



**HAL**  
open science

# Tensor Krylov subspace methods via the Einstein product with applications to image and video processing

M. El Guide, A. El Ichi, Khalide Jbilou, F.P.A. Beik

## ► To cite this version:

M. El Guide, A. El Ichi, Khalide Jbilou, F.P.A. Beik. Tensor Krylov subspace methods via the Einstein product with applications to image and video processing. *Applied Numerical Mathematics*, 2022, 181, pp.347-363. 10.1016/j.apnum.2022.06.010 . hal-04413480

**HAL Id: hal-04413480**

**<https://hal.science/hal-04413480>**

Submitted on 22 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



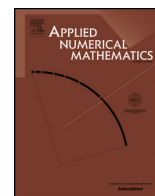
Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Applied Numerical Mathematics

[www.elsevier.com/locate/apnum](http://www.elsevier.com/locate/apnum)



# Tensor Krylov subspace methods via the Einstein product with applications to image and video processing



M. El Guide<sup>a</sup>, A. El Ichi<sup>b</sup>, K. Jbilou<sup>c,d,\*</sup>, F.P.A. Beik<sup>e</sup>

<sup>a</sup> Africa Institute for Research in Economics and Social Sciences (AIRESS), FGSES, Mohammed VI Polytechnic University, Rabat, Morocco

<sup>b</sup> Université du Littoral, Côte d'Opale, bâtiment H. Poincaré, 50 rue F. Buisson, F-62280 Calais Cedex, France

<sup>c</sup> Laboratory LMPA, 50 Rue F. Buisson, ULCO Calais Cedex, France

<sup>d</sup> University UM6P, Benguerir, Morocco

<sup>e</sup> Department of Mathematics, Vali-e-Asr University of Rafsanjan, P.O. Box 518, Rafsanjan, Iran

## ARTICLE INFO

### Article history:

Received 21 December 2021

Received in revised form 13 June 2022

Accepted 16 June 2022

Available online 22 June 2022

### Keywords:

Arnoldi process

Golub–Kahan

Ill-posed problem bidiagonalization

Tensor equation

Einstein product

Image-video processing

## ABSTRACT

In the present paper, we are interested in developing iterative Krylov subspace methods in tensor structure to solve a class of multilinear systems via the Einstein product. In particular, we develop tensor variants of the GMRES and Golub–Kahan bidiagonalization processes in tensor framework. We further consider the case that mentioned equation may be possibly corresponds to a discrete ill-posed multidimensional problem. Applications arising from color image and video restoration are included.

© 2022 Published by Elsevier B.V. on behalf of IMACS.

## 1. Introduction

In this paper, we are interested in approximating the solution of the following linear tensor equation

$$\min \|\Phi(\mathcal{X}) - \mathcal{C}\|_F, \quad (1)$$

where  $\Phi$  is a linear tensor mapping with  $\mathcal{C} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  and  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  is an unknown tensor to be determined. The linear mapping  $\Phi$  should be

$$\phi(X) = \mathcal{A} *_N \mathcal{X}, \text{ or } \phi(X) = \mathcal{A} *_N \mathcal{X} *_M \mathcal{B},$$

where  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times I_1 \times \dots \times I_N}$ ,  $\mathcal{B} \in \mathbb{R}^{J_1 \times \dots \times J_M \times J_1 \times \dots \times J_M}$ , the notations  $*_N$  and  $*_M$  represent the Einstein product along  $N$  and  $M$  modes which be defined later.

Tensor equations arise in many applications of modern sciences, e.g., engineering [27], signal processing [24], data mining [26], tensor complementarity problems, computer vision, see [10,23,28] for more details. The most recent tensor approaches used for numerically solving PDEs have been investigated in [11]. For those applications, one has to take advantage of the multidimensional structure to build rapid and robust methods for solving the related problems. For an extensive literature

\* Corresponding author at: Laboratory LMPA, 50 Rue F. Buisson, ULCO Calais Cedex, France.

E-mail address: [jbilou@univ-littoral.fr](mailto:jbilou@univ-littoral.fr) (K. Jbilou).

on tensors, one can see for example the papers in [21,22]. Over the last years, many specialized methods for solving tensor problems of type (1) have been developed, see e.g. [19] for tensor forms of the Arnoldi and Lanczos processes for well-posed problems. Huang et al. [19] pointed out that tensor equations of the form (1) appear in continuum physics, engineering, isotropic and anisotropic elastic models. Multilinear systems of the form (1) may also arise from discretization of the high-dimensional Poisson problem using finite difference approximations [3,19].

In the current paper, we are interested in developing robust and fast iterative Krylov subspace methods via the Einstein product to solve ill-posed problems originating from color image and video processing applications. Standard and global Krylov subspace methods are suitable when dealing with grayscale images, e.g., [1,2,8,9], while Krylov subspace methods can handle similar applications when the blurring linear operator can be decomposed into a Kronecker product of two matrices; see [1,2]. However, much work has to be done to numerically solve problems related to multichannel images (e.g. color images, hyper-spectral images and videos). We show that modeling these problems in the form of tensor equations (1) makes it possible to develop tensor iterative Krylov subspace methods and allows to significantly reduce the overall computational cost. We find that solving this kind of problems with the Einstein product can keep the computational cost low and the iterative methods based on this product are better than the matrix-based iterative methods. This is because, first, without vectorization, the solution and the different types of point spread function (PSF) have a one-to-one correspondence so that sophisticated and complicated boundary conditions are easily integrated in the multilinear system. Second, the tensor structure allows the application of tensor Krylov subspace methods. This method replaces the evaluation of the matrix-vector products of the standard Krylov subspace methods by evaluation of tensor-tensor products, which can be executed efficiently on many modern computers. It is also worth noting that tensor iterative Krylov subspace methods, based on T-product, have been proposed to solve large multilinear tensor equations, e.g., [12,13,8,9]. There are some advantages in computing in an Einstein tensor structured domain rather than the T-product. The PSF tensor preserves the low bandwidth since the main nodal points sit on the tensor diagonal entries and the rest of the stencil points lie on the off-diagonal positions. Although the PSF matrices used in [12,13] to build the blurring tensor model are banded, the blurring matrices in higher dimensions have larger bandwidths.

