

$$\langle \mathcal{F}(X), Y\rangle = \langle X, \mathcal{F}^{adj}(Y)\rangle.$$

For example, it is easy to see that if $\mathcal{F} : X \to AX + XB$, then $\mathcal{F}^{adj}$ can be denoted as: $\mathcal{F}^{adj} : Y \to A^TY+YB^T$. $\mathcal{S}$ denotes the set of constrained matrices $X \in \mathsf{R}^{n\times n}$ defined by $X = \mathcal{G}(X)$ where $\mathcal{G}$ is a self-conjugate involution operator from $\mathsf{R}^{n\times n}$ onto $\mathsf{R}^{n\times n}$, i.e. $\mathcal{G} = \mathcal{G}^{adj}$ and $\mathcal{G}^2 = \mathcal{I}$. Let P be some real symmetric orthogonal $n \times n$ matrix. If $A = PAP$ ($A = -PAP$) then A is called generalized centro-symmetric matrix (generalized central anti-symmetric) with respect to P. $\mathsf{CSR}_P^{n\times n}$ ($\mathsf{CASR}_P^{n\times n}$) denotes the set of order n generalized centro-symmetric (generalized central anti-symmetric) matrices with respect to $P \in \mathsf{SOR}^{n\times n}$. It is worth noting that the set of constrained matrices $\mathcal{S}$ includes symmetric, skew-symmetric, centro-symmetric, central anti-symmetric generalized reflexive and generalized anti-reflexive matrices as special cases [1].

The linear matrix equations (including the Sylvester and Lyapunov matrix equations as special cases) play an important role in mathematics, physics and engineering. For example, the Lyapunov matrix equation

$$AX - XA^T = C, \tag{1.1}$$

plays an important role in the solution of Riccati matrix equations [21], stability analysis [22], H_∞ optimal control [15] and model reduction [26, 29]. Solutions to the Sylvester matrix equation

$$AX - XC = B, \tag{1.2}$$

can be used to parameterize the feedback gains in pole assignment problem for linear systems [37]. The Stein matrix equation

$$X = AXB + C, \tag{1.3}$$

is important in stability analysis and controller design in control theory. Also the Stein matrix equation can be applied for solving the Sylvester matrix (1.2). The generalized Sylvester matrix equation

$$AX - EXF = C, \tag{1.4}$$

can be used to achieve pole assignment, robust pole assignment and observer design for descriptor linear systems [14]. Due to wide applications of solutions to the Sylvester and Lyapunov matrix equations, the problem of searching for analytical and numerical solutions to these matrix equations has been well investigated in the literature [36, 31, 32, 33, 34, 25, 12, 25]. In [23], an iterative algorithm was constructed for solving linear matrix equation

$$AXB = C, \tag{1.5}$$

over generalized centro-symmetric matrix X. In [20], an iterative method was proposed to solve (1.5) over skew-symmetric matrix X. For when the Sylvester matrix equation (1.2) has a unique solution, Hu and Cheng [19] provided a closed form solution which was expressed as a polynomial of known matrices. In [40], Zhou et al. proposed the iterative solution to the Stein matrix equation (1.3). By applying the conjugate gradient method, Wang et al. [30] proposed two iterative algorithms to solve the matrix equation

$$AXB + CX^T D = E. \tag{1.6}$$

In [7, 8, 9, 10, 11], the gradient based iterative algorithms to solve several linear matrix equations were introduced by using the hierarchical identification principle. In order to guarantee the convergence of the proposed methods in [5, 8, 11, 35, 9, 10], we need to choose an appropriate convergence factor. In general, such a convergence factor can be obtained by some complex computation. Recently Zhou et al. [39] studied the solvability, existence of unique solution, closed-form solution and numerical solution of matrix equation $X = Af(X)B + C$ with $f(X) = X^T$, $f(X) = \overline{X}$ and $f(X) = X^H$. Gu and Xue [16] proposed a hierarchical identification method (SSHI) for solving Lyapunov matrix equations. Recently by extending the conjugate gradient least square (CGLS) approach, Dehghan and Hajarian proposed some efficient algorithms for solving several linear matrix equations [2, 3, 4, 6].

