Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 350 (2013) 3328–3341

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

Matrix iterative methods for solving the Sylvester-transpose and periodic Sylvester matrix equations

Masoud Hajarian*

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

Received 14 February 2013; received in revised form 16 July 2013; accepted 16 July 2013
Available online 9 August 2013

Abstract

The problem of solving matrix equations has many applications in control and system theory. This paper is concerned with the iterative solutions of the Sylvester-transpose matrix equation

$$\sum_{i=1}^{k}(A_iXB_i + C_iX^TD_i) = E,$$

and the periodic Sylvester matrix equation

$$\hat{A}_j\hat{X}_j\hat{B}_j + \hat{C}_j\hat{X}_{j+1}\hat{D}_j = \hat{E}_j \quad \text{for } j = 1, 2, \ldots, \lambda.$$

The basic idea is to develop the conjugate gradients squared (CGS) and bi-conjugate gradient stabilized (Bi-CGSTAB) methods for obtaining matrix iterative methods for solving the Sylvester-transpose and periodic Sylvester matrix equations. Numerical test results are given to compare matrix iterative methods with other well-known methods.

1. Introduction

In this paper, first we consider the problem of solving the Sylvester-transpose matrix equation:

$$\sum_{i=1}^{k}(A_iXB_i + C_iX^TD_i) = E, \tag{1.1}$$

*Tel.: +98 2129902924.
E-mail addresses: masoudhajarian@gmail.com, mhajarian@aut.ac.ir, m_hajarian@sbu.ac.ir.

0016-0032/$32.00
http://dx.doi.org/10.1016/j.jfranklin.2013.07.008

where $A_i, B_i, C_i, D_i, E \in \mathbf{R}^{m\times m}$ are known matrices for $i = 1, 2, \ldots, k$ and $X \in \mathbf{R}^{m\times m}$ is the matrix to be determined. This class of matrix equations includes various linear matrix equations such as

$$AXB = C, \tag{1.2}$$

$$AX + XA^T = B, \text{ Lyapunov matrix equation}, \tag{1.3}$$

$$AX + XB = C, \text{ Sylvester matrix equation}, \tag{1.4}$$

$$X + AXB = C, \text{ Stein matrix equation}, \tag{1.5}$$

$$AXB + CXD = E, \text{ generalized Sylvester matrix equation}, \tag{1.6}$$

$$AXB + CX^TD = E, \text{ Sylvester}-\text{transpose matrix equation}. \tag{1.7}$$

Second we consider the periodic Sylvester matrix equation:

$$\hat{A}_j\hat{X}_j\hat{B}_j + \hat{C}_j\hat{X}_{j+1}\hat{D}_j = \hat{E}_j, \tag{1.8}$$

for $j = 1, 2, \ldots$, where the coefficient matrices $\hat{A}_j, \hat{C}_j, \hat{B}_j, \hat{D}_j \in \mathbf{R}^{m\times m}$ and the solutions $\hat{X}_j \in \mathbf{R}^{m\times m}$ are periodic with period λ, i.e., $\hat{A}_{j+\lambda} = \hat{A}_j$, $\hat{B}_{j+\lambda} = \hat{B}_j$, $\hat{C}_{j+\lambda} = \hat{C}_j$, $\hat{D}_{j+\lambda} = \hat{D}_j$, $\hat{E}_{j+\lambda} = \hat{E}_j$ and $\hat{X}_{j+\lambda} = \hat{X}_j$. The linear matrix equations play a fundamental role in a variety of fields of control theory and applied mathematics [18,20,22,29–31,36]. Therefore some authors have established the problem for finding analytical and numerical solutions of matrix [2–5,12,15,27,33,34,37]. In [11,16], the gradient-iterative (GI) algorithms were proposed for solving (generalized) Sylvester matrix equations by using the Jacobi method and hierarchical identification principle [13,14]. Piao et al. [25] presented some necessary and sufficient condition for the existence of the solution and the expressions of the matrix equation:

$$AX + X^TC = B, \tag{1.9}$$

by using Moore–Penrose generalized inverse. In [32], two iterative algorithms were proposed to solve the Sylvester-transpose matrix equation (1.7) when this matrix equation is consistent and inconsistent, respectively. Zhou et al. [35] analyzed the computational complexity of the Smith iteration and its variations for solving the Stein matrix equation (1.5). In [21,23,24], the matrix LSQR iterative methods were proposed to solve the constrained solutions of the matrix equation (1.2) and the generalized coupled Sylvester matrix equations. Recently some iterative algorithms based on the conjugate gradient method (CG) were introduced for solving Sylvester and Lyapunov matrix [6–8,10]. Kressner introduced new variants of the squared Smith iteration and Krylov subspace based methods for the approximate solution of discrete-time periodic Lyapunov equations [19]. Andersson et al. extended and applied the recursive blocking technique to solving periodic Sylvester matrix equations [1].

The CGS and Bi-CGSTAB methods are powerful algorithms for solving the unsymmetric system of linear equations

$$Ax = b, \tag{1.10}$$

wheree $A \in \mathbf{R}^{m\times m}$ and $x, b \in \mathbf{R}^m$. In this paper we directly generalize the CGS and Bi-CGSTAB methods to obtain four matrix iterative methods for solving the Sylvester-transpose matrix equation (1.1) and the periodic Sylvester matrix equation (1.8).

The rest of this paper is organized as follows. In Section 2, after we review succinctly the CGS and Bi-CGSTAB methods, we extend these methods for solving Eqs. (1.1) and (1.8) by means of the Kronecker product and vectorization operator. We present some numerical examples to illustrate the efficiency of the matrix algorithms in Section 3. In Section 4, conclusions will be drawn.

Throughout this paper, we use A^T and $\mathrm{tr}(A)$ which denote the transpose and the trace of A, respectively. We define the inner product $\langle A, B\rangle = \mathrm{tr}(B^T A)$ for all $A, B \in \mathbf{R}^{m\times n}$, then $\mathbf{R}^{m\times n}$ is a Hilbert inner product space and the norm of a matrix generated by this inner product is the matrix Frobenius norm $\|\cdot\|$. The symbol $\otimes$ is used to denote Kronecker product. For a matrix $A \in \mathbf{R}^{m\times n}$, $\mathrm{vec}(A)$ is defined as $\mathrm{vec}(A) = (a_1^T\ a_2^T\ \ldots\ a_n^T)^T$ where a_i is the i-th column of the matrix A. The set of all real polynomials of degree at most n is denoted by $\mathcal{P}_n$.