The remainder of paper is organized as follows: In Section 2, we present some symbols and notations used throughout the paper. Section 3 includes reviewing the implementation of Tikhonov regularization for tensor equation (1). Furthermore, we apply the global forms of GMRES and Golub–Kahan methods via the Einstein product in conjunction with Tikhonov regularization. We present the block tensor version of the algorithms in Section 4. Basically, the block tensor forms of the Arnoldi process, the GMRES and the Golub–Kahan algorithms are developed and their properties are studied. Section 5 deals with reporting the results corresponding to the proposed algorithms for solving numerical examples on restoring blurred and noisy color images and videos. Concluding remarks can be found in section 6.

## 2. Definitions and notations

In this section, we give a brief overview of some basic concepts and present some notions being exploited in the rest of the paper. A tensor is a multidimensional array of data and the order is the number of its indices, which is called modes or ways. Tensors can be seen as a natural extension of scalars, vectors and matrices to the higher order. Indeed, a scalar is a zero order tensor, a vector is a first order tensor and a matrix can be regarded as a tensor of order two. Throughout this work, vectors and matrices are respectively denoted by lowercase and capital letters, and tensors of higher order are represented by calligraphic letters. We first recall the definition of the  $n$ -mode tensor product with a matrix; for more details, see [22].

**Definition 1.** The  $n$ -mode product of the tensor  $\mathcal{A} = [a_{i_1 i_2 \dots i_n}] \in \mathbb{R}^{I_1 \times \dots \times I_N}$  and the matrix  $U = [u_{j_i}] \in \mathbb{R}^{J \times I_n}$  denoted by  $\mathcal{A} \times_n U$ , is a tensor of order  $I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N$  with entries defined by

$$(\mathcal{A} \times_n U)_{i_1 i_2 \dots i_{n-1} j_{i_n} \dots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 i_2 \dots i_n} u_{j_{i_n}}.$$

The  $n$ -mode product of the tensor  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  with the vector  $v = [v_{i_n}] \in \mathbb{R}^{I_n}$  is an  $(N - 1)$ -mode tensor denoted by  $\mathcal{A} \bar{\times} v$  whose elements are given by

$$(\mathcal{A} \bar{\times} v)_{i_1 \dots i_{n-1} i_{n+1} \dots i_N} = \sum_{i_n} x_{i_1 i_2 \dots i_n} v_{i_n}.$$

In what follows, we recall the definition and some properties of the tensor Einstein product which is an extension of the matrix product; for more details see [3]

**Definition 2.** Let  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_L \times K_1 \times \dots \times K_N}$ ,  $\mathcal{B} \in \mathbb{R}^{K_1 \times \dots \times K_N \times J_1 \times \dots \times J_M}$ , the Einstein product of the tensors  $\mathcal{A}$  and  $\mathcal{B}$  is the tensor of size  $\mathbb{R}^{I_1 \times \dots \times I_L \times J_1 \times \dots \times J_M}$  whose elements are defined by

$$(\mathcal{A} *_N \mathcal{B})_{i_1 \dots i_L j_1 \dots j_M} = \sum_{k_1, \dots, k_N} a_{i_1 \dots i_L k_1 \dots k_N} b_{k_1 \dots k_N j_1 \dots j_M}.$$

Here we comment that for a given tensor  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ , the transpose of  $\mathcal{A}$ , denoted by  $\mathcal{A}^T$ , is the tensor whose elements are  $b_{i_1 \dots i_M j_1 \dots j_M} = a_{j_1 \dots j_N i_1 \dots i_M}$ .

**Definition 3.** The definition of the scalar product between two tensors and its induced norm are given as follows. In particular, we state the inner product and norms with respect to the Einstein product.

1. The inner product of two same size tensors  $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  is defined by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}.$$

Notice that for even order tensors  $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ , we have

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \text{tr}(\mathcal{X}^T *_N \mathcal{Y})$$

where  $\mathcal{Y}^T \in \mathbb{R}^{J_1 \times \dots \times J_M \times I_1 \times \dots \times I_N}$  denote the transpose of  $\mathcal{Y}$ . The corresponding Frobenius norm  $\mathcal{X}$  is given by

$$\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle = \sqrt{\text{tr}(\mathcal{X}^T *_N \mathcal{X})}, \tag{2}$$

where the trace of an even-order tensor  $\mathcal{Z} \in \mathbb{R}^{I_1 \times \dots \times I_N \times I_1 \times \dots \times I_N}$  is given by

$$\text{tr}(\mathcal{Z}) = \sum_{i_1 \dots i_N} z_{i_1 \dots i_N i_1 \dots i_N}.$$

Notice that the two tensors  $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  are orthogonal if and only if  $\langle \mathcal{X}, \mathcal{Y} \rangle = 0$ .

In [4], the  $\boxtimes^N$  product between the  $N$ -mode tensors  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_{N-1} \times I_N}$  and  $\mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_{N-1} \times \tilde{I}_N}$  has been defined as an  $I_N \times \tilde{I}_N$  matrix whose  $(i, j)$ -th entry is

$$[\mathcal{X} \boxtimes^N \mathcal{Y}]_{ij} = \text{tr}(\mathcal{X}_{\dots i} \boxtimes^{N-1} \mathcal{Y}_{\dots j}), \quad N = 3, 4, \dots,$$

where

$$\mathcal{X} \boxtimes^2 \mathcal{Y} = \mathcal{X}^T \mathcal{Y}, \quad \mathcal{X} \in \mathbb{R}^{I_1 \times I_2}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \tilde{I}_2}.$$

Basically, the product  $\mathcal{X} \boxtimes^N \mathcal{Y}$  is the contracted product of the  $N$ -mode tensors  $\mathcal{X}$  and  $\mathcal{Y}$  along the first  $N - 1$  modes. It is immediate to see that for  $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ , we have

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \text{tr}(\mathcal{X} \boxtimes^N \mathcal{Y}), \quad N = 2, 3, \dots,$$

and

$$\|\mathcal{X}\|^2 = \text{tr}(\mathcal{X} \boxtimes^N \mathcal{X}) = \mathcal{X} \boxtimes^{(N+1)} \mathcal{X}.$$

We end the current subsection by recalling the following useful proposition from [4].