It is obvious that the various linear matrix equations such as (1.1)-(1.6) can be rewritten as

$$\mathcal{A}(X) = B, \tag{1.7}$$

where $\mathcal{A}$ denotes the linear operator. For example, the Sylvester matrix equation is equivalent to Equation (1.2), if we define the operator $\mathcal{A}$ as: $\mathcal{A} : X \to AX - XC$.

In this work, an iterative method based on a matrix form of the LSQR algorithm is introduced to solve the linear operator equation (1.7) and the minimum Frobenius norm residual problem

$$||\mathcal{A}(X) - B||_F, \tag{1.8}$$

where $\mathcal{A}$ is the linear operators from $\mathsf{R}^{n\times n}$ onto $\mathsf{R}^{r\times s}$, $B \in \mathsf{R}^{r\times s}$ and $X \in \mathcal{S} := \{X \in \mathsf{R}^{n\times n} \mid X = \mathcal{G}(X)\}$.

The rest of this paper is organized as follows. In Section 2, first we recall the LSQR algorithm and then by extending this algorithm we derive an efficient algorithm for solving the linear operator equation (1.7) and the minimum Frobenius norm residual problem (1.8). Section 3 gives some numerical results to illustrate the effectiveness of the algorithm derived. Also we give some conclusions in Section 4 to end this article.

2. **Main results**

In this sections, we first recall the LSQR algorithm [27, 28] for solving the linear systems

$$Mx = f, \tag{2.1}$$

and least-squares problem

$$||Mx - f||_2. \tag{2.2}$$

Then by extending the LSQR algorithm, we obtain an iterative method to solve the linear operator equation (1.7) and the minimum Frobenius norm residual problem (1.8).
Based on the bidiagonalization procedure of Golub and Kahan [17], the LSQR algorithm can be summarized as follows:

LSQR algorithm
(1) Initialization.
$x_0 = 0$, $\beta_1 u_1 = f$, $\alpha_1 v_1 = M^T u_1$, $w_1 = v_1$, $\overline{\phi}_1 = \beta_1$, $\overline{\rho}_1 = \alpha_1$.
(2) Iteration. For $i = 1, 2, ...,$ until the stopping criteria have been met.
(a) $\beta_{i+1}u_{i+1} = Mv_i - \alpha_i u_i$,
(b) $\alpha_{i+1}v_{i+1} = M^T u_{i+1} - \beta_{i+1}v_i$,
(c) $\rho_i = (\overline{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$,
(d) $c_i = \overline{\rho}_i/\rho_i$,
(e) $s_i = \beta_{i+1}/\rho_i$,
(f) $\theta_{i+1} = s_i\alpha_{i+1}$,
(g) $\rho_i = -c_i\alpha_{i+1}$,
(h) $\phi_i = c_i\overline{\phi}_i$,
(i) $\overline{\phi}_{i+1} = s_i\overline{\phi}_i$,
(j) $x_i = x_{i-1} + (\phi_i/\rho_i)w_i$,
(k) $w_{i+1} = v_{i+1} - (\phi_{i+1}/\rho_i)w_i$,

(l) Check the stopping criteria.

In the above algorithm, the scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen to make $||v_i||_2 = ||u_i||_2 = 1$. Also the sequence $\{x_i \in R(M^T)\}$ generated by the LSQR algorithm converges to the unique minimum l_2-norm solution (the unique minimum l_2-norm least squares solution) of the linear systems (2.1) (the least-squares problem (2.2)). Furthermore the bidiagonalization procedure of Golub and Kahan has finite termination property, the LSQR algorithm still keeps the finite termination property if exact arithmetic were used. Hence if we consider $\beta_1 u_1 = f - Mx_0$ instead of $\beta_1 u_1 = f$ ($0 \neq x_0 \notin R(M^T)$) in the LSQR algorithm, then the sequence $\{x_i \notin R(M^T)\}$ generated by the LSQR algorithm converges to a solution (a least-squares solution) of the linear systems (2.1) (the least-squares problem (2.2)) (for more details see [27, 28]).