2. Matrix iterative algorithms

In this section we begin with a brief explanation and survey of CGS and Bi-CGSTAB methods. Then we propose matrix algorithms based on the CGS and Bi-CGSTAB methods to solve Eqs. (1.1) and (1.8).

The bi-conjugate gradient (Bi-CG) method is a powerful Krylov subspace method for the solution of large sparse unsymmetric linear systems (1.10). The Bi-CG algorithm is susceptible to possible breakdowns and numerical instabilities. A number of hybrid Bi-CG methods such as CGS and Bi-CGSTAB have been presented to improve the convergence of Bi-CG and to avoid multiplication by the A^T. The residuals of CGS and Bi-CGSTAB are respectively expressed as the product of the residual of Bi-CG and a stabilization polynomial by

$$r(n) = (\phi(n)(A))^2 r(0), \text{ the residual of CGS,}$$

and

$$r(n) = \tau(n)(A)\phi(n)(A)r(0) = (I-\omega_1 A)(I-\omega_2 A)\ldots(I-\omega(n)A)\phi(n)(A)r(0),$$
$$\text{the residual of Bi-CGSTAB,}$$

where $r(0) = b-Ax(0)$, $\phi(n) \in \mathcal{P}(n)$, $\phi(n)(0) = 1$, and the ω_i's are chosen to locally minimize the residual by a steepest descent method, for more detail see [17,26,28]. The CGS and Bi-CGSTAB algorithms can be summarized in Tables 1 and 2, respectively. In exact arithmetic, the CGS and Bi-CGSTAB algorithms terminate after a finite number, say n^*, of iterations. Usually, $x_{n^*} = A^{-1}b$ is the solution of the linear systems (1.10).

Table 1
The CGS algorithm.

Choose $x(0) \in \mathbf{R}^m$;
$p(0) = u(0) = r(0) = b-Ax(0)$, $v(0) = Ap(0)$;
Pick an arbitrary vector $\tilde{r}(0)$ (for example $\tilde{r}(0) = r(0)$);
For $n = 1, 2, \ldots$ until convergence, do:
$\sigma(n-1) = \langle v(n-1), \tilde{r}(0)\rangle$, $\alpha(n-1) = \rho(n-1)/\sigma(n-1)$;
$q(n) = u(n-1)-\alpha(n-1)v(n-1)$;
$x(n) = x(n-1) + \alpha(n-1)(u(n-1) + q(n))$;
$r(n) = r(n-1)-\alpha(n-1)A(u(n-1) + q(n))$;
If $x(n)$ has converged: stop;
$\rho(n) = \langle r(n), \tilde{r}(0)\rangle$, $\beta(n) = \rho(n)/\rho(n-1)$;
$u(n) = r(n) + \beta(n)q(n)$;
$p(n) = u(n) + \beta(n)(q(n) + \beta(n)p(n-1))$;
$v(n) = Ap(n)$.

Table 2
The Bi-CGSTAB algorithm.

Choose $x(0) \in \mathbf{R}^m$ and compute $r(0) = b - Ax(0)$;
Pick an arbitrary vector $\tilde{r}(0)$ (for example $\tilde{r}(0) = r(0)$);
$v(0) = p(0) = 0$; $\rho(0) = \alpha(1) = \omega(0) = 1$;
For $n = 1, 2, \ldots,$ until convergence
$\rho(n) = \langle r(n-1), \tilde{r}(0) \rangle$; $\beta(n) = \left(\frac{\rho(n)}{\rho(n-1)} \right) \left(\frac{\alpha(n)}{\omega(n-1)} \right)$;
$p(n) = r(n-1) + \beta(n)(p(n-1) - \omega(n-1)v(n-1))$;
$v(n) = Ap(n)$;
$\sigma(n) = \langle v(n), \tilde{r}(0) \rangle$; $\alpha(n) = \frac{\rho(n)}{\sigma(n)}$;
$s(n) = r(n-1) - \alpha(n)v(n)$; $t(n) = As(n)$;
$\omega(n) = \frac{\langle s(n), t(n) \rangle}{\langle t(n), t(n) \rangle}$;
$r(n) = s(n) - \omega(n)t(n)$;
$x(n) = x(n-1) + \alpha(n)p(n) + \omega(n)s(n)$.

2.1. Matrix iterative algorithms for solving Eq. (1.1)

For solving the Sylvester-transpose matrix equation (1.1) by the CGS and Bi-CGSTAB algorithms, we need to transform (1.1) into linear systems (1.10). By applying the Kronecker product and vectorization operator, the Sylvester-transpose matrix equation (1.1) can be transformed to the linear systems (1.10) with parameters:

$$A = \sum_{i=1}^{k} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P) \in \mathbf{R}^{m^2 \times m^2},$$
$$x = \mathrm{vec}(X) \in \mathbf{R}^{m^2} \quad \text{and} \quad b = \mathrm{vec}(E) \in \mathbf{R}^{m^2}, \tag{2.1}$$

where $P \in \mathbf{R}^{m^2 \times m^2}$ is a unitary matrix [37]. The dimension of the associate matrix A is high when m is large. Such a dimensional problem leads to computational difficulty in that excessive computer memory is required for computation of iterative methods like the CGS and Bi-CGSTAB algorithms. In order to overcome this problem, we propose the matrix forms of the CGS and Bi-CGSTAB algorithms for solving Eq. (1.1).

