Filomat 34:13 (2020), 4419–4427
https://doi.org/10.2298/FIL2013419H

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Solving Coupled Tensor Equations via Higher Order LSQR Methods

Masoud Hajarian[a]

[a]*Department of Applied Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran.*

Abstract. Tensors have a wide application in control theory, data mining, chemistry, information sciences, documents analysis and medical engineering. The material here is motivated by the development of the efficient numerical methods for solving the coupled tensor equations

$$\begin{cases} \mathcal{A}_1 *_M \mathcal{X} *_N \mathcal{B}_1 + \mathcal{C}_1 *_M \mathcal{Y} *_N \mathcal{D}_1 = \mathcal{E}_1, \\ \mathcal{A}_2 *_M \mathcal{X} *_N \mathcal{B}_2 + \mathcal{C}_2 *_M \mathcal{Y} *_N \mathcal{D}_2 = \mathcal{E}_2, \end{cases}$$

with Einstein product. We propose the tensor form of the LSQR methods to find the solutions $\mathcal{X}$ and $\mathcal{Y}$ of the coupled tensor equations. Finally we give some numerical examples to illustrate that our proposed methods are able to accurately and efficiently find the solutions of tensor equations with Einstein product.

1. Introduction

Tensors are becoming increasingly used to describe and solve several problems of documents analysis [7], psychometrics [27], chemometrics [38] and medical engineering [35], for more details see [10, 21, 34]. Research related to tensors has increased dramatically in recent years [3, 4, 8, 9, 11, 13, 14, 20, 42]. Solving linear systems in higher dimensions is one of the most important research topics in tensors [2, 19, 31]. Ding and Wei [15] generalized the Jacobi, Gauss-Seidel and Newton methods to solve multilinear systems with $\mathcal{M}$-tensors. From solving multi-linear $\mathcal{M}$-equations is equivalent to solving nonlinear systems of equations where the involving functions are P-functions, He et al. derived a Newton-type algorithm for solving multi-linear $\mathcal{M}$-equations [25]. Based on the rank-1 approximation of the coefficient tensor, Xie et al. obtained a new tensor method for solving symmetric $\mathcal{M}$-tensor systems [45]. In the work of Dolgov and Savostyanov [16], the alternating minimal energy (AMEn) methods were introduced for solving high-dimensional symmetrical positive definite (SPD) linear systems. In [44], two neural network models were develop for solving nonsingular multi-linear tensor system. Wang et al. derived continuous time neural network and modified continuous time neural networks to solve a multi-linear system with $\mathcal{M}$-tensors [43]. In [18], the greedy Tucker approximation algorithm was generalized for solving a tensor linear equation. Brazell et al. [5] introduced a higher order biconjugate gradient (HOBG) algorithm for solving a multilinear system of equations with Einstein product. Very recently, high order tensor equations such as the Stein tensor equation, the Lyapunov tensor equation and the Sylvester tensor equation have attracted great attention [12, 30, 37]. For example, Xu and Wang extended the biconjugate gradients (Bi-CG) and

2010 *Mathematics Subject Classification*. Primary 15A24; 65H10; Secondary 15A69; 65F10
Keywords. LSQR method; Coupled tensor equations; Einstein product.
Received: 21 January 2020; Accepted: 18 May 2020
Communicated by Yimin Wei
Email address: m_hajarian@sbu.ac.ir, mhajarian@aut.ac.ir, masoudhajarian@gmail.com (Masoud Hajarian)

biconjugate residual (Bi-CR) algorithms for finding the solution of the high order Stein tensor equation [46]. In [29], the sensitivity of the continuous Lyapunov tensor equation was investigated. Influenced by Brazell et al.'s paper [5], the tensor equations with Einstein product have drawn much attention [26, 39]. For instance, Wang and Xu [41] proposed a conjugate gradient least squares (CGLS) algorithm to solve some tensor equations with Einstein product.
In the following we will concentrate on the coupled tensor equations

$$\begin{cases} \mathcal{A}_1 *_M \mathcal{X} *_N \mathcal{B}_1 + \mathcal{C}_1 *_M \mathcal{Y} *_N \mathcal{D}_1 = \mathcal{E}_1, \\ \mathcal{A}_2 *_M \mathcal{X} *_N \mathcal{B}_2 + \mathcal{C}_2 *_M \mathcal{Y} *_N \mathcal{D}_2 = \mathcal{E}_2, \end{cases} \tag{1}$$

with Einstein product where $\mathcal{A}_1, \mathcal{A}_2 \in \mathbb{R}^{L_1\times\ldots\times L_M\times K_1\times\ldots\times K_M}$, $\mathcal{B}_1, \mathcal{B}_2 \in \mathbb{R}^{P_1\times\ldots\times P_N\times Q_1\times\ldots\times Q_N}$, $\mathcal{C}_1, \mathcal{C}_2 \in \mathbb{R}^{L_1\times\ldots\times L_M\times S_1\times\ldots\times S_M}$, $\mathcal{D}_1, \mathcal{D}_2 \in \mathbb{R}^{T_1\times\ldots\times T_N\times Q_1\times\ldots\times Q_N}$, $\mathcal{E}_1, \mathcal{E}_2 \in \mathbb{R}^{L_1\times\ldots\times L_M\times Q_1\times\ldots\times Q_N}$ are known and $\mathcal{X} \in \mathbb{R}^{K_1\times\ldots\times K_M\times P_1\times\ldots\times P_N}$ and $\mathcal{Y} \in \mathbb{R}^{S_1\times\ldots\times S_M\times T_1\times\ldots\times T_N}$ are unknown. The coupled tensor equations (1) have not been dealt with yet. In this article we aim to extend Krylov subspace methods for solving the coupled tensor equations (1). In Section 2, we obtain the tensor form of LSQR methods to efficiently find the solutions of (1). Three numerical examples will be discussed in Section 3. Finally we give some conclusions in Section 4.