In the rest of this section, we introduce a matrix the LSQR iterative algorithm based on the LSQR algorithm for solving (1.7) and (1.8). First we present the following useful lemma.

Lemma 2.1. [18] *Let $\mathcal{A} : R^{n\times n} \to R^{r\times s}$ be a given linear operator. Then there exists a unique matrix $T \in R^{rs\times n^2}$, such that* $\mathrm{vec}(\mathcal{A}(X)) = T\mathrm{vec}(X)$ *and* $\mathrm{vec}(\mathcal{A}^{adj}(Y)) = T^T\mathrm{vec}(Y)$ *for all $X \in R^{n\times n}$ and $Y \in R^{r\times s}$.*

We can easily prove that the linear operator equation (1.7) has the solution $X \in \mathcal{S}$ if and only if the pair of linear operator equations

$$\begin{cases} \mathcal{A}(X) = B, \\ \mathcal{A}(\mathcal{G}(X)) = B, \end{cases} \tag{2.3}$$

is consistent. From Lemma 2.1, there exist the matrices T and N such that $\mathrm{vec}(\mathcal{A}(X)) = T\mathrm{vec}(X)$ and $\mathrm{vec}(\mathcal{G}(X)) = N\mathrm{vec}(X)$ for $X \in \mathsf{R}^{n\times n}$. Hence the pair of linear operator equations (2.3) is equivalent to the following pair of linear equations:

$$\begin{cases} T\mathrm{vec}(X) = \mathrm{vec}(B), \\ TN\mathrm{vec}(X) = \mathrm{vec}(B), \end{cases} \tag{2.4}$$

i.e.

$$\begin{pmatrix} T \\ TN \end{pmatrix} \mathrm{vec}(X) = \begin{pmatrix} \mathrm{vec}(B) \\ \mathrm{vec}(B) \end{pmatrix}. \tag{2.5}$$

This implies that the system of matrix equations (2.3) can be transformed into the linear systems (2.1) with the coefficient matrix M and vector f as

$$M = \begin{pmatrix} T \\ TN \end{pmatrix} \quad \text{and} \quad f = \begin{pmatrix} \mathrm{vec}(B) \\ \mathrm{vec}(B) \end{pmatrix}. \tag{2.6}$$

It is obvious that the size of the system (2.1) with parameters (2.6) is large. However, iterative methods consume more computer time and memory once the size of the system is large. To overcome this complication, we extend the LSQR algorithm for solving the system (2.1) with parameters (2.6). Now by considering the linear systems (2.1) with parameters (2.6), we rewrite the vectors u_i, v_i, p_i, h_i, w_i, Av_i, $A^T p_i$ and $A^T u_i$ of the LSQR algorithm in the matrix forms. We can write

$$\beta_1 u_1 = f \to \beta_1 u_1 = \begin{pmatrix} \mathrm{vec}(B) \\ \mathrm{vec}(B) \end{pmatrix}, \tag{2.7}$$

$$\alpha_1 v_1 = M^T u_1 \to \alpha_1 v_1 = \begin{pmatrix} T \\ TN \end{pmatrix}^T u_1 = \begin{pmatrix} T^T & N^T T^T \end{pmatrix} u_1, \tag{2.8}$$

$$\beta_{i+1} u_{i+1} = M v_i - \alpha_i u_i \to \beta_{i+1} u_{i+1} = \begin{pmatrix} T \\ TN \end{pmatrix} v_i - \alpha_i u_i, \tag{2.9}$$