2.1.1. Matrix form of CGS algorithm for solving Eq. (1.1)

By substituting the parameters (2.1) into the CGS algorithm, we have

$$p(0) = u(0) = r(0) = b - Ax(0) = \mathrm{vec}(E) - \sum_{i=1}^{k} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)x(0), \tag{2.2}$$

$$v(n) = Ap(n) = \sum_{i=1}^{k} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)p(n), \tag{2.3}$$

and

$$r(n) = r(n-1) - \alpha(n-1)A(u(n-1) + q(n)) = r(n-1) - \alpha(n-1) \sum_{i=1}^{k} (B_i^T \otimes A_i$$
$$+ (D_i^T \otimes C_i)P)(u(n-1) + q(n)). \tag{2.4}$$

By considering the CGS algorithm and the above equations, we define the following matrices:

$$p(n)=\mathrm{vec}(P(n)),\ \ u(n)=\mathrm{vec}(U(n)),\ \ r(n)=\mathrm{vec}(R(n)),\ \ v(n)=\mathrm{vec}(V(n)), \tag{2.5}$$

$$q(n)=\mathrm{vec}(Q(n)),\ \ \tilde{r}(0)=\mathrm{vec}(\tilde{R}(0)),\quad \text{and}\quad x(n)=\mathrm{vec}(X(n)), \tag{2.6}$$

where $P(n), U(n), R(n), V(n), Q(n), \tilde{R}(0), X(n)\in \mathbf{R}^{m\times m}$ for $n=0,1,2,\ldots$. By these definitions, we can obtain

$$\mathrm{vec}(P(0))=\mathrm{vec}(U(0))=\mathrm{vec}(R(0))=\mathrm{vec}(E)-\sum_{i=1}^{k}(B_i^T\otimes A_i +(D_i^T\otimes C_i)P)\mathrm{vec}(X(0)) \tag{2.7}$$

$$=\mathrm{vec}\left(E-\sum_{i=1}^{k}(A_iX(0)B_i+C_iX(0)^TD_i)\right), \tag{2.8}$$

$$\mathrm{vec}(V(n))=\mathrm{vec}\left(\sum_{i=1}^{k}(A_iP(n)B_i+C_iP(n)^TD_i)\right), \tag{2.9}$$

$$\mathrm{vec}(R(n))=\mathrm{vec}(R(n-1))-\alpha(n-1)\mathrm{vec}\left(\sum_{i=1}^{k}(A_i(U(n-1)+Q(n))B_i+C_i(U(n-1)+Q(n))^TD_i)\right), \tag{2.10}$$

$$\sigma(n-1)=\langle \mathrm{vec}(V(n-1)),\mathrm{vec}(\tilde{R}(0))\rangle=\langle V(n-1),\tilde{R}(0)\rangle, \tag{2.11}$$

and

$$\rho(n)=\langle \mathrm{vec}(R(n)),\mathrm{vec}(\tilde{R}(0))\rangle=\langle R(n),\tilde{R}(0)\rangle. \tag{2.12}$$

From the above results, the matrix form of CGS algorithm can be presented in Table 3.

Table 3
The matrix form of CGS algorithm for solving Eq. (1.1).

Choose $X(0)\in\mathbf{R}^{m\times m}$;
$P(0)=U(0)=R(0)=E-\sum_{i=1}^{k}(A_iX(0)B_i+C_iX(0)^TD_i)$, $V(0)=\sum_{i=1}^{k}(A_iP(0)B_i+C_iP(0)^TD_i)$;
Pick an arbitrary matrix $\tilde{R}(0)$ (for example $\tilde{R}(0)=R(0)$);
For $n=1,2,\ldots$until convergence, do:
$\sigma(n-1)=\langle V(n-1),\tilde{R}(0)\rangle$, $\alpha(n-1)=\rho(n-1)/\sigma(n-1)$;
$Q(n)=U(n-1)-\alpha(n-1)V(n-1)$;
$X(n)=X(n-1)+\alpha(n-1)(U(n-1)+Q(n))$;
$R(n)=R(n-1)-\alpha(n-1)\sum_{i=1}^{k}(A_i(U(n-1)+Q(n))B_i+C_i(U(n-1)+Q(n))^TD_i)$;
If $X(n)$ has converged: stop;
$\rho(n)=\langle R(n),\tilde{R}(0)\rangle$, $\beta(n)=\rho(n)/\rho(n-1)$;
$U(n)=R(n)+\beta(n)Q(n)$;
$P(n)=U(n)+\beta(n)(Q(n)+\beta(n)P(n-1))$;
$V(n)=\sum_{i=1}^{k}(A_iP(n)B_i+C_iP(n)^TD_i)$.

2.1.2. Matrix form of Bi-CGSTAB algorithm for solving Eq. (1.1)

If we apply the Bi-CGSTAB algorithm for the linear systems (1.10) with parameters (2.1), we get

$$r(0) = \mathrm{vec}(E) - \sum_{i=1}^{k} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)x(0), \tag{2.13}$$

$$v(n) = \sum_{i=1}^{k} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)p(n), \tag{2.14}$$

and

$$t(n) = \sum_{i=1}^{k} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)s(n). \tag{2.15}$$

We define

$$p(n) = \mathrm{vec}(P(n)), \;\; s(n) = \mathrm{vec}(S(n)), \;\; r(n) = \mathrm{vec}(R(n)), \;\; v(n) = \mathrm{vec}(V(n)), \tag{2.16}$$

$$t(n) = \mathrm{vec}(T(n)), \;\; \tilde{r}(0) = \mathrm{vec}(\tilde{R}(0)), \quad \text{and} \quad x(n) = \mathrm{vec}(X(n)), \tag{2.17}$$

where $P(n), R(n), V(n), S(n), T(n), \tilde{R}(0), X(n) \in \mathbf{R}^{m\times m}$ for $n = 0, 1, 2, \ldots$. By using the definitions, we have

$$\mathrm{vec}(R(0)) = \mathrm{vec}\left(E - \sum_{i=1}^{k} (A_i X(0) B_i + C_i X(0)^T D_i) \right), \tag{2.18}$$

$$\mathrm{vec}(V(n)) = \mathrm{vec}\left(\sum_{i=1}^{k} (A_i P(n) B_i + C_i P(n)^T D_i) \right), \tag{2.19}$$

$$\mathrm{vec}(T(n)) = \mathrm{vec}\left(\sum_{i=1}^{k} (A_i S(n) B_i + C_i S(n)^T D_i) \right), \tag{2.20}$$

$$\rho(n) = \langle R(n-1), \tilde{R}(0) \rangle, \tag{2.21}$$

$$\sigma(n) = \langle V(n), \tilde{R}(0) \rangle, \tag{2.22}$$

and

$$\omega(n) = \frac{\langle S(n), T(n) \rangle}{\langle T(n), T(n) \rangle}. \tag{2.23}$$

From the above discussion, we can present the matrix form of Bi-CGSTAB algorithm in Table 4. The stopping criteria on the matrix algorithms can be used as

$$\| E - \sum_{i=1}^{k} (A_i X(n) B_i + C_i X(n)^T D_i) \| \leq \varepsilon.$$

where $\varepsilon > 0$ is a small tolerance.