1.1. Notation and preliminary definitions

In this subsection, we present the notation system used in this article and the basic concepts of tensors. For positive integers $L_1, L_2, \ldots, L_M$, an order M tensor $\mathcal{A} = (a_{l_1\ldots l_M})_{1\leq l_j\leq L_j} (j = 1, \ldots, M)$ is a multidimensional array with $L = L_1L_2\ldots L_M$ entries. We use $\mathbb{R}^{L_1\times\ldots\times L_M}$ to represent the set of the order M tensors of dimension $L_1 \times \ldots \times L_M$ with entries from $\mathbb{R}$. We now propose the Einstein product of two tensors. Given two tensors $\mathcal{A} \in \mathbb{R}^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N}$ and $\mathcal{B} \in \mathbb{R}^{J_1\times\ldots\times J_N\times K_1\times\ldots\times K_L}$, the Einstein product of $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\mathcal{A} *_N \mathcal{B}$ [5, 17] and is defined by

$$(\mathcal{A} *_N \mathcal{B})_{i_1\ldots i_M k_1\ldots k_L} = \sum_{j_1,\ldots,j_N} a_{i_1\ldots i_M j_1\ldots j_N} b_{j_1\ldots j_N k_1\ldots k_L}.$$

The Einstein product of tensors has important applications in theory of relativity and continuum mechanics [5, 17, 28]. When $M = N = L = 1$, the Einstein product reduces to the standard matrix multiplication. The symbols

$$(\mathcal{A}^T)_{i_1\ldots i_M j_1\ldots j_M} = (\mathcal{A})_{j_1\ldots j_M i_1\ldots i_M} \in \mathbb{R}^{J_1\times\ldots\times J_M\times I_1\times\ldots\times I_M},$$

$$\operatorname{tr}(\mathcal{A}) = \sum_{i_1,\ldots,i_M} a_{i_1\ldots i_M i_1\ldots i_M},$$

and

$$\|\mathcal{A}\| = \Big(\sum_{i_1,\ldots,i_M,j_1,\ldots,j_M} (a_{i_1\ldots i_M j_1\ldots j_M})^2 \Big)^{1/2},$$

will, respectively, denote the transpose, the trace and the Frobenius norm of a given tensor $\mathcal{A} \in \mathbb{R}^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_M}$. The inner product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N}$ is defined by

$$\langle \mathcal{A}, \mathcal{B} \rangle = \operatorname{tr}(\mathcal{B}^T *_M \mathcal{A}).$$

Tensors $\mathcal{A}$ and $\mathcal{B}$ are called orthogonal if $\langle \mathcal{A}, \mathcal{B} \rangle = 0$. Given $\mathcal{A}, \mathcal{B}, \mathcal{X} \in \mathbb{R}^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N}$, $\mathcal{C} \in \mathbb{R}^{I_1\times\ldots\times I_M\times K_1\times\ldots\times K_M}$, $\mathcal{Y} \in \mathbb{R}^{K_1\times\ldots\times K_M\times J_1\times\ldots\times J_N}$ and $\lambda \in \mathbb{R}$, it can be checked that [39, 41]

$$\langle \mathcal{A}, \mathcal{B} \rangle = \operatorname{tr}(\mathcal{B}^T *_M \mathcal{A}) = \operatorname{tr}(\mathcal{A} *_N \mathcal{B}^T) = \operatorname{tr}(\mathcal{B} *_N \mathcal{A}^T) = \operatorname{tr}(\mathcal{A}^T *_M \mathcal{B}) = \langle \mathcal{B}, \mathcal{A} \rangle, \tag{2}$$

$$\langle \lambda\mathcal{A}, \mathcal{B} \rangle = \lambda\langle \mathcal{A}, \mathcal{B} \rangle, \tag{3}$$

$$\langle \mathcal{X}, \mathcal{C} *_M \mathcal{Y} \rangle = \langle \mathcal{C}^T *_M \mathcal{X}, \mathcal{Y} \rangle, \tag{4}$$

$$(\mathcal{C} *_M \mathcal{Y})^T = \mathcal{Y}^T *_M \mathcal{C}^T. \tag{5}$$

Definition 1. *[41] Define the transformation* $\phi_{IJ} : R^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N} \rightarrow R^{I\times J}$ *with* $I = I_1 I_2 \ldots I_M$, $J = J_1 J_2 \ldots J_N$ *and* $\phi_{IJ}(\mathcal{A}) = A$ *defined component-wise as*

$$(\mathcal{A})_{i_1\ldots i_M j_1\ldots j_N} \rightarrow A_{st},$$

where $\mathcal{A} \in R^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N}$, $A \in R^{I\times J}$, $s = i_M + \sum_{k=1}^{M-1}((i_k - 1)\prod_{r=k+1}^{M} I_r)$ *and* $t = j_N + \sum_{k=1}^{N-1}((j_k - 1)\prod_{r=k+1}^{N} J_r)$.

Here is an example related to the above definition. Consider the tensor $\mathcal{A} \in \mathrm{R}^{2\times3\times3\times2}$ as follows

$$\mathcal{A}(:,:,1,1) = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \quad \mathcal{A}(:,:,2,1) = \begin{pmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix},$$

$$\mathcal{A}(:,:,3,1) = \begin{pmatrix} 13 & 14 & 15 \\ 16 & 17 & 18 \end{pmatrix}, \quad \mathcal{A}(:,:,1,2) = \begin{pmatrix} 19 & 20 & 21 \\ 22 & 23 & 24 \end{pmatrix},$$

$$\mathcal{A}(:,:,2,2) = \begin{pmatrix} 25 & 26 & 27 \\ 28 & 29 & 30 \end{pmatrix}, \quad \mathcal{A}(:,:,3,2) = \begin{pmatrix} 31 & 32 & 33 \\ 34 & 35 & 36 \end{pmatrix}.$$

It follows from Definition 1 that

$$\phi_{IJ}(\mathcal{A}) = A = \begin{pmatrix} 1 & 7 & 13 & 19 & 25 & 31 \\ 2 & 8 & 14 & 20 & 26 & 32 \\ 3 & 9 & 15 & 21 & 27 & 33 \\ 4 & 10 & 16 & 22 & 28 & 34 \\ 5 & 11 & 17 & 23 & 29 & 35 \\ 6 & 12 & 18 & 24 & 30 & 36 \end{pmatrix} \in \mathrm{R}^{6\times6}.$$

Remark 1. *It is worth mentioning that we can define the inverse function of* $\phi_{IJ} : R^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N} \rightarrow R^{I\times J}$ *as follows*

$$\phi_{IJ}^{-1} : R^{I\times J} :\rightarrow R^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_N},$$

with $A_{st} \rightarrow (\mathcal{A})_{i_1\ldots i_M j_1\ldots j_N}$, *where the s-th column of the matrix A consists the s-th element in the set* $\{A(:,\ldots,:,j_1,\ldots,j_N)|\forall j_1,\ldots,j_N\}$. *Here we sort all the elements in this set in lexicographic order, that is from* $(1,\ldots,1)$ *to* $(J_1,\ldots,J_N)$.

