The 9^{th} Seminar on Numerical Analysis and its Applications

9-11 May 2022, University of Guilan, Rasht, Iran

Extended minimal residual biconjugate gradient stablilized method for generalized coupled Sylvester tensor equations

Mohammad Mahdi Izadkhah

Abstract. In this paper, a BiCG-like iterative method-minimal residual biconjugate gradient stablilized (MRBiCGStab)- is extended for solving the generalized coupled Sylvester tensor equations. The presented method uses tensor computations with no matricizations included. The reported numerical experiments show the performance of the proposed method.

1. Introduction

In this paper, we are concerned with the generalized coupled Sylvester tensor equations of the form

$$\sum_{j=1}^{n} \mathcal{X}_j \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN} = \mathcal{C}_i, \quad i = 1, 2, \ldots, n, \tag{1.1}$$

where the matrices $A_{ijl} \in \mathbb{C}^{I_{ijl} \times I_{ijl}}$, $\quad (i, j = 1, 2, \ldots, n, \quad l = 1, 2, \ldots, N)$, and the tensors $\mathcal{C}_i \in \mathbb{C}^{I_{i1} \times \cdots \times I_{iN}} (i = 1, 2, \ldots, n)$ are known and the tensors $\mathcal{X}_i \in \mathbb{C}^{I_{i1} \times \cdots \times I_{iN}} (i = 1, 2, \ldots, n)$ are unknown. In fact, for a positive integer N, an order N tensor $\mathcal{A} = (a_{i_1 \cdots i_N})(1 \leq i_j \leq I_j, j = 1, \ldots, N)$ is a multidimensional array with $I_1 I_2 \cdots I_N$ enteries. $\mathcal{O}$ with all enteries zero denotes the zero tensor. We use k-mode product $\times_k (k = 1, 2, \ldots, N)$ in (1.1) that will be defined later. In the sequel, some basic definitions which will be used, are given from [3].

Definition 1.1. The operator $\times_k (k = 1, 2, \ldots, N)$ represent the k-mode product of a tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ with a matrix $A \in \mathbb{C}^{J \times I_k}$ defined as

$$(\mathcal{X} \times_k A)_{i_1 i_2 \cdots i_{k-1} j i_{k+1} \cdots i_N} = \sum_{i_k=1}^{I_k} x_{i_1 i_2 \cdots i_{k-1} i_k i_{k+1} \cdots i_N} a_{j i_k}.$$

Definition 1.2. For a tensor $\mathcal{A} = (a_{i_1 \cdots i_N j_1 \cdots j_M}) \in \mathbb{C}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$, let $\mathcal{B} = (b_{i_1 \cdots i_M j_1 \cdots j_N}) \in \mathbb{C}^{J_1 \times \cdots \times J_M \times I_1 \times \cdots \times I_N}$ be the conjugate transpose of $\mathcal{A}$, where $b_{i_1 \cdots i_M j_1 \cdots j_N} = \bar{a}_{j_1 \cdots j_N i_1 \cdots i_M}$. The tensor $\mathcal{B}$ is denoted by $\mathcal{A}^*$. When $b_{i_1 \cdots i_M j_1 \cdots j_N} = a_{j_1 \cdots j_N i_1 \cdots i_M}$, $\mathcal{B}$ is called the transpose of $\mathcal{A}$, denoted by $\mathcal{A}^T$.

Definition 1.3. Let N and M be positive integers. The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$ is defined by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \sum_{j_1=1}^{J_1} \cdots \sum_{j_M=1}^{J_M} x_{i_1 \cdots i_N j_1 \cdots j_M} \bar{y}_{j_1 \cdots j_M i_1 \cdots i_N}.$$

Keywords: tensor equations, MRBiCGStab, iterative method, k-mode product.
AMS Mathematical Subject Classification [2010]: 15A10, 15A69, 15A72, 65F10.

So the tensor norm generated by this inner product is $\|\mathcal{X}\| = \sqrt{\langle \mathcal{X}, \mathcal{X}\rangle}$ which is called the tensor Frobenius norm.

Definition 1.4. Let $\mathcal{X}_i, \mathcal{Y}_i \in \mathbb{C}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_M}$, for $i = 1, 2, \dots, n$. If we put $\mathfrak{X} = (\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n)$ and $\mathfrak{Y} = (\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n)$, then $\langle\langle \mathfrak{X}, \mathfrak{Y}\rangle\rangle = \sum_{i=1}^{n} \langle \mathcal{X}_i, \mathcal{Y}_i\rangle$ is an inner product and the associated norm is defined by $\|\mathfrak{X}\|_* = \sqrt{\langle\langle \mathfrak{X}, \mathfrak{X}\rangle\rangle} = \sqrt{\sum_{i=1}^{n} \|\mathcal{X}_i\|^2}$.