(2.10)
$$\alpha_{i+1} v_{i+1} = M^T u_{i+1} - \beta_{i+1} v_i \to \alpha_{i+1} v_{i+1} = \begin{pmatrix} T^T & N^T T^T \end{pmatrix} u_{i+1} - \beta_{i+1} v_i.$$

By considering the above equations, we define

$$x_i = \mathrm{vec}(X_i) \text{ and } u_i = \begin{pmatrix} \mathrm{vec}(U_i) \\ \mathrm{vec}(U_i) \end{pmatrix}, \tag{2.11}$$

$$v_i = \mathrm{vec}(V_i), \text{ and } w_i = \mathrm{vec}(W_i), \tag{2.12}$$

where $X_i, V_i, W_i \in \mathsf{R}^{n\times n}$ and $U_i \in \mathsf{R}^{r\times s}$. Hence we have

$$\beta_1 U_1 = B, \qquad \alpha_1 V_1 = [\mathcal{A}^{adj}(U_1) + \mathcal{G}\mathcal{A}^{adj}(U_1)], \tag{2.13}$$

$$\beta_{i+1} U_{i+1} = \mathcal{A}(V_i) - \alpha_i U_i, \tag{2.14}$$

$$\alpha_{i+1} V_{i+1} = \mathcal{A}^{adj}(U_{i+1}) + \mathcal{G}\mathcal{A}^{adj}(U_{i+1}) - \beta_{i+1} V_i. \tag{2.15}$$

By using (2.13)-(2.15), the matrix form of LSQR algorithm can be presented as:

Algorithm 2.2. ***Matrix LSQR iterative algorithm***
(1) Initialization. Set the initial matrix $X_0 = 0$. *Calculate*
$\beta_1 U_1 = B$, $\beta_1 = \sqrt{2}||B||_F$, $\alpha_1 V_1 = [\mathcal{A}^{adj}(U_1) + \mathcal{G}\mathcal{A}^{adj}(U_1)]$,
$\alpha_1 = ||\mathcal{A}^{adj}(U_1) + \mathcal{G}\mathcal{A}^{adj}(U_1)||_F$, $W_1 = V_1$, $\overline{\phi}_1 = \beta_1$, $\overline{\rho}_1 = \alpha_1$
(2) Iteration. For $i = 1, 2, ...$, *until the stopping criteria have been met.*
(a) $\beta_{i+1} U_{i+1} = \mathcal{A}(V_i) - \alpha_i U_i$, $\beta_{i+1} = \sqrt{2}||\mathcal{A}(V_i) - \alpha_i U_i||_F$,
(b) $\alpha_{i+1} V_{i+1} = \mathcal{A}^{adj}(U_{i+1}) + \mathcal{G}\mathcal{A}^{adj}(U_{i+1}) - \beta_{i+1} V_i$, $\alpha_{i+1} V_{i+1} = ||\mathcal{A}^{adj}(U_{i+1}) + \mathcal{G}\mathcal{A}^{adj}(U_{i+1}) - \beta_{i+1} V_i||_F$,
(c) $\rho_i = (\overline{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$,
(d) $c_i = \overline{\rho}_i / \rho_i$,
(e) $s_i = \beta_{i+1} / \rho_i$,
(f) $\theta_{i+1} = s_i \alpha_{i+1}$,
(g) $\overline{\rho}_{i+1} = -c_i \alpha_{i+1}$,
(h) $\phi_i = c_i \overline{\phi}_i$,
(i) $\overline{\phi}_{i+1} = s_i \overline{\phi}_i$,
(j) $X_i = X_{i-1} + (\phi_i / \rho_i) W_i$,
(k) $W_{i+1} = V_{i+1} - (\theta_{i+1} / \rho_i) W_i$,
(l) Check the stopping criteria.

In the above algorithm, the stopping criteria may be chosen as $\sqrt{||\mathcal{A}(X_i) - B||_F} < \varepsilon$ where ε is a small positive number.

