Applied Mathematics Letters 34 (2014) 37–42

Contents lists available at ScienceDirect

Applied Mathematics Letters

journal homepage: www.elsevier.com/locate/aml

Matrix form of the CGS method for solving general coupled matrix equations

Masoud Hajarian *

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

ARTICLE INFO

Article history:
Received 6 December 2013
Received in revised form 17 March 2014
Accepted 18 March 2014
Available online 28 March 2014

Keywords:
Iterative method
CGS method
Linear system
Kronecker product
Vectorization operator

ABSTRACT

This paper deals with the problem of solving the general coupled matrix equations

$$\sum_{j=1}^{p} A_{ij}X_jB_{ij} = C_i, \quad i = 1, 2, \ldots, p,$$

(including several linear matrix equations as special cases) which plays important roles in system and control theory. Based on the conjugate gradients squared (CGS) method, a simple and efficient matrix algorithm is derived to solve the general coupled matrix equations. The derived iterative algorithm is illustrated by two numerical examples and is compared with other popular iterative solvers in use today.

1. Introduction

In this paper, the numerical solutions of the general coupled matrix equations

$$\sum_{j=1}^{p} A_{ij}X_jB_{ij} = C_i, \quad i = 1, 2, \ldots, p, \tag{1.1}$$

are considered, where $A_{ij}, X_j, B_{ij} \in \mathbf{C}^{m\times m}$ for $i, j = 1, 2, \ldots, p$. This kind of matrix equations includes various linear matrix equations such as

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

$$AXB + CYD = E, \tag{1.3}$$

$$\begin{cases} A_1XB_1 = C_1, \\ A_2XB_2 = C_2, \end{cases} \tag{1.4}$$

and

$$\begin{cases} AX - YB = E, \\ CX - YD = F. \end{cases} \tag{1.5}$$

Many problems in computational mathematics, control and system theory require the solution of the above matrix equations [1–3]. Hence solving matrix equations has been widely discussed in a large number of papers [4–7]. By applying the

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

http://dx.doi.org/10.1016/j.aml.2014.03.013

canonical correlation decomposition (CCD) of matrix pairs, Xu et al. obtained expressions of the least-squares solutions of the matrix equation (1.3), and sufficient and necessary conditions for the existence and uniqueness of the solutions [8]. In [9], a finite iterative method was introduced for solving the linear matrix equation (1.3) with unknown real matrices X and Y. Navarra et al. [10] introduced a representation of the general common solution X to the matrix equation (1.4). In [11], an analytical expression of the least squares Hermitian solution with the least norm of the matrix equation (1.4) over the skew field of quaternions was derived. Liao and Li applied the projection theorem, GSVD and CCD to investigate the solution of the linear matrix equation (1.4). Zhou et al. proposed a general iterative algorithm to produce the unique solution to general coupled linear matrix equations by means of the gradient search principle [3]. The gradient-based iterative (GI) algorithms [12,13] and the least squares based iterative algorithm [14] for solving (coupled) matrix equations were presented based on the hierarchical identification principle [15]. By development of the idea of the conjugate gradient least square method (CGLS) [16], some finite iterative methods were introduced for finding reflexive, generalized centro-symmetric and generalized bisymmetric solutions of (1.2)–(1.5) [17–19]. Recently based on a matrix form of the least-squares QR-factorization (LSQR) algorithm, the LSQR iterative method was proposed for solving (1.2) and (1.1) [20,21].

The CGS method is a Krylov subspace algorithm that can be applied to find fast solutions of the non-Hermitian linear systems

$$Ax = b, \tag{1.6}$$

where $A \in \mathbf{C}^{m\times m}$ and $b \in \mathbf{C}^m$. The purpose of this paper is to extend the CGS method for solving the general coupled matrix equation (1.1).

The remainder of this paper is organized as follows. In Section 2 first by recalling the CGS method and using the Kronecker product and the vectorization operator, the CGS method is extended to solve the general coupled matrix equation (1.1). A numerical example is presented to verify the method and compare the convergence rate between the method and some existing methods in Section 3.