**Proposition 1.** Suppose that  $\mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_N \times m}$  is an  $(N + 1)$ -mode tensor with the column tensors  $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_m \in \mathbb{R}^{I_1 \times \dots \times I_N}$  and  $z = (z_1, z_2, \dots, z_m)^T \in \mathbb{R}^m$ . For an arbitrary  $(N + 1)$ -mode tensor  $\mathcal{A}$  with  $N$ -mode column tensors  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m$ , the following statement holds

$$\mathcal{A} \boxtimes^{(N+1)} (\mathcal{B} \bar{\times}_{N+1} z) = (\mathcal{A} \boxtimes^{(N+1)} \mathcal{B})z. \tag{3}$$

### 3. Krylov subspace methods via Einstein product and Tikhonov regularization

In this section, we define the tensor global Arnoldi generalizing the classical one [20] and propose iterative methods based on Global Arnoldi and Global Golub–Kahan bidiagonalization (GGKB) combined with Tikhonov regularization that are applied to the restoration of color images and videos from an available blur and noise-contaminated versions.

### 3.1. Tikhonov regularization

Many applications require the solution of several ill-conditioning systems of equations of the form (1) with a right hand side contaminated by an additive error,

$$\Phi(\mathcal{X}) = \mathcal{C} + \mathcal{E}, \tag{4}$$

where  $\mathcal{E}$  is the tensor of error terms that may stem from measurement and discretization errors. An ill-posed tensor equation may appear in color image restoration, video restoration, and when solving some partial differential equations in several space dimensions. In order to diminish the effect of the noise in the data, we replace the original problem by a regularized one. The most popular regularization method is due to Tikhonov [30] for which the original problem is replaced by the new one

$$\mathcal{X}_\mu = \operatorname{argmin}_{\mathcal{X}} \left( \|\Phi(\mathcal{X}) - \mathcal{C}\|_F^2 + \mu \|\mathcal{X}\|_F^2 \right). \tag{5}$$

The choice of the parameter  $\mu$  affects how sensitive  $\mathcal{X}_\mu$  is to the error  $\mathcal{E}$  in the contaminated right-hand side. For the vector and matrix cases, many techniques for choosing a suitable value of  $\mu$  were examined in the literature; for instance see [7,9,31] and references therein.

### 3.2. The GMRES method via Einstein product

Let  $\Phi$  be a linear mapping from the tensor space  $\mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  onto  $\mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ . The  $m$ -th tensor Krylov subspace is defined by

$$\mathcal{K}_m(\Phi, \mathcal{V}) = \operatorname{span}\{\mathcal{V}, \Phi(\mathcal{V}), \dots, \Phi^{m-1}(\mathcal{V})\} \subseteq \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M} \tag{6}$$

where  $\Phi^i(\mathcal{V}) = \Phi(\Phi^{i-1}(\mathcal{V}))$  for  $i = 1, 2, 3, \dots$ , where  $\Phi^0(\mathcal{V}) = \mathcal{V}$ . The process for constructing an orthonormal basis of (6) is given in Algorithm 1.

---

#### Algorithm 1 Global Arnoldi process.

---

1. Inputs: A linear mapping  $\Phi$ , and a tensor  $\mathcal{V} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  and an integer  $m$ .
  2. Set  $\beta = \|\mathcal{V}\|_F$  and  $\mathcal{V}_1 = \mathcal{V}/\beta$ .
  3. For  $j = 1, \dots, m$ 
    - (a)  $\mathcal{W} = \Phi(\mathcal{V}_j)$
    - (b) for  $i = 1, \dots, j$ 
      - $h_{ij} = \langle \mathcal{V}_i, \mathcal{W} \rangle$ ,
      - $\mathcal{W} = \mathcal{W} - h_{ij}\mathcal{V}_i$
    - (c) endfor
    - (d)  $h_{j+1,j} = \|\mathcal{W}\|_F$ . If  $h_{j+1,j} = 0$ , stop; else
    - (e)  $\mathcal{V}_{j+1} = \mathcal{W}/h_{j+1,j}$ .
  4. EndFor
- 

Let  $\tilde{H}_m$  be the upper Hessenberg matrix of size  $(m + 1) \times m$  whose entries are the  $h_{ij}$  from Algorithm 1 and let  $H_m$  be the matrix obtained from  $\tilde{H}_m$  by deleting the last row. Then, it is not difficult to verify that  $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_m$  obtained from Algorithm 1 form an orthonormal basis of the tensor Krylov subspace  $\mathcal{K}_m(\Phi, \mathcal{V})$ . Analogous to [4,20], we can prove the following proposition.

**Proposition 2.** Let  $\mathbb{V}$  be the  $(M + N + 1)$ -mode tensor with frontal slices  $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_m$  and  $\mathbb{W}_m$  be the  $(M + N + 1)$ -mode tensor with frontal slices  $\Phi(\mathcal{V}_1), \dots, \Phi(\mathcal{V}_m)$ . Then

$$\begin{aligned} \mathbb{W}_m &= \mathbb{V}_{m+1} \times_{(M+N+1)} \tilde{H}_m^T \\ &= \mathbb{V}_m \times_{(M+N+1)} H_m^T + h_{m+1,m} \mathcal{L}_m \times_{(M+N+1)} E_m, \end{aligned} \tag{7}$$

where  $E_m = [0, 0, \dots, 0, e_m]$ . Here, the vector  $e_m$  is the  $m$ -th column of the identity matrix  $I_m$  and  $\mathcal{L}_m$  is the  $(M + N + 1)$ -mode whose frontal slices are all zero except that last one being equal to  $\mathcal{V}_{m+1}$ .