Table 4
The matrix form of Bi-CGSTAB algorithm for solving Eq. (1.1).

Choose $X(0)\in\mathbf{R}^{m\times m}$ and compute $R(0)=E-\sum_{i=1}^{k}(A_iX(0)B_i+C_iX(0)^TD_i)$;
Pick an arbitrary matrix $\tilde{R}(0)$ (for example $\tilde{R}(0)=R(0)$);
$V(0)=P(0)=0$; $\rho(0)=\alpha(1)=\omega(0)=1$;
For $n=1,2,\ldots,$ until convergence
$\rho(n)=\langle R(n-1),\tilde{R}(0)\rangle$; $\beta(n)=\left(\dfrac{\rho(n)}{\rho(n-1)}\right)\left(\dfrac{\alpha(n)}{\omega(n-1)}\right)$;
$P(n)=R(n-1)+\beta(n)(P(n-1)-\omega(n-1)V(n-1))$;
$V(n)=\sum_{i=1}^{k}(A_iP(n)B_i+C_iP(n)^TD_i)$;
$\sigma(n)=\langle V(n),\tilde{R}(0)\rangle$; $\alpha(n)=\dfrac{\rho(n)}{\sigma(n)}$;
$S(n)=R(n-1)-\alpha(n)V(n)$; $T(n)=\sum_{i=1}^{k}(A_iS(n)B_i+C_iS(n)^TD_i)$;
$\omega(n)=\dfrac{\langle S(n),T(n)\rangle}{\langle T(n),T(n)\rangle}$;
$R(n)=S(n)-\omega(n)T(n)$;
$X(n)=X(n-1)+\alpha(n)P(n)+\omega(n)S(n)$.

2.2. *Matrix iterative algorithms for solving Eq. (1.8)*

We can easily show that the periodic Sylvester matrix equation (1.8) is equivalent to the following Sylvester matrix equation:

$$\mathcal{A}\mathcal{X}\mathcal{B}+\mathcal{C}\mathcal{X}\mathcal{D}=\mathcal{E}, \tag{2.24}$$

where

$$\mathcal{A}=\begin{bmatrix}0&\cdots&0&\hat{A}_1\\ \hat{A}_2&&&0\\ &\ddots&&\vdots\\ 0&&\hat{A}_\lambda&0\end{bmatrix},\quad \mathcal{B}=\begin{bmatrix}0&\hat{B}_2&&0\\ \vdots&&\ddots&\\ 0&&&\hat{B}_\lambda\\ \hat{B}_1&0&\cdots&0\end{bmatrix},\quad \mathcal{C}=\mathrm{diag}\left[\hat{C}_1,\hat{C}_2,\ldots,\hat{C}_\lambda\right],$$

$$\mathcal{D}=\mathrm{diag}\left[\hat{D}_1,\hat{D}_2,\ldots,\hat{D}_\lambda\right],\quad \mathcal{E}=\mathrm{diag}\left[\hat{E}_1,\hat{E}_2,\ldots,\hat{E}_\lambda\right],\quad \mathcal{X}=\mathrm{diag}\left[\hat{X}_2,\hat{X}_3,\ldots,\hat{X}_\lambda,\hat{X}_1\right].$$

Obviously the Sylvester matrix equation (2.24) can be solved by applying the matrix forms of CGS and Bi-CGSTAB algorithms presented in Tables 3 and 4. But the size of coefficient matrices $\mathcal{A},\mathcal{B},\mathcal{C},\mathcal{D},\mathcal{E}$, and the solution matrix $\mathcal{X}$ is very large when m is large. Here by considering Eqs. (1.8), (2.24) and the matrix forms of CGS and Bi-CGSTAB algorithms, we propose two iterative algorithms for solving Eq. (1.8) in Tables 5 and 6, respectively.

The stopping criteria on these two algorithms can be used as

$$\sqrt{\sum_{j=1}^{\lambda}\|\hat{E}_j-\hat{A}_j\hat{X}_j(n)\hat{B}_j-\hat{C}_j\hat{X}_{j+1}(n)\hat{D}_j\|^2}\le\varepsilon,$$

where $\varepsilon>0$ is a small tolerance.

3. Numerical examples

In this section, the matrix algorithms are employed to solve some linear matrix equations and compared with the extended CG method (CG_M) [9,10,32] and the matrix LSQR iterative method (LSQR_M) [21,23,24]. All computations were done using MATLAB.

Example 1. First, we consider the matrix equation (1.2) where

$$A = \mathrm{triu}(\mathrm{rand}(150,150),1) + \mathrm{diag}(1.75 + \mathrm{diag}(\mathrm{rand}(150))) \in \mathbf{R}^{150\times 150},$$
$$B = \mathrm{tril}(\mathrm{rand}(150,150),1) + \mathrm{diag}(2 + \mathrm{diag}(\mathrm{rand}(150))) \in \mathbf{R}^{150\times 150}, \quad \text{and}$$
$$C = \mathrm{rand}(150,150) \in \mathbf{R}^{150\times 150}.$$

By applying the above mentioned methods, we compute the sequence $\{X(n)\}$ with $X(0)=0$. The numerical results are given in Fig. 1 where $r_n = \log_{10}\| C - AX(n)B \|$.