3. Numerical reports

In this section, two numerical examples are performed to demonstrate the effectiveness of Algorithm 2.2. Computations were done on a PC Pentium IV using MATLAB 7.10.

Example 3.1. Consider the Sylvester matrix equation

$$AX + XB = C, \tag{3.1}$$

with the following parameters:

$$A = \begin{pmatrix} -510.1366 & -75.4476 & -60.9278 & -40.1649 & -5.7312 \\ -22.8827 & -515.1221 & -78.4018 & -92.6115 & -34.9339 \\ -60.0774 & -1.8319 & -539.2496 & -90.7735 & -80.5035 \\ -48.1123 & -81.3193 & -73.0825 & -474.1854 & -0.9763 \\ -88.2386 & -44.0256 & -17.4503 & -88.4713 & -428.5259 \end{pmatrix},$$

$$B = \begin{pmatrix} -385.7209 & -69.0919 & -49.1587 & -65.3625 & -71.9842 \\ -67.5401 & -429.1864 & -89.0771 & -33.8551 & -30.6197 \\ -29.9737 & -85.1411 & -403.3219 & -28.6829 & -83.0111 \\ -53.6257 & -84.5119 & -63.8461 & -350.4928 & -56.2392 \\ -14.9364 & -58.7627 & -80.9795 & -52.8738 & -431.5161 \end{pmatrix},$$

$$C = 10^6 \times \begin{pmatrix} 0.9262 & 0.1658 & 0.2283 & 0.1229 & 0.1091 \\ 0.1184 & 1.0036 & 0.1922 & 0.2830 & 0.0848 \\ 0.2214 & 0.1262 & 1.0456 & 0.1337 & 0.3242 \\ 0.1199 & 0.3109 & 0.1598 & 0.8027 & 0.0715 \\ 0.2497 & 0.1253 & 0.1367 & 0.1499 & 0.7565 \end{pmatrix}.$$

For these matrices, the generalized centro-symmetric solution of the Sylvester matrix equation (3.1) is

$$X^* = 10^3 \times \begin{pmatrix} -1.0203 & 0 & -0.1219 & 0 & -0.0115 \\ 0 & -1.0302 & 0 & -0.1852 & 0 \\ -0.1202 & 0 & -1.0785 & 0 & -0.1610 \\ 0 & -0.1626 & 0 & -0.9484 & 0 \\ -0.1765 & 0 & -0.0349 & 0 & -0.8571 \end{pmatrix},$$

where

$$P = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

By applying Algorithm 2.2 with $X_0 = 0$, we obtain the sequence $\{X_i\}$. The convergence behaviors of Algorithm 2.2 is displayed in Figure 1 where

$$r_i = \log_{10} ||C - AX_i - X_iB||_F,$$

and

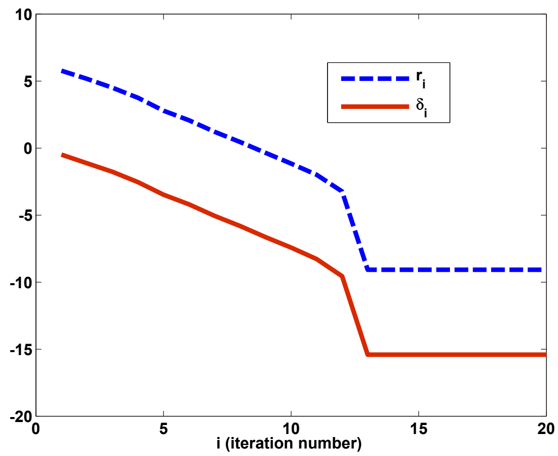
$$\delta_i = \log_{10} \frac{||X_i - X^*||_F}{||X^*||_F}.$$

It is obvious that both r_i and δ_i decrease, and converge to zero as i increases. The obtained results demonstrate the efficiency of Algorithm 2.2.