We say that $\mathfrak{X}$ and $\mathfrak{Y}$ are orthogonal if and only if $\langle\langle \mathfrak{X}, \mathfrak{Y}\rangle\rangle = 0$.

Definition 1.5. Let $\mathbb{H}_i = \mathbb{C}^{I_{i1}\times\cdots\times I_{iN}} (i = 1, 2, \dots, n)$. Then

$$\mathcal{L} : \mathbb{H}_1 \times \mathbb{H}_2 \times \cdots \times \mathbb{H}_n \longrightarrow \mathbb{H}_1 \times \mathbb{H}_2 \times \cdots \times \mathbb{H}_n$$

$$\mathcal{L}(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n) = \begin{pmatrix} \sum_{j=1}^{n} \mathcal{X}_j \times_1 A_{1j1} \times_2 A_{1j2} \cdots \times_N A_{1jN} \\ \sum_{j=1}^{n} \mathcal{X}_j \times_1 A_{2j1} \times_2 A_{2j2} \cdots \times_N A_{2jN} \\ \cdots \\ \sum_{j=1}^{n} \mathcal{X}_j \times_1 A_{nj1} \times_2 A_{nj2} \cdots \times_N A_{njN} \end{pmatrix}.$$

According to Definition 1.5, the inear system (1.1) can be expressed as $\mathcal{L}(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n) = \mathcal{C}$, where $\mathcal{C} = (\mathcal{C}_1^T, \mathcal{C}_2^T, \dots, \mathcal{C}_n^T)^T$. We use Definition 1.4 and operator $\mathcal{L}$ in Definition 1.5 to construct the k-th residual tensor of EMRBiCGStab method $\mathcal{R}_k = \mathcal{C} - \mathcal{L}(\mathcal{X}_{1,k}, \mathcal{X}_{2,k}, \dots, \mathcal{X}_{n,k})$ where $\mathcal{X}_{i,k}, i = 1, 2, \dots, n$ are k-th approximations for the tensor solutions $\mathcal{X}_i, i = 1, 2, \dots, n$.

2. Extended MRBiCGStab Method

In the special case of (1.1), Beik et al. have developed the gradient-based iterative method [1]. We would like to mention that in the BiCGStab method for solving the system $Ax = b$, the product of the linear polynomials are considered in construction of the residual form. In order to overcome the stagnation of convergence of BiCGStab in some discretized dominated problems, quadratic stabilization polynomials are used in the MRBiCGStab algorithm [4]. The MRBiCGStab method computes an approximation x_{2k} whose residual is of the form $r^*_{2k} = Q^*_{2k}(A)P_{2k}(A)r_0$, where $Q^*_0(t) = 1$ and for $k \geq 1$ $\quad Q^*_{2k}(t) = (1 + w_1 t + w_2 t^2)(1 + w_3 t + w_4 t^2) \cdots (1 + w_{2k-1} t + w_{2k} t^2)$ and the parameters w_{2k-1} and w_{2k} are determined at the k-th iteration so that $\|r^*_{2k}\|$ is minimized. Based on this algorithm, we propose the Extended MRBiCGStab algorithm according to the tensor form for solving (1.1). Since the $\|\mathcal{R}^*_{k+2}\|_*$ is minimized over two dimensional vector space $\mathbb{R}^2$, it may be expected that the extended MRBiCGStab converges faster than the BiCGStab in which residual norm is minimized over one dimensional vector space $\mathbb{R}$.

3. Numerical example

In this section, we give numerical results of the EMRBiCGStab method to solve the following problem

$$\mathcal{X} \times_1 A_1 \times_2 A_2 \times_3 A_3 + \mathcal{Y} \times_1 B_1 \times_2 B_2 \times_3 B_3 = \mathcal{C}_1,$$
$$\mathcal{X} \times_1 E_1 \times_2 E_2 \times_3 E_3 + \mathcal{Y} \times_1 F_1 \times_2 F_2 \times_3 F_3 = \mathcal{C}_2,$$

which show the effectiveness of the proposed algorithm. The matrices $A_i, B_i, E_i, F_i, i = 1, 2, 3$ have been chosen from [2] by the MATLAB function `rand` of appropriate size, and construct the right-hand side tensors $\mathcal{C}_i, i = 1, 2$ such that the exact solutions $\mathcal{X}^*, \mathcal{Y}^* \in \mathbb{C}^{m\times n\times l}$ would be tensors with all enteries equal to one. The initial guess was taken to be the zero tensors and the stopping criterion $\frac{\|\mathcal{R}^*_k\|_*}{\|\mathcal{R}^*_0\|_*} \leq 10^{-6}$ or Max-iteration $= 2000$ were used. The corresponding convergence histories of EMRBiCGStab method with EBiCG [2] method are depicted in Figure 1.