Example 2. In this example, the Sylvester matrix equation (1.4) is considered with parameters

$$A = B = M + 2rN + \frac{100}{(m+1)^2} I \in \mathbf{R}^{m\times m}, \quad C = \mathrm{rand}(m) \in \mathbf{R}^{m\times m},$$

where

$$M = \mathrm{tridiag}(-1,2,-1) \in \mathbf{R}^{m\times m} \quad \text{and} \quad N = \mathrm{tridiag}(0.5,0,-0.5) \in \mathbf{R}^{m\times m}.$$

When $m=200$ and $r=0.01$, by using the above mentioned methods, we calculate the sequence $\{X(n)\}$ with $X(0)=0$. The plot of convergence behavior is shown in Fig. 2 where $r_n = \log_{10}\| C - AX(n) - X(n)B \|$.

Example 3. Now by the mentioned methods we solve the Stein matrix equation (1.4) with the following matrices:

$$A = \mathrm{tril}(\mathrm{rand}(150,150),1) + \mathrm{diag}(2 + \mathrm{diag}(\mathrm{rand}(150))) \in \mathbf{R}^{150\times 150},$$

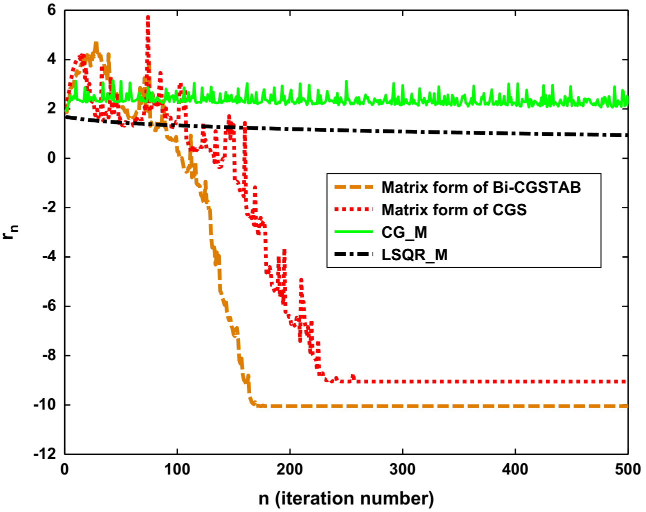

Fig. 1. Comparison of residuals for Example 1.

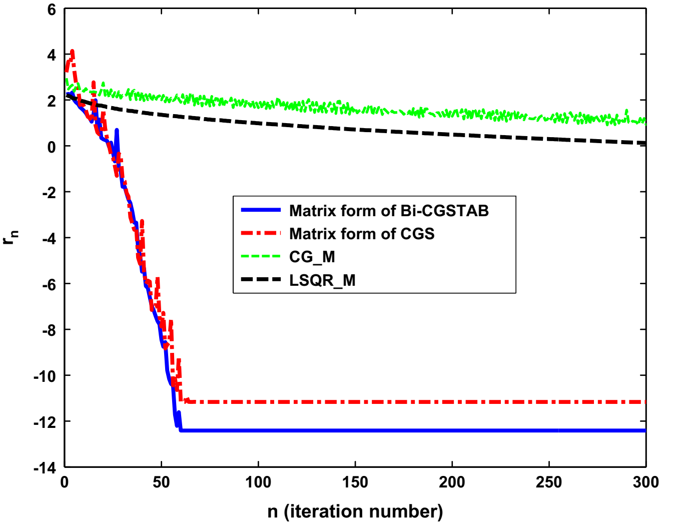

Fig. 2. Comparison of residuals for Example 2.

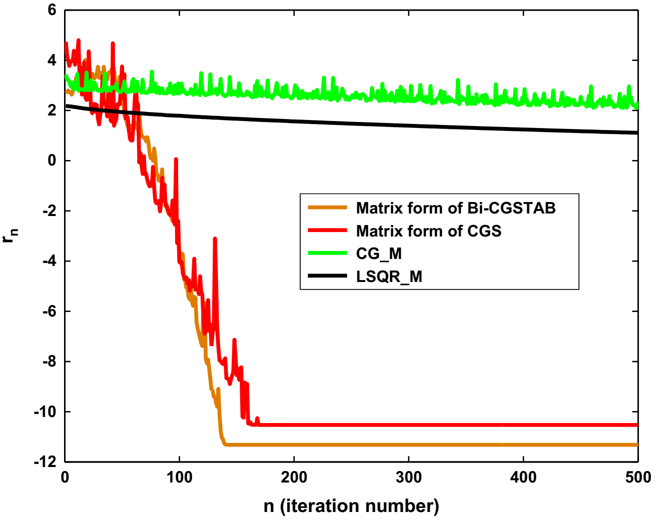

Fig. 3. Comparison of residuals for Example 3.

$B = \text{tril}(\text{rand}(150, 150), 1) + \text{diag}(2 + \text{diag}(\text{rand}(150))) \in \mathbf{R}^{150\times 150}$, and
$C = \text{rand}(150) \in \mathbf{R}^{150\times 150}$.

In Fig. 3, the comparison of residuals are plotted where $r_n = \log_{10} \| C - X(n) - AX(n)B \|$.

Example 4. In this example we study the Sylvester-transpose matrix equation (1.7) with

$$A = \text{triu}(\text{rand}(50, 50), 1) + \text{diag}(3 + \text{diag}(\text{rand}(50))) \in \mathbf{R}^{50\times 50},$$
$$B = \text{tril}(\text{rand}(50, 50), 1) + \text{diag}(8 + \text{diag}(\text{rand}(50))) \in \mathbf{R}^{50\times 50},$$
$$C = \text{triu}(\text{rand}(50, 50), 1) + \text{diag}(3 + \text{diag}(\text{rand}(50))) \in \mathbf{R}^{50\times 50},$$

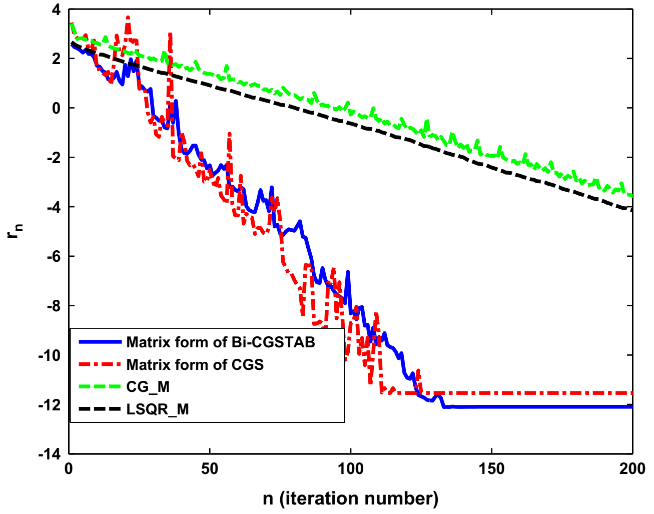

Fig. 4. Comparison of residuals for Example 4.