Let  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ ,  $\Phi$  a linear tensor mapping,  $\mathcal{C} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  and consider the following tensor equation

$$\Phi(\mathcal{X}) = \mathcal{C}. \tag{8}$$

Using Algorithm 1, we can propose the tensor global GMRES method to solve the problem (8). As for the global GMRES, we seek for an approximate solution  $\mathcal{X}_m$ , starting from  $\mathcal{X}_0$  such that  $\mathcal{X}_m \in \mathcal{X}_0 + \mathcal{K}_m(\Phi, \mathcal{V})$  and by solving the minimization problem

$$\|\mathcal{R}_m\|_F = \min_{\mathcal{X} \in \mathcal{X}_0 + \mathcal{K}_m(\Phi, \mathcal{V})} \|\mathcal{C} - \Phi(\mathcal{X})\|_F, \tag{9}$$

where  $\mathcal{R}_m = \mathcal{C} - \Phi(\mathcal{X}_m)$ . Assume that  $m$  steps of Algorithm 1 have been performed. Given an initial guess  $\mathcal{X}_0$ , we set

$$\mathcal{X}_m = \mathcal{X}_0 + \mathbb{V}_m \bar{\times}_{(M+N+1)} y_m, \tag{10}$$

which gives  $\mathcal{R}_m = \mathcal{R}_0 - \mathbb{W}_m \bar{\times}_{(M+N+1)} y_m$ . In view of Proposition 1 and using the relations (7) from Proposition 2, it follows that

$$\begin{aligned} \|\mathcal{C} - \Phi(\mathcal{X}_m)\|_F &= \|\mathbb{V}_m \boxtimes^{(M+N+1)} (\mathcal{C} - \Phi(\mathcal{X}_m))\|_2 \\ &= \|\mathbb{V}_m \boxtimes^{(M+N+1)} (\mathcal{R}_0 - \mathbb{W}_m \bar{\times}_{(M+N+1)} y_m)\|_2 \\ &= \|\beta e_1^{m+1} - \mathbb{V}_m \boxtimes^{(M+N+1)} (\mathbb{W}_m \bar{\times}_{(M+N+1)} y_m)\|_2 \\ &= \|\beta e_1^{m+1} - (\mathbb{V}_m \boxtimes^{(M+N+1)} \mathbb{W}_m) y_m\|_2 \end{aligned}$$

which shows that the unknown vector  $y_m$  is determined as follows:

$$y_m = \arg \min_y \|\beta e_1^{m+1} - \tilde{H}_m y\|_2. \tag{11}$$

The relations (10) and (11) define the tensor global GMRES (TG-GMRES). Setting  $\mathcal{X}_0 = 0$  and using the relations (9), (10) and (11) it follows that instead of solving the problem (5) we can consider the following low dimensional Tikhonov regularization problem

$$\|\beta e_1^{m+1} - \tilde{H}_m y\|_2^2 + \mu \|y\|_2^2. \tag{12}$$

The solution of the problem (12) is given by

$$y_{m,\mu} = \arg \min \left\| \begin{pmatrix} \tilde{H}_m \\ \sqrt{\mu} I \end{pmatrix} y - \begin{pmatrix} \beta e_1^{m+1} \\ 0 \end{pmatrix} \right\|_2. \tag{13}$$

The minimizer  $y_{m,\mu}$  of the problem (13) is computed as the solution of the linear system of equations

$$\tilde{H}_{m,\mu} y = \tilde{H}_m^T \beta e_1^{m+1} \tag{14}$$

where  $\tilde{H}_{m,\mu} = (\tilde{H}_m^T \tilde{H}_m + \mu I)$ . However, solving (13) is numerically better than solving the (14).

Notice that the Tikhonov problem (12) is a matrix one with small dimension as  $m$  is generally small. Hence it can be solved by some techniques such as the GCV method [15] or the L-curve criterion [16,17,8,9]. An appropriate selection of the regularization parameter  $\mu$  is important in Tikhonov regularization. Here we can use the generalized cross-validation (GCV) method [6,15,31]. For this method, the regularization parameter is chosen to minimize the GCV function

$$GCV(\mu) = \frac{\|\tilde{H}_m y_{m,\mu} - \beta e_1^{m+1}\|_2^2}{[tr(I - \tilde{H}_m \tilde{H}_{m,\mu}^{-1} \tilde{H}_m^T)]^2} = \frac{\|(I - \tilde{H}_m \tilde{H}_{m,\mu}^{-1} \tilde{H}_m^T) \beta e_1^{m+1}\|_2^2}{[tr(I - H_m H_{m,\mu}^{-1} \tilde{H}_m^T)]^2}.$$

As the projected problem we are dealing with is of small size, we can use the SVD decomposition of  $\tilde{H}_m$  to obtain a more simple and computable expression of  $GCV(\mu)$ . Consider the SVD decomposition of  $\tilde{H}_m = U \Sigma V^T$ . Then the GCV function can be expressed as (see [31])

$$GCV(\mu) = \frac{\sum_{i=1}^m \left(\frac{\tilde{g}_i}{\sigma_i^2 + \mu}\right)^2}{\left(\sum_{i=1}^m \frac{1}{\sigma_i^2 + \mu}\right)^2}, \tag{15}$$

where  $\sigma_i$  is the  $i$ th singular value of the matrix  $\tilde{H}_m$  and  $\tilde{g} = \beta_1 U^T e_1^{m+1}$ .

As the number of outer iterations increases, it is possible to compute the  $m$ -th residual without forming the solution. This is described in the following theorem.