Throughout this paper, the following notation is used. $\mathbf{C}^{m\times n}$ ($\mathbf{R}^{m\times n}$) stands for the sets of all $m \times n$ complex (real) matrices. For any matrix $A \in \mathbf{C}^{m\times n}$, the symbols A^H and $\mathrm{tr}(A)$ denote the conjugate transpose and the trace of A, respectively. For $A = (a_1, a_2, \ldots, a_n) = (a_{ij})$ and a matrix B, $A \otimes B = (a_{ij}B)$ is a Kronecker product and $\mathrm{vec}(A)$ is a vector defined by $\mathrm{vec}(A) = (a_1^T, a_2^T, \ldots a_n^T)^T$. The inner product is defined $\langle A, B\rangle = \mathrm{trace}(B^H A)$ for all $A, B \in \mathbf{C}^{m\times n}$, then $\mathbf{C}^{m\times n}$ is a Hilbert inner product space and the norm of a matrix generated by this inner product is the matrix Frobenius norm $\|.\|$. The notation

$$\mathscr{K}_m(d, A) := \{d, Ad, A^2d, \ldots, A^{m-1}d\},$$

is used for the mth Krylov subspace of $\mathbf{C}^m$ generated by $d \in \mathbf{C}^m$ and the $m \times m$ matrix A. The set of all complex polynomials of degree at most m is denoted by

$$\mathscr{P}_m := \{\varphi_m(\lambda) \equiv \gamma_0 + \gamma_1\lambda + \cdots + \gamma_m\lambda^m \mid \gamma_0, \gamma_1, \ldots, \gamma_m \in \mathbf{C}\}.$$

2. A new algorithm

In this section, first the CGS method is surveyed for solving (1.6). The CGS method is based on the biconjugate gradient (BiCG) method and not the original CG algorithm. One major drawback of the BiCG method is that it requires a multiplication with the conjugate transpose of A. A number of hybrid BiCG methods such as CGS have been presented to improve the convergence of BiCG and to avoid multiplication by A^H. The CGS method avoids using A^H and accelerates the convergence by squaring the BiCG polynomials [22]. The CGS method constructs an approximation $x_n \in x_0 + \mathscr{K}_{2n}(r_0, A)$ such that the residual is given by

$$r_n = b - Ax_n = (\varphi_n(A))^2 r_0. \tag{2.1}$$

The CGS algorithm can be summarized as follows [23]:

Algorithm 1 (*CGS Algorithm*)**.** Choose $x_0 \in \mathbf{C}^m$;
Set $p_0 = u_0 = r_0 = b - Ax_0$, $v_0 = Ap_0$;
Choose $\widetilde{r}_0$ such that $\rho_0 = \langle r_0, \widetilde{r}_0\rangle \neq 0$ (for example $\widetilde{r}_0 = r_0$);
For $n = 1, 2, \ldots$ until $(\|r_{n-1}\|/\|b\|) \leq \varepsilon$, do:
Set $\sigma_{n-1} = \langle v_{n-1}, \widetilde{r}_0\rangle$, $\alpha_{n-1} = \rho_{n-1}/\sigma_{n-1}$;
$q_n = u_{n-1} - \alpha_{n-1}v_{n-1}$;
Set $x_n = x_{n-1} + \alpha_{n-1}(u_{n-1} + q_n)$;
$r_n = r_{n-1} - \alpha_{n-1}A(u_{n-1} + q_n)$;
Set $\rho_n = \langle r_n, \widetilde{r}_0\rangle$, $\beta_n = \rho_n/\rho_{n-1}$;
$u_n = r_n + \beta_n q_n$;
$p_n = u_n + \beta_n(q_n + \beta_n p_{n-1})$; $v_n = Mp_n$.

In the above algorithm, the stopping tolerance ε is a small positive number. In exact arithmetic, Algorithm 1 terminates after a finite number, say n^*, of iterations. Usually, $x_{n^*} = A^{-1}b$ is the solution of the linear systems (1.6) [23]. For more details about the CGS algorithm see [23,22,24].