$D = \mathrm{triu}(\mathrm{rand})50, 50), 1) + \mathrm{diag}(1 + \mathrm{diag}(\mathrm{rand}(50))) \in \mathbf{R}^{50\times 50}$, and
$E = 10 \times \mathrm{rand}(50) \in \mathbf{R}^{50\times 50}$.

The numerical results obtained from mentioned methods with $X(0) = 0$ are depicted in Fig. 4 where $r_n = \log_{10} \| E - AX(n)B - CX^T(n)D \|$.

From the above examples, we can see that the proposed matrix algorithms have faster convergence rate and higher accuracy than other methods.

Example 5. As the final example, we consider the periodic Sylvester matrix equation:

$$\hat{X}_j + \hat{C}_j \hat{X}_{j+1} \hat{D}_j = \hat{E}_j \quad \text{for } j = 1, 2,$$

with parameters

$$\hat{C}_1 = \mathrm{tril}(\mathrm{rand}(20, 20), 1) + \mathrm{diag}(2 + \mathrm{diag}(\mathrm{rand}(20))) \in \mathbf{R}^{20\times 20},$$
$$\hat{D}_1 = \mathrm{triu}(\mathrm{rand}(20, 20), 1) + \mathrm{diag}(1.75 + \mathrm{diag}(\mathrm{rand}(20))) \in \mathbf{R}^{20\times 20},$$
$$\hat{C}_2 = \mathrm{triu}(\mathrm{rand}(20, 20), 1) + \mathrm{diag}(1.75 + \mathrm{diag}(\mathrm{rand}(20))) \in \mathbf{R}^{20\times 20},$$
$$\hat{D}_2 = \mathrm{tril}(\mathrm{rand}(20, 20), 20) + \mathrm{diag}(2 + \mathrm{diag}(\mathrm{rand}(20))) \in \mathbf{R}^{20\times 20},$$
$$\hat{E}_1 = \hat{E}_2 = \mathrm{rand}(20, 20) \in \mathbf{R}^{20\times 20}.$$

By applying the matrix forms of CGS and Bi-CGSTAB algorithms given Tables 5 and 6 with $\hat{X}_1(0) = \hat{X}_2(0)$, we compute the sequences $\{\hat{X}_1(n)\}$ and $\{\hat{X}_2(n)\}$. The numerical results are shown in Fig. 5 where

$$r_{1,n} = \log_{10} \| \hat{E}_1 - \hat{X}_1(n) - \hat{C}_1 \hat{X}_2(n) \hat{D}_1 \| \quad \text{and} \quad r_{2,n} = \log_{10} \| \hat{E}_2 - \hat{X}_2(n) - \hat{C}_2 \hat{X}_1(n) \hat{D}_2 \|. \tag{3.1}$$

Obviously $r_{1,n}$ and $r_{2,n}$ decrease, and converge to zero as n increases.

Table 5
The matrix form of CGS algorithm for solving Eq. (1.8).

Choose $\hat{X}_j(0) \in \mathbf{R}^{m\times m}$ for $j=1,2,\ldots,\lambda$ and set $\hat{X}_{\lambda+1}(0)=\hat{X}_1(0)$;
$P_j(0)=U_j(0)=R_j(0)=\hat{E}_j-(\hat{A}_j\hat{X}_j(0)\hat{B}_j+\hat{C}_j\hat{X}_{j+1}(0)\hat{D}_j)$ for $j=1,2,\ldots,\lambda$;
Set $P_{\lambda+1}(0)=P_1(0)$ and $U_{\lambda+1}(0)=U_1(0)$;
$V_j(0)=\hat{A}_jP_j(0)\hat{B}_j+\hat{C}_jP_{j+1}(0)\hat{D}_j$ for $j=1,2,\ldots,\lambda$;
Pick arbitrary matrices $\tilde{R}_j(0)$ (for example $\tilde{R}_j(0)=R_j(0)$) for $j=1,2,\ldots,\lambda$;
For $n=1,2,\ldots$ until convergence, do:
$\sigma(n-1)=\sum_{j=1}^{\lambda}\langle V_j(n-1),\tilde{R}_j(0)\rangle$, $\alpha(n-1)=\rho(n-1)/\sigma(n-1)$;
$Q_j(n)=U_j(n-1)-\alpha(n-1)V_j(n-1)$ for $j=1,2,\ldots,\lambda$;
Set $Q_{\lambda+1}(n)=Q_1(n)$;
$\hat{X}_j(n)=\hat{X}_j(n-1)+\alpha(n-1)(U_j(n-1)+Q_j(n))$ for $j=1,2,\ldots,\lambda$;
Set $\hat{X}_{\lambda+1}(n)=\hat{X}_1(n)$;
$R_j(n)=R_j(n-1)-\alpha(n-1)(\hat{A}_j(U_j(n-1)+Q_j(n))\hat{B}_j+\hat{C}_j(U_{j+1}(n-1)+Q_{j+1}(n))\hat{D}_j)$ for $j=1,2,\ldots,\lambda$;
If $(\hat{X}_1(n),\hat{X}_2(n),\ldots,\hat{X}_\lambda(n))$ has converged: stop;
$\rho(n)=\sum_{j=1}^{\lambda}\langle R_j(n),\tilde{R}_j(0)\rangle$, $\beta(n)=\rho(n)/\rho(n-1)$;
$U_j(n)=R_j(n)+\beta(n)Q_j(n)$ for $j=1,2,\ldots,\lambda$;
$P_j(n)=U_j(n)+\beta(n)(Q_j(n)+\beta(n)P_j(n-1))$ for $j=1,2,\ldots,\lambda$;
Set $P_{\lambda+1}(n)=P_1(n)$ and $U_{\lambda+1}(n)=U_1(n)$;
$V_j(n)=\hat{A}_jP_j(n)\hat{B}_j+\hat{C}_jP_{j+1}(n)\hat{D}_j$ for $j=1,2,\ldots,\lambda$.