**Proposition 3.** *At step  $m$ , the residual  $\mathcal{R}_m = \mathcal{C} - \Phi(\mathcal{X}_m)$  produced by the tensor global GMRES method for solving (1) has the following expression*

$$\mathcal{R}_m = \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (\gamma_{m+1} Q_m e_{m+1}), \tag{16}$$

where  $Q_m$  is the unitary matrix obtained by QR decomposition of the upper Hessenberg matrix  $\tilde{H}_m$  and  $\gamma_{m+1}$  is the last component of the vector  $\beta Q_m^T e_{m+1}$  in which  $\beta = \|\mathcal{R}_0\|_F$  and  $e_\ell \in \mathbb{R}^\ell$  is the last column of identity matrix. Furthermore,

$$\|\mathcal{R}_m\|_F = |\gamma_{m+1}| \tag{17}$$

**Proof.** At step  $m$ , the residual  $\mathcal{R}_m = \mathcal{R}_0 - \mathbb{W}_m \bar{\times}_{(M+N+1)} y_m$  can be expressed as

$$\begin{aligned} \mathcal{R}_m &= \mathcal{R}_0 - (\mathbb{V}_{m+1} \times_{(M+N+1)} \tilde{H}_m^T) \bar{\times}_{(M+N+1)} y_m \\ &= \mathcal{R}_0 - \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (\tilde{H}_m y_m) \end{aligned}$$

by considering the QR decomposition  $\tilde{H}_m = Q_m \tilde{U}_m$  of the  $(m + 1) \times m$  matrix  $\tilde{H}_m$ , we get

$$\mathcal{R}_m = \mathcal{R}_0 - \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (Q_m \tilde{U}_m y_m).$$

Using Proposition 1 and straightforward computations show that

$$\begin{aligned} \|\mathcal{R}_m\|_F^2 &= \|\mathcal{R}_0 - \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (Q_m \tilde{U}_m y_m)\|_F^2 \\ &= \|\mathbb{V}_m \boxtimes^{(M+N+1)} (\mathcal{R}_0 - \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (Q_m \tilde{U}_m y_m))\|_2^2 \\ &= \|Q_m (Q_m^T (\beta e_1^{m+1}) - \tilde{U}_m y_m)\|_2^2 \\ &= \|\beta Q_m^T e_1^{m+1} - \tilde{U}_m y_m\|_2^2 \\ &= \|z_m - \tilde{U}_m y_m\|_2^2 + |\gamma_{m+1}|^2 \end{aligned}$$

where  $z_m$  denotes vector obtained by deleting the last component of  $Q_m^T (\beta e_1^{m+1})$ . Since  $y_m$  solves problem (11), it follows that  $y_m$  is the solution of  $\tilde{U}_m y_m = z_m$ , i.e.,

$$\|z_m - \tilde{U}_m y_m\|_2 = 0.$$

Note that  $\mathcal{R}_m$  can be written in the following form

$$\begin{aligned} \mathcal{R}_m &= \beta \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} e_1^{m+1} - \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (\tilde{H}_m y_m) \\ &= \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (\beta e_1^{m+1} - \tilde{H}_m y_m) \\ &= \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (Q_m (Q_m^T (\beta e_1^{m+1}) - \tilde{U}_m y_m)) \\ &= \mathbb{V}_{m+1} \bar{\times}_{(M+N+1)} (Q_m \gamma_{m+1} e_{m+1}). \end{aligned}$$

Now the result follows immediately from the above computations.  $\square$

The tensor global GMRES algorithm with the Tikhonov regularization for the projected problem, is summarized as follows:

---

**Algorithm 2** Global GMRES method via Einstein product.

---

1. **Inputs** A linear mapping  $\Phi$ , initial guess  $\mathcal{X}_0$ , a tolerance  $\varepsilon$ , number of iterations between restarts  $m$  and *Maxit* a maximum number of outer iterations.
  2. Compute  $\mathcal{R}_0 = \mathcal{C} - \Phi(\mathcal{X}_0)$ , set  $\mathcal{V} = \mathcal{R}_0$  and  $k = 0$
  3. Determine the orthonormal frontal slices  $\mathcal{V}_1, \dots, \mathcal{V}_m$  of  $\mathbb{V}_m$ , and the upper Hessenberg matrix  $\tilde{H}_m$  by applying Algorithm 1 to the pair  $(\Phi, \mathcal{V})$ .
  4. Determine  $\mu_k$  as the parameter minimizing the GCV function given by (15)
  5. Determine  $y_m$  as the solution of low-dimensional Tikhonov regularization problem (12) and compute  $\mathcal{R}_m$
  6. If  $\|\mathcal{R}_m\|_F < \varepsilon$  or  $k > \text{Maxit}$ ; Set  $\mathcal{X}_m = \mathcal{X}_0 + \mathbb{V}_m \bar{\times}_{(M+N+1)} y_m$  and stop.  
 else: Set  $\mathcal{V} = \mathcal{R}_m$ ,  $k = k + 1$  and go to Step 3.
- 

3.3. The Golub–Kahan method via the Einstein product

Instead of projecting orthogonally onto a Krylov subspace and using GMRES method for solving the obtained minimization problem, one can apply others schemes based on the Krylov subspaces  $\mathcal{K}_m(\Phi, \mathcal{V})$  and  $\mathcal{K}_m(\Phi^T, \mathcal{W})$ .

Here, we exploit the tensor Golub–Kahan algorithm via the Einstein product. It should be commented here that the Golub–Kahan algorithm has been already examined for solving ill-posed Sylvester and Lyapunov tensor equations with applications to color image restoration [5] using the classical  $n$ -mode product.

Let  $\mathcal{C} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  be given tensor. Then, the global Golub–Kahan bidiagonalization (GGKB) algorithm associated to the linear operator  $\Phi$  and the initial tensor  $\mathcal{C}$  is summarized as follows.

