International Journal of Control, Automation, and Systems (2014) 12(6):1362-1365 ISSN:1598-6446 eISSN:2005-4092
DOI 10.1007/s12555-013-0516-8 http://www.springer.com/12555

Extending the GPBiCG Algorithm for Solving the Generalized Sylvester-transpose Matrix Equation

Masoud Hajarian

Abstract: By applying Kronecker product and vectorization operator, we extend the generalized product bi-conjugate gradient (GPBiCG) algorithm for solving the generalized Sylvester-transpose matrix equation $\sum_{i=1}^{r}(A_iXB_i + C_iX^TD_i) = E$. By using numerical results, we compare the new method with other popular iterative solvers in use today.

Keywords: GPBiCG algorithm, Kronecker product, Sylvester matrix equation.

1. INTRODUCTION

A digital filter can be characterized by the state variable equations

$$x(n+1) = Ax(n) + bu(n), \tag{1}$$

and

$$y(n) = cx(n) + du(n). \tag{2}$$

The solutions K and W of the Lyapunov matrix equations

$$K = AKA^T + bb^T, \tag{3}$$

and

$$W = A^TWA + c^Tc \tag{4}$$

can analyzed the quantization noise generated by a digital filter [1]. The Lyapunov and Sylvester matrix equations play a crucial role in control system analysis and design, system stability, boundedness analysis, filters design, transient behavior estimates, optimal and robust controllers [2-9]. In the last two decades, the Lyapunov and Sylvester matrix equations have attracted much attention and a large number of papers have studied linear matrix equations [10-15]. By extending the conjugate gradient least square (CGLS) approach, Dehghan and Hajarian introduced iterative algorithms for finding reflexive, anti-reflexive and generalized bisymmetric solutions of Lyapunov and Sylvester matrix equations [16-19]. By applying the hierarchical identification principle [20], Ding and Chen presented some iterative methods for solving Lyapunov and Sylvester matrix equations [21,22]. Wu *et al.* established an explicit solution to the generalized Sylvester matrix equation [23]. Zhou *et al.* presented a new Smith accelerative iteration for solving the Stein matrix equation $X = AXB + C$ [24]. In this article by extending the GPBiCG method we proposed an iterative matrix algorithm to solve the generalized Sylvester-transpose matrix equation

$$\sum_{i=1}^{r}(A_iXB_i + C_iX^TD_i) = E, \tag{5}$$

where $A_i, B_i, C_i, D_i, E \in \mathrm{R}^{m\times m}$ for $i = 1,2,...,r$ are known matrices and $X \in \mathrm{R}^{m\times m}$ is the matrix to be determined. The generalized Sylvester-transpose matrix equation (5) is general and includes several linear matrix equations such as the Lyapunov and Sylvester matrix equations. The GPBiCG technique is a new powerful method for solving the non-symmetric linear systems

$$Mx = b, \tag{6}$$

where $M \in \mathrm{R}^{n\times n}$ and $b \in \mathrm{R}^n$. The remainder of this article is organized as follows. First we shortly recall the GPBiCG technique and then we extend the GPBiCG algorithm for solving (5) in Section 2. In Section 3 simulation studies are performed. We give the conclusions in Section 4.

The following symbol conventions are used in this paper. The symbol $\mathrm{R}^{m\times n}$ denotes the set of all $m\times n$ real matrices. The notations A^T, $\mathrm{tr}(A)$ and $\|A\|$ stand for the transpose, the trace and the Frobenius norm of a complex matrix A, respectively. For a matrix $A \in \mathrm{R}^{m\times n}$, $\mathrm{vec}(A)$ is defined as $\mathrm{vec}(A) = (a_1^T\ a_2^T \dots a_n^T)^T$ where a_i is the i-th column of the matrix A. $A\otimes B$ stands for the Kronecker product of matrices A and B. For $A, B \in \mathrm{R}^{m\times n}$, we define an inner product in $\mathrm{R}^{m\times n}$: $< A, B >= \mathrm{trace}(B^TA)$; then $\mathrm{R}^{m\times n}$ is a Hilbert space. The matrix norm $\|.\|$ induced by the inner product is the Frobenius norm.

Manuscript received November 21, 2013; revised March 13, 2014; accepted April 13, 2014. Recommended by Associate Editor Choon Ki Ahn under the direction of Editor PooGyeon Park.

The author would like to express his heartfelt thanks to two anonymous referees for their valuable comments and careful reading of the manuscript.

Masoud Hajarian is with the Department of Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, General Campus, Evin, Tehran 19839, Iran (e-mail: m_hajarian@sbu.ac.ir; mhajarian@aut.ac.ir; masoudhajarian@gmail.com).

2. MAIN RESULTS

In this section first we recall the GPBiCG algorithm which is an powerful iterative method for computing a solution x to the linear systems (6). The Bi-Conjugate Gradient (Bi-CG) [2] is a well-known method for solving linear systems (6). So far, a number of hybrid Bi-CG methods such as the Conjugate Gradient Squared (CGS), Bi-CG STABilized (Bi-CGSTAB), BiCGSTAB2 and BiCGSTAB(l) have been developed to improve the convergence of Bi-CG [25]. Recently a class of generalizations of Bi-CGSTAB with little computational work and low storage costs was proposed. By defining a new three-term recurrence relation modelled after the residual polynomial of Bi-CG, Zhang proposed a diverse collection of generalized product-type methods based on Bi-CG (GPBiCG) that includes the well-known CGS, Bi-CGSTAB, and Bi-CGSTAB2, without the disadvantage of storing extra iterates like that in GMRES and GCR [26]. The GPBiCG algorithm can be summarized as follows [26].