Lemma 1. *[41] For* $\mathcal{A} \in R^{I_1\times\ldots\times I_M\times J_1\times\ldots\times J_M}$, $\mathcal{X} \in R^{J_1\times\ldots\times J_M\times K_1\times\ldots\times K_N}$ *and* $\mathcal{C} \in R^{I_1\times\ldots\times I_M\times K_1\times\ldots\times K_N}$ *we have*

$$\mathcal{A} *_M \mathcal{X} = \mathcal{C} \Leftrightarrow \phi_{IJ}(\mathcal{A})\phi_{JK}(\mathcal{X}) = \phi_{IK}(\mathcal{C}).$$

Using the above lemma, we can obtain the following well-known proposition which gives the solvability conditions and the expression of the solutions to the coupled tensor equations (1). In the following, we set

$$A = \begin{pmatrix} \phi_{PQ}(\mathcal{B}_1)^T \otimes \phi_{LK}(\mathcal{A}_1) & \phi_{QT}(\mathcal{D}_1)^T \otimes \phi_{LS}(\mathcal{C}_1) \\ \phi_{PQ}(\mathcal{B}_2)^T \otimes \phi_{LK}(\mathcal{A}_2) & \phi_{QT}(\mathcal{D}_2)^T \otimes \phi_{LS}(\mathcal{C}_2) \end{pmatrix}, \tag{6}$$

$$x = \begin{pmatrix} \mathrm{vec}(\phi_{KP}(\mathcal{X})) \\ \mathrm{vec}(\phi_{ST}(\mathcal{Y})) \end{pmatrix}, \quad b = \begin{pmatrix} \mathrm{vec}(\phi_{LQ}(\mathcal{E}_1)) \\ \mathrm{vec}(\phi_{LQ}(\mathcal{E}_2)) \end{pmatrix}. \tag{7}$$

Proposition 1. *The coupled tensor equations* (1) *have a unique solution pair* $(\mathcal{X},\mathcal{Y})$ *if and only if* $\mathrm{rank}((A,b)) = \mathrm{rank}(A)$ *and A has a full column rank. In that case, the solution pair* $(\mathcal{X},\mathcal{Y})$ *of* (1) *can be expressed as*

$$\begin{pmatrix} \mathrm{vec}(\phi_{KP}(\mathcal{X})) \\ \mathrm{vec}(\phi_{ST}(\mathcal{Y})) \end{pmatrix} = (A^TA)^{-1}A^Tb. \tag{8}$$

2. Higher order LSQR methods for (1)

Because there is extensive literature on iterative methods to efficiently and effectively solve the unsymmetric linear systems [6, 36, 40], recently the development of iterative methods for solving linear matrix equations has become increasingly popular [22–24]. The main purpose of this section is to propose higher order LSQR methods for solving the coupled tensor equations (1). In this section, we begin by recalling the LSQR methods for solving the unsymmetric linear systems $Ax = b$. As is well known, by using the Golub-Kahan bidiagonalization process, two types of the LSQR method were constructed in [32] to compute an approximation solution of the linear systems $Ax = b$ and unconstrained least-squares problem $\min_x \|Ax - b\|$. First we present the pseudocode for the LSQR method in the following. Two types of the LSQR method can be summarized as follows [32, 33].

Algorithm 1. ***Type 1 of LSQR method***
$\tau(0) = 1$; $\xi(0) = -1$; $\omega(0) = 0$; $w(0) = 0$; $z(0) = 0$; $\beta(1) = \|b\|$; $\beta(1)u(1) = b$; $\alpha(1) = \|A^T u(1)\|$; $\alpha(1)v(1) = A^T u(1)$;
For $i = 1, 2, ...,$ until convergence, do:
$\xi(i) = -\xi(i-1)\beta(i)/\alpha(i)$; $z(i) = z(i-1)+\xi(i)v(i)$; $w(i) = (\tau(i-1)-\beta(i)w(i-1))/\alpha(i)$; $\omega(i) = \omega(i-1)+w(i)v(i)$; $\beta(i+1) = \|Av(i)-\alpha(i)u(i)\|$; $\beta(i+1)u(i+1) = Av(i)-\alpha(i)u(i)$; $\tau(i) = -\tau(i-1)\alpha(i)/\beta(i+1)$; $\alpha(i+1) = \|A^T u(i+1)-\beta(i+1)v(i)\|$; $\alpha(i+1)v(i+1) = A^T u(i+1) - \beta(i+1)v(i)$; $\gamma(i) = \beta(i+1)\xi(i)/(\beta(i+1)w(i) - \tau(i))$; $x(i) = z(i) - \gamma(i)\omega(i)$.

Algorithm 2. ***Type 2 of LSQR method***
$\theta(1) = \|A^T b\|$; $\theta(1)v(1) = A^T b$; $\rho(1) = \|Av(1)\|$; $\rho(1)p(1) = Av(1)$; $\omega(1) = v(1)/\rho(1)$; $\xi(1) = \theta(1)/\rho(1)$; $x(1) = \xi(1)\omega(1)$;
For $i = 1, 2, ...,$ until convergence, do:
$\theta(i+1) = \|A^T p(i) - \rho(i)v(i)\|$; $\theta(i+1)v(i+1) = A^T p(i) - \rho(i)v(i)$; $\rho(i+1) = \|Av(i+1) - \theta(i+1)p(i)\|$; $\rho(i+1)p(i+1) = Av(i+1) - \theta(i+1)p(i)$; $\omega(i+1) = (v(i+1) - \theta(i+1)\omega(i))/\rho(i+1)$; $\xi(i+1) = -\xi(i)\theta(i+1)/\rho(i+1)$; $x(i+1) = x(i) + \xi(i+1)\omega(i+1)$.

Theorem 1. *[33] LSQR algorithms return the minimum-norm solution.*

It can be easily see that the scalars $\alpha(i) \geq 0$, $\beta(i) \geq 0$, $\rho(i) \geq 0$ and $\theta(i) \geq 0$ are selected to make $\|u(i)\|_2 = 1$ and $\|v(i)\|_2 = 1$, respectively. Also the stopping criterion can be considered as $\|b - Ax(i)\|_2 < \varepsilon$ where ε is a small positive number (for more details see [33]). It is worth noting that Algorithms 1 and 2 can be applied for solving the linear system $Ax = b$ with the parameters (6) and (7). But obviously this system is a large system of equations and computing solution of such systems is a major computational challenge. Therefore it would be preferable to formulate Algorithms 1 and 2 in the tensor form. In order to obtain the tensor form of Algorithms 1 and 2, we substitute the parameters (6) and (7) into the above algorithms and then we obtain new parameters. It can be obtained that