**Algorithm 3** Global Golub–Kahan algorithm.

1. **Inputs** A linear mapping  $\Phi$  and an integer  $\ell$ .
2. Set  $\sigma_1 = \|C\|_F$ ,  $\mathcal{U}_1 = C/\sigma_1$  and  $\mathcal{V}_0 = 0$
3. For  $j = 1, 2, \dots, \ell$  Do
4.  $\tilde{\mathcal{V}} = \Phi^T(\mathcal{U}_j) - \sigma_j \mathcal{V}_{j-1}$
5.  $\rho_j = \|\tilde{\mathcal{V}}\|_F$  if  $\rho_j = 0$  stop, else
6.  $\mathcal{V}_j = \tilde{\mathcal{V}}/\rho_j$
7.  $\tilde{\mathcal{U}} = \Phi(\mathcal{V}_j) - \rho_j \mathcal{U}_j$
8.  $\sigma_{j+1} = \|\tilde{\mathcal{U}}\|_F$
9. if  $\rho_j = 0$  stop, else
10.  $\mathcal{U}_{j+1} = \tilde{\mathcal{U}}/\sigma_{j+1}$
11. EndDo

Assume that  $\ell$  steps of the GGKB process have been performed, we form the lower bidiagonal matrix  $C_\ell \in \mathbb{R}^{\ell \times \ell}$

$$C_\ell = \begin{bmatrix} \rho_1 & & & & & \\ \sigma_2 & \rho_2 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \sigma_{\ell-1} & \rho_{\ell-1} \\ & & & & & \sigma_\ell & \rho_\ell \end{bmatrix}$$

and

$$\tilde{C}_\ell = \begin{bmatrix} C_\ell \\ \sigma_{\ell+1} e_\ell^T \end{bmatrix} \in \mathbb{R}^{(\ell+1) \times \ell}.$$

**Proposition 4.** Assume that  $\ell$  steps of Algorithm 3 have performed and all non-trivial entries of the matrix  $\tilde{C}_\ell$  are positive. Let  $\mathbb{V}_\tau$  and  $\mathbb{U}_\tau$  be  $(M + N + 1)$ -mode tensors whose frontal slices are given by  $\mathcal{V}_j$  and  $\mathcal{U}_j$  for  $j = 1, 2, \dots, \tau$ , respectively. Furthermore, let  $\mathbb{W}_\tau$  and  $\tilde{\mathbb{W}}_\tau$  be the  $(M + N + 1)$ -mode tensors having frontal slices  $\Phi(\mathcal{V}_j)$  and  $\Phi^T(\mathcal{U}_j)$  for  $j = 1, 2, \dots, \tau$ , respectively. The following relations hold:

$$\mathbb{W}_\ell = \mathbb{U}_{\ell+1} \times_{(M+N+1)} \tilde{C}_\ell^T, \tag{18}$$

$$\tilde{\mathbb{W}}_\ell = \mathbb{V}_\ell \times_{(M+N+1)} C_\ell^T. \tag{19}$$

**Proof.** From Lines 7 and 10 of Algorithm 3, we have

$$\Phi(\mathcal{V}_j) = \rho_j \mathcal{U}_j + \sigma_{j+1} \mathcal{U}_{j+1} \quad j = 1, 2, \dots, \ell$$

which conclude (18) from definition of  $n$ -mode product. Similarly, Eq. (19) follows from Lines 4 and 6 of Algorithm 3.  $\square$

Here, we apply the following Tikhonov regularization approach and solve the new problem

$$\min_{\mathcal{X}} \left( \|\Phi(\mathcal{X}) - C\|_F^2 + \mu^{-1} \|\mathcal{X}\|_F^2 \right), \tag{20}$$

We comment on the use of  $\mu^{-1}$  in (20) instead of  $\mu$  below. As for the iterative tensor Global GMRES method discussed in the previous subsection, the computation of an accurate approximation  $\mathcal{X}_\mu$  requires that a suitable value of the regularization parameter be used. In this subsection, we use the discrepancy principle to determine a suitable regularization parameter assuming that an approximation of the norm of additive error is available, i.e., we have a bound  $\epsilon$  for  $\|\mathcal{E}\|_F$ . This priori information suggests that  $\mu$  has to be determined such that,

$$\|C - \Phi(\mathcal{X}_\mu)\|_F = \eta\epsilon, \tag{21}$$

where  $\eta > 1$  is the safety factor for the discrepancy principle. A zero-finder method can be used to solve (21) in order to find a suitable regularization parameter which also implies that  $\|C - \Phi(\mathcal{X}_\mu)\|_F$  has to be evaluated for several  $\mu$ -values. When the tensor  $\mathcal{A}$  is of moderate size, the quantity  $\|C - \Phi(\mathcal{X}_\mu)\|_F$  can be easily evaluated. This computation becomes expensive when  $\mathcal{A}$  is a large tensor, which means that its evaluation by a zero-finder method can be very difficult and computationally expensive. In what follows, it is shown that this difficulty can be remedied by using a connection between the Golub–Kahan bidiagonalization (GGKB) and Gauss-type quadrature rules. This connection provides approximations of moderate sizes to the quantity  $\|C - \Phi(\mathcal{X}_\mu)\|_F$  and therefore gives a solution method to inexpensively solve (21) by evaluating these small quantities; see [1,2] for discussion on this method.



Let us consider the following functions of  $\mu$ ,

$$\phi(\mu) = \|C - \Phi(\mathcal{X}_\mu)\|_F^2 \tag{22}$$

$$\mathcal{G}_\ell f_\mu = \|C\|_F^2 e_1^T (\mu C_\ell C_\ell^T + I_\ell)^{-2} e_1, \tag{23}$$

$$\mathcal{R}_{\ell+1} f_\mu = \|C\|_F^2 e_1^T (\mu \tilde{C}_\ell \tilde{C}_\ell^T + I_{\ell+1})^{-2} e_1; \tag{24}$$

$\mathcal{G}_\ell f$  and  $\mathcal{R}_{\ell+1} f_\mu$  are pairs of Gauss and Gauss-Radau quadrature rules, respectively, and they approximate  $\phi(\mu)$  as follows

$$\mathcal{G}_\ell f_\mu \leq \phi(\mu) \leq \mathcal{R}_{\ell+1} f_\mu \tag{25}$$

As shown in [1,2], for a given value of  $l \geq 2$ , we solve for  $\mu$  the nonlinear equation

$$\mathcal{G}_\ell f_\mu = \epsilon^2 \tag{26}$$

by using Newton’s method.