Example 3.2. In this example we consider the linear matrix equation

$$AX + X^TB = C, \tag{3.2}$$

FIGURE 1. The relative error of solution and the residual for Examples 3.1



with the following parameters:

$$A = \begin{pmatrix} -202.7372 & 68.5887 & 55.3889 & 36.5136 & 5.2102 \\ 20.8025 & -296.1286 & 71.2743 & 84.1923 & 31.7581 \\ 54.6158 & 1.6653 & -234.3006 & 82.5214 & 73.1850 \\ 43.7384 & 73.9266 & 66.4387 & -267.2290 & 0.8875 \\ 80.2169 & 40.0233 & 15.8640 & 80.4285 & -274.5687 \end{pmatrix},$$

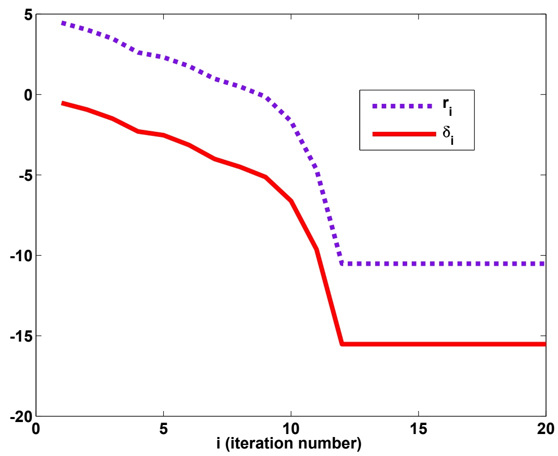
$$B = \begin{pmatrix} -39.2168 & 53.7382 & 38.2345 & 50.8375 & 55.9877 \\ 52.5312 & -44.5422 & 69.2822 & 26.3317 & 23.8153 \\ 23.3129 & 66.2209 & 43.8361 & 22.3089 & 64.5642 \\ 41.7089 & 65.7314 & 49.6581 & 10.9383 & 43.7416 \\ 11.6172 & 45.7043 & 62.9840 & 41.1241 & -47.5799 \end{pmatrix},$$

$$C = 10^4 \times \begin{pmatrix} 1.9250 & -1.3174 & 2.4552 & -1.4035 & 1.2612 \\ -2.4154 & 3.3302 & -1.6999 & 2.0085 & 0.3480 \\ 1.8480 & -1.8115 & 2.2958 & 0.5853 & 1.4943 \\ -1.6516 & 2.5759 & -0.5273 & 1.7031 & -1.5197 \\ 1.7065 & -1.0747 & 2.1178 & -1.1508 & 1.3903 \end{pmatrix}.$$

For the above matrices, the generalized central anti-symmetric solution of the matrix equation (3.2) is

$$X^* = \begin{pmatrix} 0 & 107.4764 & 0 & 101.6750 & 0 \\ 105.0624 & 0 & 138.5645 & 0 & 47.6307 \\ 0 & 132.4418 & 0 & 44.6178 & 0 \\ 83.4178 & 0 & 99.3162 & 0 & 87.4832 \\ 0 & 91.4087 & 0 & 82.2482 & 0 \end{pmatrix}.$$

FIGURE 2. The relative error of solution and the residual for Examples 3.2



By using Algorithm 2.2 with $X_0 = 0$, we obtain results shown in Figure 2 where

$$r_i = \log_{10} ||C - AX_i - X_i^T B||_F.$$

The results demonstrate that Algorithm 2.2 is quite efficient.

4. Concluding remarks

Linear matrix equations have numerous applications in control and system theory. Based on the LSQR algorithm, we have derived Algorithm 2.2 for solving the linear operator equation (1.7) and the minimum Frobenius norm residual problem (1.8). It is worth noting that Algorithm 2.2 are very simple and neat and they do not need to choose an appropriate convergence factor. The numerical results have shown that Algorithm 2.2 obtains results which are efficient and effective in practice.