$$\alpha(1)v(1) = A^T u(1) = \begin{pmatrix} \phi_{PQ}(\mathcal{B}_1)^T \otimes \phi_{LK}(\mathcal{A}_1) & \phi_{QT}(\mathcal{D}_1)^T \otimes \phi_{LS}(\mathcal{C}_1) \\ \phi_{PQ}(\mathcal{B}_2)^T \otimes \phi_{LK}(\mathcal{A}_2) & \phi_{QT}(\mathcal{D}_2)^T \otimes \phi_{LS}(\mathcal{C}_2) \end{pmatrix}^T u(1), \tag{9}$$

$$\beta(i+1)u(i+1) = Av(i) - \alpha(i)u(i) = \begin{pmatrix} \phi_{PQ}(\mathcal{B}_1)^T \otimes \phi_{LK}(\mathcal{A}_1) & \phi_{QT}(\mathcal{D}_1)^T \otimes \phi_{LS}(\mathcal{C}_1) \\ \phi_{PQ}(\mathcal{B}_2)^T \otimes \phi_{LK}(\mathcal{A}_2) & \phi_{QT}(\mathcal{D}_2)^T \otimes \phi_{LS}(\mathcal{C}_2) \end{pmatrix} v(i) - \alpha(i)u(i), \tag{10}$$

$$\alpha(i+1)v(i+1) = A^T u(i+1) - \beta(i+1)v(i) = \begin{pmatrix} \phi_{PQ}(\mathcal{B}_1)^T \otimes \phi_{LK}(\mathcal{A}_1) & \phi_{QT}(\mathcal{D}_1)^T \otimes \phi_{LS}(\mathcal{C}_1) \\ \phi_{PQ}(\mathcal{B}_2)^T \otimes \phi_{LK}(\mathcal{A}_2) & \phi_{QT}(\mathcal{D}_2)^T \otimes \phi_{LS}(\mathcal{C}_2) \end{pmatrix}^T u(i+1) - \beta(i+1)v(i), \tag{11}$$

$$\theta(1)v(1) = A^T b = \begin{pmatrix} \phi_{PQ}(\mathcal{B}_1)^T \otimes \phi_{LK}(\mathcal{A}_1) & \phi_{QT}(\mathcal{D}_1)^T \otimes \phi_{LS}(\mathcal{C}_1) \\ \phi_{PQ}(\mathcal{B}_2)^T \otimes \phi_{LK}(\mathcal{A}_2) & \phi_{QT}(\mathcal{D}_2)^T \otimes \phi_{LS}(\mathcal{C}_2) \end{pmatrix}^T \begin{pmatrix} \mathrm{vec}(\phi_{LQ}(\mathcal{E}_1)) \\ \mathrm{vec}(\phi_{LQ}(\mathcal{E}_2)) \end{pmatrix}, \tag{12}$$

$$\theta(i+1)v(i+1) = A^T p(i) - \rho(i)v(i) = \begin{pmatrix} \phi_{PQ}(\mathcal{B}_1)^T \otimes \phi_{LK}(\mathcal{A}_1) & \phi_{QT}(\mathcal{D}_1)^T \otimes \phi_{LS}(\mathcal{C}_1) \\ \phi_{PQ}(\mathcal{B}_2)^T \otimes \phi_{LK}(\mathcal{A}_2) & \phi_{QT}(\mathcal{D}_2)^T \otimes \phi_{LS}(\mathcal{C}_2) \end{pmatrix}^T p(i) - \rho(i)v(i), \tag{13}$$

$$\rho(i+1)p(i+1) = Av(i+1) - \theta(i+1)p(i) = \begin{pmatrix} \phi_{PQ}(\mathcal{B}_1)^T \otimes \phi_{LK}(\mathcal{A}_1) & \phi_{QT}(\mathcal{D}_1)^T \otimes \phi_{LS}(\mathcal{C}_1) \\ \phi_{PQ}(\mathcal{B}_2)^T \otimes \phi_{LK}(\mathcal{A}_2) & \phi_{QT}(\mathcal{D}_2)^T \otimes \phi_{LS}(\mathcal{C}_2) \end{pmatrix} v(i+1) - \theta(i+1)p(i). \quad (14)$$

In view of (9)-(11), we define new parameters as follows

$$p(i) = \begin{pmatrix} \mathrm{vec}(\phi_{LQ}(\mathcal{P}_1(i))) \\ \mathrm{vec}(\phi_{LQ}(\mathcal{P}_2(i))) \end{pmatrix}, \ u(i) = \begin{pmatrix} \mathrm{vec}(\phi_{LQ}(\mathcal{U}_1(i))) \\ \mathrm{vec}(\phi_{LQ}(\mathcal{U}_2(i))) \end{pmatrix}, \widetilde{r_0} = \begin{pmatrix} \mathrm{vec}(\phi_{LQ}(\widetilde{\mathcal{R}}_1(0))) \\ \mathrm{vec}(\phi_{LQ}(\widetilde{\mathcal{R}}_1(0))) \end{pmatrix}, r(i) = \begin{pmatrix} \mathrm{vec}(\phi_{LQ}(\mathcal{R}_1(i))) \\ \mathrm{vec}(\phi_{LQ}(\mathcal{R}_2(i))) \end{pmatrix}, \quad (15)$$

$$v(i) = \begin{pmatrix} \mathrm{vec}(\phi_{KP}(\mathcal{V}_1(i))) \\ \mathrm{vec}(\phi_{ST}(\mathcal{V}_2(i))) \end{pmatrix}, z(i) = \begin{pmatrix} \mathrm{vec}(\phi_{KP}(\mathcal{Z}_1(i))) \\ \mathrm{vec}(\phi_{ST}(\mathcal{Z}_2(i))) \end{pmatrix}, q(i) = \begin{pmatrix} \mathrm{vec}(\phi_{KP}(\mathcal{Q}_1(i))) \\ \mathrm{vec}(\phi_{ST}(\mathcal{Q}_2(i))) \end{pmatrix}, \quad (16)$$

$$x(i) = \begin{pmatrix} \mathrm{vec}(\phi_{KP}(\mathcal{X}(i))) \\ \mathrm{vec}(\phi_{ST}(\mathcal{Y}(i))) \end{pmatrix}, \ \omega(i) = \begin{pmatrix} \mathrm{vec}(\phi_{KP}(\mathcal{W}_1(i))) \\ \mathrm{vec}(\phi_{ST}(\mathcal{W}_2(i))) \end{pmatrix}, \quad (17)$$