Algorithm 1 The Extended MRBiCGStab method for solving (1.1)

1. **Input:** matrices A_{ijl} and tensors $\mathcal{X}_{j,0}, \mathcal{C}_i$, for $i, j = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, N$
2. Compute $\mathcal{R}^*_{i,0} = \mathcal{C}_i - \sum_{j=1}^{n} \mathcal{X}_{j,0} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}, \quad i = 1, 2, \ldots, n$
3. Put $\mathcal{P}^*_{i,0} = \mathcal{R}^*_{i,0}, \quad i = 1, 2, \ldots, n$
4. Choose arbitrary tensors $\tilde{\mathcal{R}}_{i,0}$ such that $\sum_{i=1}^{n} \langle \tilde{\mathcal{R}}_{i,0}, \mathcal{R}^*_{i,0} \rangle \neq 0$, for $i = 1, 2, \ldots, n$
6. For $k = 0, 2, \ldots, 2m, \ldots$ until $\|\mathcal{R}^*_{k+2}\|_*$ small enough Do
7. Compute $\mathcal{V}^*_{i,k} = \sum_{j=1}^{n} \mathcal{P}^*_{j,k} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, for $i = 1, 2, \ldots, n$
8. $\alpha_k = \frac{\sum_{i=1}^{n} \langle \mathcal{R}^*_{i,k}, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^{n} \langle \mathcal{V}^*_{i,k}, \tilde{\mathcal{R}}_{i,0} \rangle}$
9. $\bar{\mathcal{X}}_{i,k+1} = \mathcal{X}_{i,k} + \alpha_k \mathcal{P}^*_{i,k}$, for $i = 1, 2, \ldots, n$
10. $\bar{\mathcal{R}}_{i,k+1} = \mathcal{R}^*_{i,k} - \alpha_k \mathcal{V}^*_{i,k}$, for $i = 1, 2, \ldots, n$
11. Compute $\hat{\mathcal{R}}_{i,k+1} = \sum_{j=1}^{n} \bar{\mathcal{R}}_{j,k+1} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}, \quad i = 1, 2, \ldots, n$
12. $\beta_k = -\alpha_k \frac{\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+1}, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^{n} \langle \mathcal{R}^*_{i,k}, \tilde{\mathcal{R}}_{i,0} \rangle}$
13. $\bar{\mathcal{P}}_{i,k+1} = \bar{\mathcal{R}}_{i,k+1} + \beta_k \mathcal{P}^*_{i,k}$, for $i = 1, 2, \ldots, n$
14. $\hat{\mathcal{P}}_{i,k+1} = \hat{\mathcal{R}}_{i,k+1} + \beta_k \mathcal{V}^*_{i,k}$, for $i = 1, 2, \ldots, n$
15. $\hat{\mathcal{P}}^*_{i,k+1} = \sum_{j=1}^{n} \hat{\mathcal{P}}_{j,k+1} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, for $i = 1, 2, \ldots, n$
16. $\alpha_{k+1} = \frac{\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+1}, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^{n} \langle \hat{\mathcal{P}}^*_{i,k+1}, \tilde{\mathcal{R}}_{i,0} \rangle}$
17. $\bar{\mathcal{X}}_{i,k+2} = \bar{\mathcal{X}}_{i,k+1} + \alpha_{k+1} \bar{\mathcal{P}}_{i,k+1}$, for $i = 1, 2, \ldots, n$
18. $\bar{\mathcal{R}}_{i,k+2} = \bar{\mathcal{R}}_{i,k+1} - \alpha_{k+1} \hat{\mathcal{P}}_{i,k+1}$, for $i = 1, 2, \ldots, n$
19. $\hat{\mathcal{R}}_{i,k+2} = \hat{\mathcal{R}}_{i,k+1} - \alpha_{k+1} \hat{\mathcal{P}}^*_{i,k+1}$, for $i = 1, 2, \ldots, n$
20. $\hat{\mathcal{R}}^*_{i,k+2} = \sum_{j=1}^{n} \hat{\mathcal{R}}_{j,k+2} \times_1 A_{ij1} \times_2 A_{ij2} \cdots \times_N A_{ijN}$, for $i = 1, 2, \ldots, n$
21. $w_{k+1} = \frac{\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right) - \left(\sum_{i=1}^{n} \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}^*_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right)}{\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}^*_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right) - \left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right)}$
22. $w_{k+2} = \frac{\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right) - \left(\sum_{i=1}^{n} \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \bar{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle\right)}{\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right) - \left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+2}, \hat{\mathcal{R}}_{i,k+2} \rangle\right)\left(\sum_{i=1}^{n} \langle \hat{\mathcal{R}}^*_{i,k+2}, \hat{\mathcal{R}}^*_{i,k+2} \rangle\right)}$
23. $\mathcal{X}_{i,k+2} = \bar{\mathcal{X}}_{i,k+2} - w_{k+1} \bar{\mathcal{R}}_{i,k+2} - w_{k+2} \hat{\mathcal{R}}_{i,k+2}$, for $i = 1, 2, \ldots, n$
24. $\mathcal{R}^*_{i,k+2} = \bar{\mathcal{R}}_{i,k+2} + w_{k+1} \hat{\mathcal{R}}_{i,k+2} + w_{k+2} \hat{\mathcal{R}}^*_{i,k+2}$, for $i = 1, 2, \ldots, n$
25. $\beta_{k+1} = -\alpha_{k+1} \frac{\sum_{i=1}^{n} \langle \hat{\mathcal{R}}^*_{i,k+2}, \tilde{\mathcal{R}}_{i,0} \rangle}{\sum_{i=1}^{n} \langle \hat{\mathcal{R}}_{i,k+1}, \tilde{\mathcal{R}}_{i,0} \rangle}$
26. $\mathcal{P}^*_{i,k+2} = \mathcal{R}^*_{i,k+2} + \beta_{k+1}(\bar{\mathcal{P}}_{i,k+1} + w_{k+1} \hat{\mathcal{P}}_{i,k+1} + w_{k+2} \hat{\mathcal{P}}^*_{i,k+1})$, for $i = 1, 2, \ldots, n$
27. EndDo
28. **Output:** solution $\mathcal{X}_j$ for (1.1)