Algorithm 1: The GPBiCG algorithm

Choose x and compute $r = b - Mx$;
Pick an arbitrary vector $\tilde{r}$ (for example $\tilde{r} = r$);
Set $t = \hat{t} = w = p = z = 0,\ \beta = 0$;
For $n = 1, 2, \ldots,$ until convergence

$p = r + \beta(p - u),\ v = Mp;$

$\alpha = \dfrac{<\tilde{r}, r>}{<\tilde{r}, v>},\ y = t - r - \alpha w + \alpha v \tilde{r},\ t = r - \alpha v,\ s = Mt;$

$\zeta = \dfrac{<y,y><s,t> - <y,t><s,y>}{<s,s><y,y> - <y,s><s,y>},$

$\eta = \dfrac{<s,s><y,t> - <y,s><s,t>}{<s,s><y,y> - <y,s><s,y>};$

(if $n = 1$, then $\zeta = \dfrac{<s,t>}{<s,s>},\ \eta = 0$);

$u = \zeta v + \eta(\hat{t} - r + \beta u),\ z = \zeta r + \eta z - \alpha u;$

$x = x + \alpha p + z,\ \hat{r} = t - \eta y - \zeta s,\ \beta = \dfrac{\alpha}{\zeta} \cdot \dfrac{<\tilde{r}, \hat{r}>}{<\tilde{r}, r>};$

$w = s + \beta v,\ r = \hat{r},\ \hat{t} = t.$

Here we develop Algorithm 1 to solve the generalized Sylvester-transpose matrix equation (5). First by applying Kronecker product and vectorization operator we can transform (5) into the following nonsymmetric linear systems

$$\underbrace{\left(\sum_{i=1}^{r} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)\right)}_{M} \underbrace{\mathrm{vec}(X)}_{x} = \underbrace{\mathrm{vec}(E)}_{b}, \quad (7)$$

where $M \in \mathrm{R}^{m^2 \times m^2}$, $x, b \in \mathrm{R}^{m^2}$, and $P \in \mathrm{R}^{m^2 \times m^2}$ is a unitary matrix [27]. The dimension of the associate matrix M is high when m is large. Such a dimensional problem leads to computational difficulty in that excessive computer memory is required for computation and inversion of large matrices. In order to overcome this problem, Algorithm 1 is directly extended to solve (5). By considering Algorithm 1 for (7), we have

$$\begin{aligned} r &= b - Mx \\ &= \mathrm{vec}(E) - \left(\sum_{i=1}^{r} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)\right)x, \end{aligned} \quad (8)$$

$$v = Mp = \left(\sum_{i=1}^{r} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)\right)p, \quad (9)$$

and

$$s = Mt = \left(\sum_{i=1}^{r} (B_i^T \otimes A_i + (D_i^T \otimes C_i)P)\right)t. \quad (10)$$

Now we define

$$p = \mathrm{vec}(P),\ u = \mathrm{vec}(U),\ r = \mathrm{vec}(R),\ v = \mathrm{vec}(V), \quad (11)$$

$$y = \mathrm{vec}(Y),\ z = \mathrm{vec}(Z),\ w = \mathrm{vec}(W),\ \tilde{r} = \mathrm{vec}(\tilde{R}), \quad (12)$$

$$t = \mathrm{vec}(T),\ \tilde{t} = \mathrm{vec}(\tilde{T}),\ x = \mathrm{vec}(X), \quad (13)$$

where $P, U, R, V, Z, W, Y, \tilde{R}, T, \tilde{T}, X \in \mathrm{R}^{m \times m}$. It follows from the above equations that

$$r = \mathrm{vec}(E) - \mathrm{vec}\left(\sum_{i=1}^{r} (A_i X B_i + C_i X^T D_i)\right), \quad (14)$$

$$\mathrm{vec}(V) = \mathrm{vec}\left(\sum_{i=1}^{r} (A_i P B_i + C_i P^T D_i)\right), \quad (15)$$

and

$$\mathrm{vec}(S) = \mathrm{vec}\left(\sum_{i=1}^{r} (A_i T B_i + C_i T^T D_i)\right). \quad (16)$$

Also we can write

$$\zeta = \frac{<Y,Y><S,T> - <Y,T><S,Y>}{<S,S><Y,Y> - <Y,S><S,Y>}, \quad (17)$$

and

$$\eta = \frac{<S,S><Y,T> - <Y,S><S,T>}{<S,S><Y,Y> - <Y,S><S,Y>}. \quad (18)$$

According to (11)-(18), we can obtain the following matrix form of Algorithm 1 for solving the generalized Sylvester-transpose matrix equation (5).

Algorithm 2 (Matrix form of GPBiCG):

Choose $X \in \mathrm{R}^{m \times m}$ and compute $R = E - \sum_{i=1}^{r} (A_i X B_i + C_i X^T D_i)$;
Pick an arbitrary matrix $\tilde{R} \in \mathrm{R}^{n \times n}$ (for example $\tilde{R} = R$);
Set $T = \hat{T} = W = P = Z = 0 \in \mathrm{R}^{n \times n},\ \beta = 0$;
For $n = 1, 2, \ldots,$ until convergence

$P = R + \beta(P - U),\ V = \sum_{i=1}^{r} (A_i P B_i + C_i P^T D_i);$

$\alpha = \dfrac{<\tilde{R}, R>}{<\tilde{R}, V>},\quad Y = T - R - \alpha W + \alpha V,\quad T = R - \alpha V;$

$S = \sum_{i=1}^{r} (A_i T B_i + C_i T^T D_i);$