where

$$\mathcal{P}_1(i), \mathcal{P}_2(i), \mathcal{U}_1(i), \mathcal{U}_2(i), \mathcal{R}_1(i), \mathcal{R}_2(i), \widetilde{\mathcal{R}}_1(0), \widetilde{\mathcal{R}}_2(0) \in \mathrm{R}^{\mathrm{L}_1 \times \ldots \times \mathrm{L}_M \times \mathrm{Q}_1 \times \ldots \times \mathrm{Q}_N}, \quad (18)$$

and

$$\mathcal{V}_1(i), \mathcal{Z}_1(i), \mathcal{W}_1(i), \mathcal{X}(i) \in \mathrm{R}^{\mathrm{K}_1 \times \ldots \times \mathrm{K}_M \times \mathrm{P}_1 \times \ldots \times \mathrm{P}_N}, \quad \mathcal{V}_2(i), \mathcal{Z}_2(i), \mathcal{W}_2(i), \mathcal{Y}(i) \in \mathrm{R}^{\mathrm{S}_1 \times \ldots \times \mathrm{S}_M \times \mathrm{T}_1 \times \ldots \times \mathrm{T}_N}. \quad (19)$$

Now using the above results and Remark 1, we can get the tensor forms of Algorithms 1 and 2 to solve the coupled tensor equations (1). The resulting algorithms are given as Algorithms 3 and 4, respectively.

Algorithm 3. ***Type 1 of LSQR method for*** (1)
$\tau(0) = 1$; $\xi(0) = -1$; $\mathcal{W}_1(0) = 0$; $\mathcal{W}_2(0) = 0$; $w(0) = 0$; $\mathcal{Z}_1(0) = 0$; $\mathcal{Z}_2(0) = 0$;
$\beta(1) = [\|\mathcal{E}_1\|^2 + \|\mathcal{E}_2\|^2]^{1/2}$; $\mathcal{U}_1(1) = \mathcal{E}_1/\beta(1)$; $\mathcal{U}_2(1) = \mathcal{E}_2/\beta(1)$;
$\alpha(1) = [\|\mathcal{A}_1^T *_M \mathcal{U}_1(1) *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{U}_2(1) *_N \mathcal{B}_2^T\|^2 + \|\mathcal{C}_1^T *_M \mathcal{U}_1(1) *_N \mathcal{D}_1^T + \mathcal{C}_2^T *_M \mathcal{U}_2(1) *_N \mathcal{D}_2^T\|^2]^{1/2}$;
$\mathcal{V}_1(1) = [\mathcal{A}_1^T *_M \mathcal{U}_1(1) *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{U}_2(1) *_N \mathcal{B}_2^T]/\alpha(1)$;
$\mathcal{V}_2(1) = [\mathcal{C}_1^T *_M \mathcal{U}_1(1) *_N \mathcal{D}_1^T + \mathcal{C}_2^T *_M \mathcal{U}_2(1) *_N \mathcal{D}_2^T]/\alpha(1)$
For $i = 1, 2, \ldots$, *until convergence, do:*
$\xi(i) = -\xi(i-1)\beta(i)/\alpha(i)$;
$\mathcal{Z}_1(i) = \mathcal{Z}_1(i-1) + \xi(i)\mathcal{V}_1(i)$;
$\mathcal{Z}_2(i) = \mathcal{Z}_2(i-1) + \xi(i)\mathcal{V}_2(i)$;
$w(i) = (\tau(i-1) - \beta(i)w(i-1))/\alpha(i)$;
$\mathcal{W}_1(i) = \mathcal{W}_1(i-1) + w(i)\mathcal{V}_1(i)$;
$\mathcal{W}_2(i) = \mathcal{W}_2(i-1) + w(i)\mathcal{V}_2(i)$;
$\beta(i+1) = [\|\mathcal{A}_1 *_M \mathcal{V}_1(i) *_N \mathcal{B}_1 + \mathcal{C}_1 *_M \mathcal{V}_2(i) *_N \mathcal{D}_1 - \alpha(i)\mathcal{U}_1(i)\|^2$
$+ \|\mathcal{A}_2 *_M \mathcal{V}_1(i) *_N \mathcal{B}_2 + \mathcal{C}_2 *_M \mathcal{V}_2(i) *_N \mathcal{D}_2 - \alpha(i)\mathcal{U}_2(i)\|^2]^{1/2}$;
$\mathcal{U}_1(i+1) = [\mathcal{A}_1 *_M \mathcal{V}_1(i) *_N \mathcal{B}_1 + \mathcal{C}_1 *_M \mathcal{V}_2(i) *_N \mathcal{D}_1 - \alpha(i)\mathcal{U}_1(i)]/\beta(i+1)$;
$\mathcal{U}_2(i+1) = [\mathcal{A}_2 *_M \mathcal{V}_1(i) *_N \mathcal{B}_2 + \mathcal{C}_2 *_M \mathcal{V}_2(i) *_N \mathcal{D}_2 - \alpha(i)\mathcal{U}_2(i)]/\beta(i+1)$;
$\tau(i) = -\tau(i-1)\alpha(i)/\beta(i+1)$;
$\alpha(i+1) = [\|\mathcal{A}_1^T *_M \mathcal{U}_1(i+1) *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{U}_2(i+1) *_N \mathcal{B}_2^T - \beta(i+1)\mathcal{V}_1(i)\|^2$
$+ \|\mathcal{C}_1^T *_M \mathcal{U}_1(i+1) *_N \mathcal{D}_1^T + \mathcal{C}_2^T *_M \mathcal{U}_2(i+1) *_N \mathcal{D}_2^T - \beta(i+1)\mathcal{V}_2(i)\|^2]^{1/2}$;
$\mathcal{V}_1(i+1) = [\mathcal{A}_1^T *_M \mathcal{U}_1(i+1) *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{U}_2(i+1) *_N \mathcal{B}_2^T - \beta(i+1)\mathcal{V}_1(i)]/\alpha(i+1)$;
$\mathcal{V}_2(i+1) = [\mathcal{C}_1^T *_M \mathcal{U}_1(i+1) *_N \mathcal{D}_1^T + \mathcal{C}_2^T *_M \mathcal{U}_2(i+1) *_N \mathcal{D}_2^T - \beta(i+1)\mathcal{V}_2(i)]/\alpha(i+1)$;
$\gamma(i) = \beta(i+1)\xi(i)/(\beta(i+1)w(i) - \tau(i))$;
$\mathcal{X}(i) = \mathcal{Z}_1(i) - \gamma(i)\mathcal{W}_1(i)$;
$\mathcal{Y}(i) = \mathcal{Z}_2(i) - \gamma(i)\mathcal{W}_2(i)$.