Acknowledgments

The author thanks the Editor and anonymous reviewer for their valuable comments.

References

[1] H. C. Chen, Generalized reflexive matrices: special properties and applications, *SIAM J. Matrix Anal. Appl.* **19** (1998), no. 1, 140–153.

[2] M. Dehghan and M. Hajarian, A lower bound for the product of eigenvalues of solutions to matrix equations, *Appl. Math. Lett.* **22** (2009), no. 12, 1786–1788.

[3] M. Dehghan and M. Hajarian, On the generalized bisymmetric and skew-symmetric solutions of the systemof generalized Sylvester matrix equations, *Linear Multilinear Algebra* **59** (2011), no. 11, 1281–1309.

[4] M. Dehghan and M. Hajarian, Analysis of an iterative algorithm to solve the generalized coupled Sylvester matrix equations, *Appl. Math. Modell.* **35** (2011), no. 7, 3285–3300.

[5] M. Dehghan and M. Hajarian, Solving the generalized Sylvester matrix equation $\sum_{i=1}^{p} A_iXB_i + \sum_{j=1}^{q} C_jYD_j = E$ over reflexive and anti-reflexive matrices, *Int. J. Control Autom.* **9** (2011), no. 1, 118–124.

[6] M. Dehghan and M. Hajarian, The general coupled matrix equations over generalized bisymmetric matrices, *Linear Algebra Appl.* **432** (2010), no. 6, 1531–1552.

[7] F. Ding, P. X. Liu and J. Ding, Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle, *Appl. Math. Comput.* **197** (2008), no. 1, 41–50.

[8] F. Ding and T. Chen, Gradient based iterative algorithms for solving a class of matrix equations, *IEEE Trans. Autom. Control* **50** (2005), no. 8, 1216–1221.

[9] F. Ding and T. Chen, Iterative least squares solutions of coupled Sylvester matrix equations, *Systems Control Lett.* **54** (2005), no. 2, 95–107.

[10] F. Ding and T. Chen, On iterative solutions of general coupled matrix equations, *SIAM J. Control Optim.* **44** (2006), no. 6, 2269–2284.

[11] J. Ding, Y. Liu and F. Ding, Iterative solutions to matrix equations of the form $A_iXB_i = F$, *Comput. Math. Appl.* **59** (2010), no. 11, 3500–3507.

[12] J. Ding, F. Ding, X. P. Liu and G. Liu, Hierarchical least squares identification for linear SISO systems with dual-rate sampled-data, *IEEE Trans. Automat. Control* **56** (2011), no. 11, 2677–2683.

[13] F. Ding, Transformations between some special matrices, *Comput. Math. Appl.* **59** (2010), no. 8, 2676–2695.

[14] G. R. Duan, On the solution to the Sylvester matrix equation $AV+BW = EVF$, *IEEE Trans. Automat. Control* **41** (1996), no. 4, 612–614.

[15] B. A. Francis, A Course in H_∞ Control Theory, Lecture Notes in Control and Information Sciences, 88, Springer-Verlag, New York, 1987.

[16] C. Gu, H. Xue, A shift-splitting hierarchical identification method for solving Lyapunov matrix equations, *Linear Algebra Appl.* **430** (2009), no. 5-6, 1517–1530.

[17] G. H. Golub and W. Kahan, Calculating the singular values and pseudoinverse of a matrix, *J. Soc. Indust. Appl. Math. Ser. B Numer. Anal.* **2** (1965) 205–224.

[18] R. A. Horn and C. R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, 1991.

[19] Q. Hu and D. Cheng, The polynomial solution to the Sylvester matrix equation, *Appl. Math. Lett.* **19** (2006), no. 9, 859–864.

[20] G. X. Huang, F. Yin and K. Guo, An iterative method for the skew-symmetric solution and the optimal approximate solution of the matrix equation $AXB = C$, *J. Comput. Appl. Math.* **212** (2008), no. 2, 231–244.