Table 6
The matrix form of Bi-CGSTAB algorithm for solving Eq. (1.8).

Choose $\hat{X}_j(0) \in \mathbf{R}^{m\times m}$ and compute $R_j(0)=\hat{E}_j-(\hat{A}_j\hat{X}_j(0)\hat{B}_j+\hat{C}_j\hat{X}_{j+1}(0)\hat{D}_j)$ for $j=1,2,\ldots,\lambda$;
Pick arbitrary matrices $\tilde{R}_j(0)$ (for example $\tilde{R}_j(0)=R_j(0)$) for $j=1,2,\ldots,\lambda$;
$V_j(0)=P_j(0)=0$, $\rho(0)=\alpha(1)=\omega(0)=1$ for $j=1,2,\ldots,\lambda$;
For $n=1,2,\ldots,$ until convergence
$\rho(n)=\sum_{j=1}^{\lambda}\langle R_j(n-1),\tilde{R}_j(0)\rangle$; $\beta(n)=\left(\frac{\rho(n)}{\rho(n-1)}\right)\left(\frac{\alpha(n)}{\omega(n-1)}\right)$;
$P_j(n)=R_j(n-1)+\beta(n)(P_j(n-1)-\omega(n-1)V_j(n-1))$ for $j=1,2,\ldots,\lambda$;
Set $P_{\lambda+1}(n)=P_1(n)$;
$V_j(n)=\hat{A}_jP_j(n)\hat{B}_j+\hat{C}_jP_{j+1}(n)\hat{D}_j$ for $j=1,2,\ldots,\lambda$;
$\sigma(n)=\sum_{j=1}^{\lambda}\langle V_j(n),\tilde{R}_j(0)\rangle$; $\alpha(n)=\frac{\rho(n)}{\sigma(n)}$;
$S_j(n)=R_j(n-1)-\alpha(n)V_j(n)$ for $j=1,2,\ldots,\lambda$;
Set $S_{\lambda+1}(n)=S_1(n)$;
$T_j(n)=\hat{A}_jS_j(n)\hat{B}_j+\hat{C}_jS_{j+1}(n)\hat{D}_j$ for $j=1,2,\ldots,\lambda$;
$\omega(n)=\frac{\sum_{j=1}^{\lambda}\langle S_j(n),T_j(n)\rangle}{\sum_{j=1}^{\lambda}\langle T_j(n),T_j(n)\rangle}$;
$R_j(n)=S_j(n)-\omega(n)T_j(n)$ for $j=1,2,\ldots,\lambda$;
$\hat{X}_j(n)=\hat{X}_j(n-1)+\alpha(n)P_j(n)+\omega(n)S_j(n)$ for $j=1,2,\ldots,\lambda$;
Set $\hat{X}_{\lambda+1}(n)=\hat{X}_1(n)$.

4. Conclusions

In this paper we have introduced effective matrix methods for solving the Sylvester-transpose matrix equation (1.1) and the periodic Sylvester matrix equation (1.8). The introduced matrix

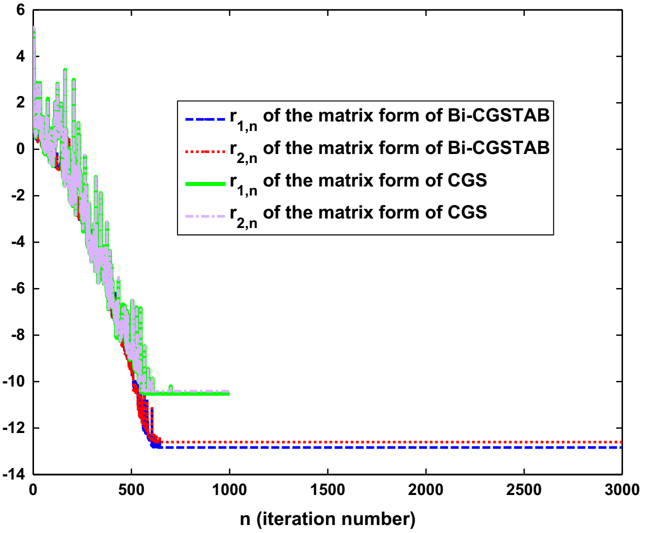

Fig. 5. The residuals for Example 5.

methods are based on the CGS and Bi-CGSTAB techniques. Finally, the convergence and performance of the matrix methods were illustrated and compared with some existing methods on simulated examples.

Acknowledgments

The author is very much indebted to an anonymous referee for his/her valuable comments and careful reading of the manuscript.

References

[1] P. Andersson, R. Granat, I. Jonsson, B. Kagstrom, Parallel algorithms for triangular periodic Sylvester-type matrix equations, Euro-Par 2008 – Parallel Processing (2008) 780–789.

[2] P. Benner, M.S. Hossain, Structure preserving iterative solution of periodic projected Lyapunov equations, in: Proceedings of MATHMOD Conference, Vienna, 2012.

[3] P. Benner, M.S. Hossain, T. Stykel, Low rank iterative methods of periodic projected Lyapunov equations and their application in model reduction of periodic descriptor systems, Chemnitz Scientific Computing Preprints 11–01 (2011).

[4] P. Benner, M.S. Hossain, T. Stykel, Model reduction of periodic descriptor systems using balanced truncation, in: P. Benner, M. Hinze, J. Ter Maten (Eds.), Model Reduction in Circuit Simulation: Lecture Notes in Electrical Engineering, vol. 74, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 193–206.