Algorithm 4. ***Type 2 of LSQR method for*** (1)
$\theta(1) = [\|\mathcal{A}_1^T *_M \mathcal{E}_1 *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{E}_2 *_N \mathcal{B}_2^T\|^2 + \|\mathcal{C}_1^T *_M \mathcal{E}_1 *_N \mathcal{D}_1^T + \mathcal{C}_2^T *_M \mathcal{E}_2 *_N \mathcal{D}_2^T\|^2]^{1/2}$;
$\mathcal{V}_1(1) = [\mathcal{A}_1^T *_M \mathcal{E}_1 *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{E}_2 *_N \mathcal{B}_2^T]/\theta(1)$;

$\mathcal{V}_2(1) = [C_1^T *_M \mathcal{E}_1 *_N \mathcal{D}_1^T + C_2^T *_M \mathcal{E}_2 *_N \mathcal{D}_2^T]/\theta(1);$
$\rho(1) = [\|\mathcal{A}_1 *_M \mathcal{V}_1(1) *_N \mathcal{B}_1 + C_1 *_M \mathcal{V}_2(1) *_N \mathcal{D}_1\|^2 + \|\mathcal{A}_2 *_M \mathcal{V}_1(1) *_N \mathcal{B}_2 + C_2 *_M \mathcal{V}_2(1) *_N \mathcal{D}_2\|^2]^{1/2};$
$\mathcal{P}_1(1) = [\mathcal{A}_1 *_M \mathcal{V}_1(1) *_N \mathcal{B}_1 + C_1 *_M \mathcal{V}_2(1) *_N \mathcal{D}_1]/\rho(1);$
$\mathcal{P}_2(1) = [\mathcal{A}_2 *_M \mathcal{V}_1(1) *_N \mathcal{B}_2 + C_2 *_M \mathcal{V}_2(1) *_N \mathcal{D}_2]/\rho(1);$
$\mathcal{W}_1(1) = \mathcal{V}_1(1)/\rho(1);$
$\mathcal{W}_2(1) = \mathcal{V}_2(1)/\rho(1);$
$\xi(1) = \theta(1)/\rho(1);$
$\mathcal{X}(1) = \xi(1)\mathcal{W}_1(1);$
$\mathcal{Y}(1) = \xi(1)\mathcal{W}_2(1);$
For $i = 1, 2, ...,$ until convergence, do:
$\theta(i+1) = [\|\mathcal{A}_1^T *_M \mathcal{P}_1(i) *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{P}_2(i) *_N \mathcal{B}_2^T - \rho(i)\mathcal{V}_1(i)\|^2$
$+ \|C_1^T *_M \mathcal{P}_1(i) *_N \mathcal{D}_1^T + C_2^T *_M \mathcal{P}_2(i) *_N \mathcal{D}_2^T - \rho(i)\mathcal{V}_2(i)\|^2]^{1/2};$
$\mathcal{V}_1(i+1) = [\mathcal{A}_1^T *_M \mathcal{P}_1(i) *_N \mathcal{B}_1^T + \mathcal{A}_2^T *_M \mathcal{P}_2(i) *_N \mathcal{B}_2^T - \rho(i)\mathcal{V}_1(i)]/\theta(i+1);$
$\mathcal{V}_2(i+1) = [C_1^T *_M \mathcal{P}_1(i) *_N \mathcal{D}_1^T + C_2^T *_M \mathcal{P}_2(i) *_N \mathcal{D}_2^T - \rho(i)\mathcal{V}_2(i)]/\theta(i+1);$
$\rho(i+1) = [\|\mathcal{A}_1 *_M \mathcal{V}_1(i+1) *_N \mathcal{B}_1 + C_1 *_M \mathcal{V}_2(i+1) *_N \mathcal{D}_1 - \theta(i+1)\mathcal{P}_1(i)\|^2$
$+ \|\mathcal{A}_2 *_M \mathcal{V}_1(i+1) *_N \mathcal{B}_2 + C_2 *_M \mathcal{V}_2(i+1) *_N \mathcal{D}_2 - \theta(i+1)\mathcal{P}_2(i)\|^2]^{1/2};$
$\mathcal{P}_1(i+1) = [\mathcal{A}_1 *_M \mathcal{V}_1(i+1) *_N \mathcal{B}_1 + C_1 *_M \mathcal{V}_2(i+1) *_N \mathcal{D}_1 - \theta(i+1)\mathcal{P}_1(i)]/\rho(i+1);$
$\mathcal{P}_2(i+1) = [\mathcal{A}_2 *_M \mathcal{V}_1(i+1) *_N \mathcal{B}_2 + C_2 *_M \mathcal{V}_2(i+1) *_N \mathcal{D}_2 - \theta(i+1)\mathcal{P}_2(i)]/\rho(i+1);$
$\mathcal{W}_1(i+1) = (\mathcal{V}_1(i+1) - \theta(i+1)\mathcal{W}_1(i))/\rho(i+1);$
$\mathcal{W}_2(i+1) = (\mathcal{V}_2(i+1) - \theta(i+1)\mathcal{W}_2(i))/\rho(i+1);$
$\xi(i+1) = -\xi(i)\theta(i+1)/\rho(i+1);$
$\mathcal{X}(i+1) = \mathcal{X}(i) + \xi(i+1)\mathcal{W}_1(i+1);$
$\mathcal{Y}(i+1) = \mathcal{Y}(i) + \xi(i+1)\mathcal{W}_2(i+1).$

Stopping criterion. As a stopping criterion, we choose to satisfy

$$\sqrt{\sum_{k=1}^{2} \|\mathcal{E}_k - \mathcal{A}_k *_M \mathcal{X}(i) *_N \mathcal{B}_k - C_k *_M \mathcal{Y}(i) *_N \mathcal{D}_k\|^2} = 0$$

So our convergence criterion for the LSQR methods becomes

$$\sqrt{\sum_{k=1}^{2} \|\mathcal{E}_k - \mathcal{A}_k *_M \mathcal{X}(i) *_N \mathcal{B}_k - C_k *_M \mathcal{Y}(i) *_N \mathcal{D}_k\|^2} \leq \texttt{tol},$$

where `tol` is a chosen fixed threshold.