[21] D. Kleinman, On an iterative technique for Riccati equation computations, *IEEE Trans. Automat. Control* **13** (1968) 114–115.

[22] J. LaSalle and S. Lefschetz, Stability of Lyapunov's Direct Method, Academic Press, New York, 1961.

[23] M. L. Liang, C. H. You and L. F. Dai, An efficient algorithm for the generalized centro-symmetric solution of matrix equation $AXB = C$, *Numer. Algorithms* **44** (2007), no. 2, 173–184.

[24] Y. J. Liu, J. Sheng and R. F. Ding, Convergence of stochastic gradient estimation algorithm for multivariable ARX-like systems, *Comput. Math. Appl.* **59** (2010), no. 8, 2615–2627.

[25] Y. J. Liu, Y. S. Xiao and X. L. Zhao, Multi-innovation stochastic gradient algorithm for multiple-input single-output systems using the auxiliary model, *Appl. Math. Comput.* **215** (2009), no. 4, 1477–1483.

[26] B. C. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, *IEEE Trans. Automat. Control* **26** (1981) no. 1, 17–31.

[27] C. C. Paige, Bidiagonalization of matrices and solution of linear equation, *SIAM. J. Numer. Anal.* **11** (1974) 197–209.

[28] C. C. Paige and M. A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software* **8** (1982), no. 1, 43–47.

[29] M. Tombs and I. Postlethwaite, Truncated balanced realization of a stable non-minimal state-space system, *Internat. J. Control* **46** (1987), no. 4, 1319–1330.

[30] M. Wang, X. Cheng and M. Wei, Iterative algorithms for solving the matrix equation $AXB + CX^TD = E$, *Appl. Math. Comput.* **187** (2007), no. 2, 622–629.

[31] Q. W. Wang, Bisymmetric and centrosymmetric solutions to systems of real quaternion matrix equations, *Comput. Math. Appl.* **49** (2005), no. 5-6, 641–650.

[32] Q. W. Wang, J. W. Woude and H. X. Chang, A system of real quaternion matrix equations with applications, *Linear Algebra Appl.* **431** (2009), no. 12, 2291–2303.

[33] Q. W. Wang and C. K. Li, Ranks and the least-norm of the general solution to a system of quaternion matrix equations, *Linear Algebra Appl.* **430** (2009) 1626–1640.

[34] Q. W. Wang and F. Zhang, The reflexive re-nonnegative definite solution to a quaternion matrix equation, *Electron. J. Linear Algebra* **17** (2008) 88–101.

[35] L. Xie, J. Ding and F. Ding, Gradient based iterative solutions for general linear matrix equations, *Comput. Math. Appl.* **58** (2009), no. 7, 1441–1448.

[36] L. Xie, Y. J. Liu and H. Z. Yang, Gradient based and least squares based iterative algorithms for matrix equations $AXB+CX^TD = F$, *Appl. Math. Comput.* **217** (2010), no. 5, 2191–2199.

[37] B. Zhou and G. R. Duan, An explicit solution to the matrix equation $AX-XF = BY$, *Linear Algebra Appl.* **402** (2005) 345–366.

[38] B. Zhou, J. Lam and G. R. Duan, On Smith-type iterative algorithms for the Stein matrix equation, *Appl. Math. Lett.* **22** (2009), no. 7, 1038–1044.

[39] B. Zhou, J. Lam and G. R. Duan, Toward solution of matrix equation $X = Af(X)B + C$, *Linear Algebra Appl.* **435** (2011), no. 6, 1370–1398.

[40] B. Zhou, G. R. Duan and Z. Y. Li, Gradient based iterative algorithm for solving coupled matrix equations, *Systems Control Lett.* **58** (2009), no. 5, 327–333.

(Masoud Hajarian) Department of Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, General Campus, Evin, Tehran 19839, Iran

E-mail address: m_hajarian@sbu.ac.ir and mhajarian@aut.ac.ir