[5] R. Byers, N. Rhee, Cyclic Schur and Hessenberg Schur Numerical Methods for Solving Periodic Lyapunov and Sylvester Equations, Technical Report, Dept. of Mathematics, Univ. of Missouri at Kansas City, 1995.

[6] M. Dehghan, M. Hajarian, On the generalized bisymmetric and skew-symmetric solutions of the system of generalized Sylvester matrix equations, Linear and Multilinear Algebra 59 (11) (2011) 1281–1309.

[7] M. Dehghan, M. Hajarian, The general coupled matrix equations over generalized bisymmetric matrices, Linear Algebra and its Applications 432 (6) (2010) 1531–1552.

[8] M. Dehghan, M. Hajarian, Efficient iterative method for solving the second-order Sylvester matrix equation $EVF^2-AVF-CV=BW$, IET Control Theory and Applications 3 (10) (2009) 1401–1408.

[9] M. Hajarian, M. Dehghan, The generalized centro-symmetric and least squares generalized centro-symmetric solutions of the matrix equation $AYB+CY^TD=E$, Mathematical Methods in the Applied Sciences 34 (13) (2011) 1562–1579.

[10] M. Dehghan, M. Hajarian, Analysis of an iterative algorithm to solve the generalized coupled Sylvester matrix equations, Applied Mathematical Modelling 35 (7) (2011) 3285–3300.

[11] F. Ding, T. Chen, Gradient based iterative algorithms for solving a class of matrix equations, IEEE Transactions on Automatic Control 50 (8) (2005) 1216–1221.

[12] F. Ding, T. Chen, Iterative least squares solutions of coupled Sylvester matrix equations, Systems and Control Letters 54 (2) (2005) 95–107.

[13] F. Ding, T. Chen, Hierarchical gradient-based identification of multivariable discrete-time systems, Automatica 41 (2) (2005) 315–325.

[14] F. Ding, T. Chen, Hierarchical least squares identification methods for multivariable systems, IEEE Transactions on Automatic Control 50 (3) (2005) 397–402.

[15] F. Ding, T. Chen, On iterative solutions of general coupled matrix equations, SIAM Journal on Control and Optimization 44 (6) (2006) 2269–2284.

[16] F. Ding, P.X. Liu, J. Ding, Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle, Applied Mathematics and Computation 197 (1) (2008) 41–50.

[17] T.L. Doug, Conjugate Gradient-Type Product Methods for Solving Nonsymmetric Linear Systems, Ph.D. Thesis, 1994.

[18] T. Hu, Z. Lin, J. Lam, Unified gradient approach to performance optimization under a pole assignment constrain, Journal of Optimization Theory and Applications 121 (2) (2004) 361–383.

[19] D. Kressner, Large periodic Lyapunov equations: algorithms and applications, in: Proceedings of ECC03, Cambridge, UK, 2003.

[20] J. Lam, W. Yan, T. Hu, Pole assignment with eigenvalue and stability robustness, International Journal of Control 72 (13) (1999) 1165–1174.

[21] S.K. Li, T.Z. Huang, LSQR iterative method for generalized coupled Sylvester matrix equations, Applied Mathematical Modelling 36 (8) (2012) 3545–3554.

[22] T. Penzl, A cyclic low-rank Smith method for large sparse Lyapunov equations, SIAM Journal on Scientific Computing 21 (4) (2000) 1401–1418.

[23] Z.Y. Peng, New matrix iterative methods for constraint solutions of the matrix equation $AXB=C$, Journal of Computational and Applied Mathematics 235 (3) (2010) 726–735.

[24] Z.Y. Peng, A matrix LSQR iterative method to solve matrix equation $AXB=C$, International Journal of Computer Mathematics 87 (8) (2010) 1820–1830.

[25] F. Piao, Q. Zhang, Z. Wang, The solution to matrix equation $AX+X^TC=B$, Journal of the Franklin Institute 344 (8) (2007) 1056–1062.

[26] P. Sonneveld, CGS, A fast Lanczos-type solver for nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 10 (1989) 36–52.

[27] J. Sreedhar, P. Van Dooren, Periodic Schur form and some matrix equations, in: U. Helmke, R. Mennicken, J. Saurer (Eds.), Proceedings of the Symposium on the Mathematical Theory of Networks and Systems (MTNS'93), vol. 1, Regensburg, Germany, 1994, pp. 339–362.

[28] H.A. Van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (1992) 631–644.

[29] A. Varga, Periodic Lyapunov equations: some applications and new algorithms, International Journal of Control 67 (1) (1997) 69–87.

[30] A. Varga, S. Pieters, Gradient-based approach to solve optimal periodic output feedback control problems, Automatica 34 (4) (1998) 477–481.

[31] A. Varga, P. Van Dooren, Computational methods for periodic systems – an overview, in: Proceedings of IFAC Workshop on Periodic Control Systems, Como, Italy, 2001, pp. 171–176.

[32] M. Wang, X. Cheng, M. Wei, Iterative algorithms for solving the matrix equation $AXB+CX^TD=E$, Applied Mathematics and Computation 187 (2) (2007) 622–629.

[33] Q.W. Wang, J.W. Woude, H.X. Chang, A system of real quaternion matrix equations with applications, Linear Algebra and its Applications 431 (12) (2009) 2291–2303.

[34] Q.W. Wang, C.K. Li, Ranks and the least-norm of the general solution to a system of quaternion matrix equations, Linear Algebra and its Applications 430 (5–6) (2009) 1626–1640.

[35] B. Zhou, J. Lam, G.R. Duan, On Smith-type iterative algorithms for the Stein matrix equation, Applied Mathematics Letters 22 (7) (2009) 1038–1044.

[36] B. Zhou, G.R. Duan, Z.Y. Li, Gradient based iterative algorithm for solving coupled matrix equations, Systems and Control Letters 58 (5) (2009) 327–333.
[37] B. Zhou, J. Lamb, G.R. Duan, Toward solution of matrix equation $X = Af(X)B + C$, Linear Algebra and its Applications 435 (6) (2011) 1370–1398.