The use the parameter  $\mu$  in (20) instead of  $1/\mu$ , implies that the left-hand side of (21) is a decreasing convex function of  $\mu$ . Therefore, there is a unique solution, denoted by  $\mu_\epsilon$ , of

$$\phi(\mu) = \epsilon^2$$

for almost all values of  $\epsilon > 0$  of practical interest and therefore also of (26) for  $\ell$  sufficiently large; see [1,2] for analyses. We accept  $\mu_\epsilon$  that solve (21) as an approximation of  $\mu$ , whenever we have

$$\mathcal{R}_{\ell+1} f_\mu \leq \eta^2 \epsilon^2. \tag{27}$$

If (27) does not hold for  $\mu_\ell$ , we carry out one more GGKB steps, replacing  $\ell$  by  $\ell + 1$  and solve the nonlinear equation

$$\mathcal{G}_{\ell+1} f_\mu = \epsilon^2; \tag{28}$$

see [1,2] for more details. Assume now that (27) holds for some  $\mu_\ell$ . The corresponding regularized solution is then computed by

$$\mathcal{X}_\ell = \mathbb{V}_\ell \bar{\mathbb{X}}_{(M+N+1)} y_\ell, \tag{29}$$

where  $y_{\mu_\ell}$  solves

$$(\tilde{C}_\ell^T \tilde{C}_\ell + \mu_\ell^{-1} I_\ell) y = \sigma_1 \tilde{C}_\ell^T e_1, \quad \sigma_1 = \|C\|_F. \tag{30}$$

The unknown vector  $y$  can also be computed by solving the least-squares problem

$$\min_{y \in \mathbb{R}^\ell} \left\| \begin{bmatrix} \mu_\ell^{1/2} \tilde{C}_\ell \\ I_\ell \end{bmatrix} y - \sigma_1 \mu_\ell^{1/2} e_1 \right\|_2, \tag{31}$$

which is numerically better than solving (30). The following result shows an important property of the approximate solution (29). We include a proof for completeness.

**Proposition 5.** Under assumptions of Proposition 4, let  $\mu_\ell$  solve (26) and let  $y_{\mu_\ell}$  solve (31). Then the associated approximate solution (29) of (20) satisfies

$$\|\Phi(\mathcal{X}_{\mu_\ell}) - C\|_F^2 = R_{\ell+1} f_{\mu_\ell}$$

**Proof.** By Eq. (18), we have  $\Phi(\mathcal{X}_{\mu_\ell}) = \mathbb{W}_\ell \bar{\mathbb{X}}_{(M+N+1)} y_\ell = \mathbb{U}_{\ell+1} \bar{\mathbb{X}}_{(M+N+1)} (\tilde{C}_\ell y_\ell)$ . Using the above expression, we obtain

$$\begin{aligned} \|\Phi(\mathcal{X}_{\mu_\ell}) - C\|_F^2 &= \|\mathbb{U}_{\ell+1} \bar{\mathbb{X}}_{(M+N+1)} (\tilde{C}_\ell y_\ell) - \sigma_1 \mathcal{U}_1\|_F^2 \\ &= \|\mathbb{U}_{\ell+1} \bar{\mathbb{X}}_{(M+N+1)} (\tilde{C}_\ell y_\ell) - \mathbb{U}_{\ell+1} \bar{\mathbb{X}}_{(M+N+1)} (\sigma_1 e_1)\|_F^2 \\ &= \|\mathbb{U}_{\ell+1} \bar{\mathbb{X}}_{(M+N+1)} (\tilde{C}_\ell y_\ell - \sigma_1 e_1)\|_F^2 \\ &= \|\mathbb{U}_{\ell+1} \boxtimes^{(M+N+1)} (\mathbb{U}_{\ell+1} \bar{\mathbb{X}}_{(M+N+1)} (\tilde{C}_\ell y_\ell - \sigma_1 e_1))\|_F^2 \\ &= \left\| \left( \mathbb{U}_{\ell+1} \boxtimes^{(M+N+1)} \mathbb{U}_{\ell+1} \right) (\tilde{C}_\ell y_\ell - \sigma_1 e_1) \right\|_2^2 \\ &= \|\tilde{C}_\ell y_\ell - \sigma_1 e_1\|_2^2 \end{aligned}$$

recalling that  $\sigma_1 = \|\mathcal{C}\|_F$ . We now express  $y_{\mu_\ell}$  with the aid of (30) and apply the following identity

$$I - A \left( A^T A + \mu^{-1} I \right)^{-1} A^T = \left( \mu A A^T + I \right)^{-1}$$

with  $A$  replaced by  $\widehat{C}_\ell$ , to obtain

$$\begin{aligned} \|\Phi(\mathcal{X}_{\mu_\ell}) - \mathcal{C}\|_F^2 &= \sigma_1^2 \left\| e_1 - \widetilde{C}_\ell \left( \widetilde{C}_\ell^T \widetilde{C}_\ell + \mu_\ell^{-1} I_\ell \right)^{-1} \widetilde{C}_\ell^T e_1 \right\|_F^2 \\ &= \sigma_1^2 e_1^T \left( \mu_\ell \widetilde{C}_\ell \widetilde{C}_\ell^T + I_{\ell+1} \right)^{-2} e_1 \\ &= R_{\ell+1} f_{\mu_\ell} \end{aligned}$$

which conclude the assertion.  $\square$

The following algorithm summarizes the main steps to compute a regularization parameter and a corresponding regularized solution of (1) using GGKB and quadrature rules method for Tikhonov regularization.