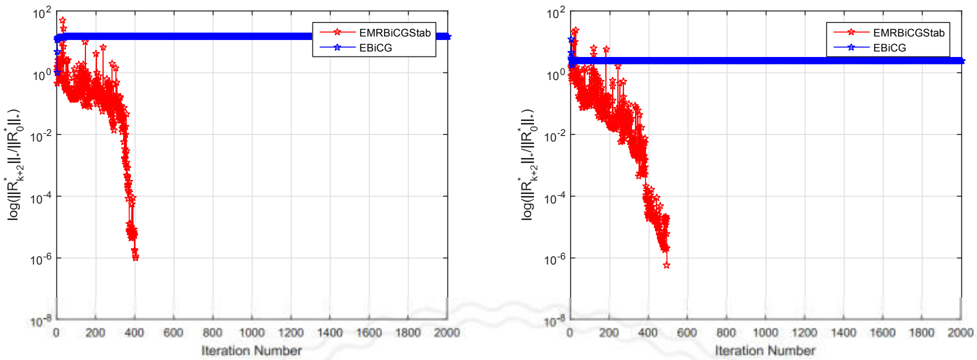

Figure 1: Convergence histories of REE for $m = 6, n = 4, l = 3$(left) and $m = 8, n = 4, l = 3$(right)

References

[1] F.P.A. Beik, S. Ahmadi-Asl, Residual norm steepest descent based iterative algorithms for Sylvester tensor equations, *Journal of Mathematical Modeling*, 2 (2015) 115–131.

[2] E.D. Khosravi, S. Karimi, Extended conjugate gradient squared and conjugate residual squared methods for solving the generalized coupled Sylvester tensor equations, *Transactions of the Institute of Measurement and Control*, 43 (2021) 519–527.

[3] T.G. Kolda, B.W. Bader, ; Tensor decompositions and applications, *SIAM Review*, 51 (2009) 455–500.

[4] J.H. Yun, M.S. Joo, BCG like methods for solving nonsymmetric linear systems, *Journal of The Chungcheong Mathematical Society*, 8 (1995) 55–69.

Mohammad Mahdi Izadkhah
Department of Computer Science, Faculty of Computer & Industrial Engineering, Birjand University of Technology, Birjand, Iran
email address: `izadkhah@birjandut.ac.ir`