3. Numerical experiments

To numerically test the effectiveness and efficiency of the LSQR methods, we give two numerical examples. The calculations were carried out in MATLAB [1].
First we consider the Sylvester tensor equation

$$\mathcal{A} *_M \mathcal{X} *_N \mathcal{B} + C *_M \mathcal{Y} *_N \mathcal{D} = \mathcal{E}, \tag{20}$$

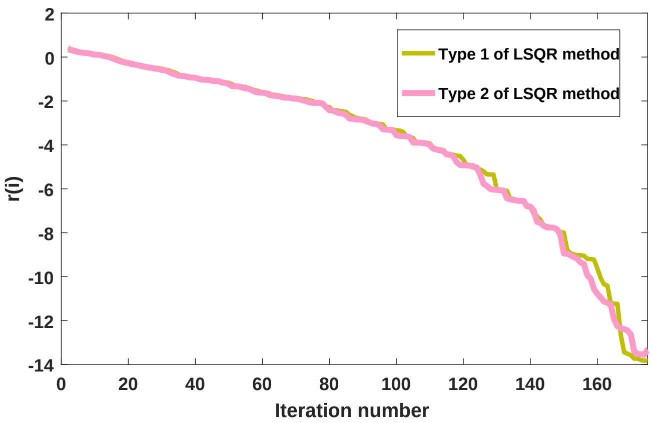
with

$$\mathcal{A} = -10 \times tenrand([3\ 2\ 4\ 3]) \in \mathbb{R}^{3\times2\times4\times3}, \ \ \mathcal{B} = 20 \times tenrand([3\ 4\ 5\ 2]) \in \mathbb{R}^{3\times4\times5\times2},$$
$$C = -30 \times tenrand([3\ 2\ 5\ 2]) \in \mathbb{R}^{3\times2\times5\times2}, \ \ \mathcal{D} = 40 \times tenrand([2\ 5\ 5\ 2]) \in \mathbb{R}^{2\times5\times5\times2},$$

and

$$\mathcal{E} = tenrand([3\ 2\ 5\ 2]) \in \mathbb{R}^{3\times2\times5\times2}.$$

Figure 1: Plot of the residual of Algorithms 1 and 2.



The results obtained for this example are depicted in Figure 1 where

$$r(i) = \sqrt{\log \|\mathcal{E} - \mathcal{A} *_M \mathcal{X}(i) *_N \mathcal{B} - \mathcal{C} *_M \mathcal{Y}(i) *_N \mathcal{D}\|}.$$

As the second example, we study the coupled tensor equations (1) with

$$\mathcal{A}_1 = 22 \times tenrand([3\ 2\ 4\ 3]) \in \mathbb{R}^{3\times2\times4\times3},\ \ \mathcal{B}_1 = 12 \times tenrand([3\ 4\ 5\ 2]) \in \mathbb{R}^{3\times4\times5\times2},$$
$$\mathcal{C}_1 = 45 \times tenrand([3\ 2\ 5\ 2]) \in \mathbb{R}^{3\times2\times5\times2},\ \ \mathcal{D}_1 = 55 \times tenrand([2\ 5\ 5\ 2]) \in \mathbb{R}^{2\times5\times5\times2},$$
$$\mathcal{A}_2 = 66 \times tenrand([3\ 2\ 4\ 3]) \in \mathbb{R}^{3\times2\times4\times3},\ \ \mathcal{B}_2 = 11 \times tenrand([3\ 4\ 5\ 2]) \in \mathbb{R}^{3\times4\times5\times2},$$
$$\mathcal{C}_2 = 33 \times tenrand([3\ 2\ 5\ 2]) \in \mathbb{R}^{3\times2\times5\times2},\ \ \mathcal{D}_2 = 40 \times tenrand([2\ 5\ 5\ 2]) \in \mathbb{R}^{2\times5\times5\times2},$$

and

$$\mathcal{E}_1 = tenrand([3\ 2\ 5\ 2]) \in \mathbb{R}^{3\times2\times5\times2},\ \ \mathcal{E}_2 = tenrand([3\ 2\ 5\ 2]) \in \mathbb{R}^{3\times2\times5\times2}.$$

The results of numerical results are illustrated in Figure 2 where

$$r(i) = \log \sqrt{\sum_{k=1}^{2} \|\mathcal{E}_k - \mathcal{A}_k *_M \mathcal{X}(i) *_N \mathcal{B}_k - \mathcal{C}_k *_M \mathcal{Y}(i) *_N \mathcal{D}_k\|^2}.$$

The numerical experiments confirmed the accuracy and efficiency of Algorithms 1 and 2.

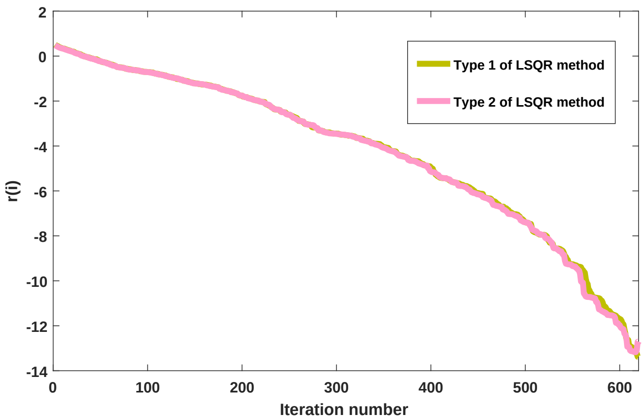
4. Conclusions

In this paper, we developed two tensor types of the LSQR method for computing the solutions of the coupled tensor equations (1). The resulting LSQR algorithms were easy to implement and simple to use. Finally, we presented numerical results which illustrate the efficiency of the LSQR methods.

Acknowledgments

The author wishes to express his gratitude to the editor and an anonymous reviewer for helpful remarks and suggestions.

Figure 2: Plot of the residual of Algorithms 1 and 2.



References

[1] B. W. Bader, T. G. Kolda, et al. Matlab tensor toolbox version 2.6. *Available online*, February 2015. URL: http://www.sandia.gov/ tgkolda/TensorToolbox/.
[2] G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26:2133–2159, 2005.
[3] H. Bozorgmanesh and M. Hajarian. Convergence of a transition probability tensor of a higher-order markov chain to the stationary probability vector. *Numerical Linear Algebra with Applications*, 23:972–988, 2016.
[4] H. Bozorgmanesh and M. Hajarian. Solving tensor E-eigenvalue problem faster . *Applied Mathematics Letters*, 2019, In Press.
[5] M. Brazell, N. Li, C. Navasca, and C. Tamon. Solving multilinear systems via tensor inversion. *SIAM Journal on Matrix Analysis and Applications*, 34:542–570, 2013.
[6] C. G. Broyden and M. T. Vespucci. *Krylov Solvers for Linear Algebraic Systems*. ELSEVIER, 2004.
[7] D. Cai, X. He, and J. Han. Tensor space model for document analysis. In: Proc. 29th Annu. ACM SIGIR Int. Conf. Research and Development in Information Retrieval, pages 625–626, Washington, August 2006.
[8] M. Che, L. Qi, and Y. Wei. Positive-definite tensors to nonlinear complementarity problems. *Journal of Optimization Theory and Applications*, 168:475–487, 2016.
[9] M. Che and Y. Wei. Randomized algorithms for the approximations of Tucker and the tensor train decompositions. *Advances in Computational Mathematics*, 45:395–428, 2019.
[10] M. Che and Y. Wei. *Theory and Computation of Complex Tensors and its Applications*. Springer, Singapore, 2020.
[11] M. Che, Y. Wei, and H. Yan. The computation for low multilinear rank approximations of tensors via power scheme and random projection. *SIAM Journal on Matrix Analysis and Applications*, 41:605–636, 2020.
[12] Z. Chen and L. Lu. A projection method and Kronecker product preconditioner for solving Sylvester tensor equations. *Science China Mathematics*, 55:1281–1292, 2012.
[13] W. Ding, Z. Hou, and Y. Wei. Tensor logarithmic norm and its applications. *Numerical Linear Algebra with Applications*, 23:989–1006, 2016.
[14] W. Ding, L. Qi, and Y. Wei. $\mathcal{M}$-tensors and nonsingular $\mathcal{M}$-tensors. *Linear Algebra and its Applications*, 439:3264–3278, 2013.
[15] W. Ding and Y. Wei. Solving multi-linear systems with $\mathcal{M}$-tensors. *Journal of Scientific Computing*, 68:689–715, 2016.
[16] S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM Journal on Scientific Computing*, 36:A2248–A2271, 2014.
[17] A. Einstein. The foundation of the general theory of relativity. In A. Kox, M. Klein, and R. Schulmann, editors, *The Collected Papers of Albert Einstein*, volume 6, pages 146–200, Princeton University Press, Princeton, 2007.
[18] I. Georgieva and C. Hofreither. Greedy low-rank approximation in Tucker format of solutions of tensor linear systems. *Journal of Computational and Applied Mathematics*, 358:206–220, 2019.
[19] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36:53–78, 2013.
[20] Y. Guan, M. T. Chu, and D. Chu. SVD-based algorithms for the best rank-1 approximation of a symmetric tensor. *SIAM Journal on Matrix Analysis and Applications*, 39:1095–1115, 2018.
[21] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer, Berlin, 2012.

[22] M. Hajarian. The generalized QMRCGSTAB algorithm for solving Sylvester-transpose matrix equations. *Applied Mathematics Letters*, 26:1013–1017, 2013.

[23] M. Hajarian. Developing BiCOR and CORS methods for coupled Sylvester-transpose and periodic Sylvester matrix equations. *Applied Mathematical Modelling*, 39:6073–6084, 2015.

[24] M. Hajarian. Recent developments in iterative algorithms for solving linear matrix equations. In: Advances in Linear Algebra Research (I. Kyrchei, ed.), pages 239–286. Nova Science Publishers, New York, 2015.

[25] H. He, C. Ling, L. Qi, and G. Zhou. A globally and quadratically convergent algorithm for solving multilinear systems with $\mathcal{M}$-tensors. *Journal of Scientific Computing*, 76:1718–1741, 2018.

[26] J. Ji and Y. Wei. Weighted Moore-Penrose inverses and fundamental theorem of even-order tensors with Einstein product. *Frontiers of Mathematics in China*, 12:1319–1337, 2017.

[27] P. M. Kroonenberg. *Three-Mode Principal Component Analysis: Theory and Applications*. DSWO Press, Leiden, The Netherlands, 1983.

[28] W. M. Lai, D. Rubin, and E. Krempl. *Introduction to Continuum Mechanics*. Oxford, Butterworth-Heinemann, fourth edition, 2010.

[29] L. Liang and B. Zheng. Sensitivity analysis of the Lyapunov tensor equation. *Linear and Multilinear Algebra*, 67:555–572, 2019.

[30] M. Najafi-Kalyani, F. P. A. Beik, and K. Jbilou. On global iterative schemes based on Hessenberg process for (ill-posed) Sylvester tensor equations. *Journal of Computational and Applied Mathematics*, 373:e112216, 2020.

[31] A. Nouy. Low-rank methods for high-dimensional approximation and model order reduction. in: P. Benner, M. Ohlberger, A. Cohen, K. Willcox (Eds.), Model Reduction and Approximation, pages 171–226. Society for Industrial and Applied Mathematics, 2017.

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

[33] C. Paige and M. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8:43–47, 1982.

[34] L. Qi and Z. Luo. *Tensor Analysis: Spectral Theory and Special Tensors*. SIAM, 2017.

[35] L. Qi, Y. Wang, and E. X. Wu. D-eigenvalues of diffusion kurtosis tensors. *Journal of Computational and Applied Mathematics*, 221:150–157, 2008.

[36] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelpha, 2nd edition, 2003.

[37] X. Shi, Y. Wei, and S. Ling. Backward error and perturbation bounds for high order Sylvester tensor equation. *Linear and Multilinear Algebra*, 61:1436–1446, 2013.

[38] A. Smilde, R. Bro, and P. Geladi. *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley & Sons, New York, 2004.

[39] L. Sun, B. Zheng, C. Bu, and Y. Wei. Moore Penrose inverse of tensors via Einstein product. *Linear and Multilinear Algebra*, 64:686–698, 2016.

[40] H. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, New York, 2003.

[41] Q.-W. Wang and X. Xu. Iterative algorithms for solving some tensor equations. *Linear and Multilinear Algebra*, 67:1325–1349, 2019.

[42] X. Wang, M. Che, L. Qi, and Y. Wei. Modied gradient dynamic approach to the tensor complementarity problem. *Optimization Methods and Software*, 35:394–415, 2020.

[43] X. Wang, M. Che, and Y. Wei. Neural networks based approach solving multi-linear systems with M-tensors. *Neurocomputing*, 351:33–42, 2019.

[44] X. Wang, M. Che, and Y. Wei. Neural network approach for solving nonsingular multi-linear tensor systems. *Journal of Computational and Applied Mathematics*, 368:e112569, 2020.

[45] Z. Xie, X. Jin, and Y. Wei. Tensor methods for solving symmetric M-tensor systems. *Journal of Scientific Computing*, 74:412–425, 2018.

[46] X. Xu and Q.-W. Wang. Extending BiCG and BiCR methods to solve the Stein tensor equation. *Computers and Mathematics with Applications*, 77:3117–3127, 2019.