---

**Algorithm 4** GGKB and quadrature rules method for Tikhonov regularization.

---

1. **Inputs** A linear mapping  $\Phi$ ,  $\eta \leq 1$  and  $\varepsilon$ .
  2. Determine the orthonormal bases  $\mathbb{U}_{\ell+1}$  and  $\mathbb{V}_\ell$  of tensors, and the bidiagonal matrices  $C_\ell$  and  $\widetilde{C}_\ell$  by implementing Algorithm 3.
  3. Determine  $\mu_\ell$  that satisfies (26) with Newton's method.
  4. Determine  $y_{\mu_\ell}$  by solving (31) and then compute  $X_{\mu_\ell}$  by (29).
- 

**4. Block Krylov subspace methods via Einstein**

In this section the block generalizations of the GMRES and Golub–Kahan methods via Einstein product are theoretically discussed. To this end, first, some required definitions and proprieties are given. Then, we present the block Arnoldi process in tensor framework.

4.1. Basic concepts

In this part, we briefly recall some preliminaries which are used for deriving the block versions of the algorithms, for more details see [3,25,29]. The unfolded tensor is given by the following definition which is referred to a matrix obtained by systematically reorganizing the tensor's entries into a two-dimensional array.

**Definition 4.** The transformation  $\Psi_{IJ}$  from the tensor space  $\mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  into the matrix space  $\mathbb{R}^{(I_1 \dots I_N) \times (J_1 \dots J_M)}$  with  $\Psi_{IJ}(\mathcal{X}) = X$  is defined component-wise as

$$\mathcal{X}_{i_1 \dots i_N j_1 \dots j_M} = (X_{i\text{vec}(\mathbf{i}, \mathbb{I}) i\text{vec}(\mathbf{j}, \mathbb{J})})$$

where  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ ,  $X \in \mathbb{R}^{(I_1 \dots I_N) \times (J_1 \dots J_M)}$ ,  $i\text{vec}(\mathbf{i}, \mathbb{I})$  and  $i\text{vec}(\mathbf{j}, \mathbb{J})$  are respectively two index functions corresponding to the subscript set  $\mathbf{i} := \{i_1, \dots, i_N\}$  and  $\mathbf{j} := \{j_1, \dots, j_M\}$ , i.e.,

$$i\text{vec}(\mathbf{i}, \mathbb{I}) = i_1 + \sum_{r=2}^N (i_r - 1) \prod_{u=1}^{r-1} I_u. \tag{32}$$

$$i\text{vec}(\mathbf{j}, \mathbb{J}) = j_1 + \sum_{s=2}^M (j_s - 1) \prod_{v=1}^{s-1} J_v, \tag{33}$$

and  $\mathbb{I} := \{I_1, \dots, I_N\}$ ,  $\mathbb{J} := \{J_1, \dots, J_M\}$  are respectively referred to the row mode and the column mode of  $\mathcal{X}$ .

By Definition 4, the concepts of upper/lower triangular, sub-column and row/column block for matrices can be extended for tensors using the following three definitions.

**Definition 5.** Let  $\mathcal{U}$  and  $\mathcal{R}$  belong to  $\mathbb{C}^{I_1 \times \dots \times I_N \times I_1 \times \dots \times I_N}$ .

1. The tensor  $\mathcal{U}$  is upper triangular if the entries  $u_{i_1, \dots, i_N, j_1, \dots, j_N} = 0$  when  $i\text{vec}(\mathbf{i}, \mathbb{I}) \geq i\text{vec}(\mathbf{j}, \mathbb{J})$ .
2. The tensor  $\mathcal{R}$  is lower triangular if the entries  $r_{i_1, \dots, i_N, j_1, \dots, j_N} = 0$  when  $i\text{vec}(\mathbf{i}, \mathbb{I}) \leq i\text{vec}(\mathbf{j}, \mathbb{J})$ .

**Definition 6 (Sub-column tensor).** Let  $\mathcal{A} = (a_{i_1, \dots, i_N, j_1, \dots, j_M}) \in \mathbb{C}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ . The  $j$ -th sub-column of  $\mathcal{A}$  is the tensor  $\mathcal{A}_{(\cdot, \dots, \cdot | j_1, \dots, j_M)} = (a_{\cdot, \dots, \cdot, j_1, \dots, j_M}) \in \mathbb{C}^{I_1 \times \dots \times I_N}$  where  $j = j_M + \sum_{r=1}^{M-1} \left[ (j_r - 1) \prod_{L=r+1}^M J_L \right]$ .

**Definition 7.** Let  $\mathcal{A} \in \mathbb{C}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ ,  $\mathcal{B} \in \mathbb{C}^{I_1 \times \dots \times I_N \times K_1 \times \dots \times K_M}$ ,  $\mathcal{C} \in \mathbb{C}^{J_1 \times \dots \times J_M \times I_1 \times \dots \times I_N}$  and  $\mathcal{D} \in \mathbb{C}^{K_1 \times \dots \times K_M \times I_1 \times \dots \times I_N}$ . Then, the following block tensors can be defined analogous to block matrices.

- Row block tensor:** The row block tensor of  $\mathcal{A}$  and  $\mathcal{B}$  denoted by  $[\mathcal{A} \ \mathcal{B}]$  is the tensor of size  $\mathcal{I}^N \times \beta_1 \times \dots \times \beta_M$  where  $\mathcal{I}^N = I_1 \times I_2 \times \dots \times I_N$ , and  $\beta_i = J_i + K_i$  for  $i = 1, \dots, M$ .
- Column block tensor:** The column block tensor of  $\mathcal{C}$  and  $\mathcal{D}$  is defined as

$$\begin{bmatrix} \mathcal{C} \\ \mathcal{D} \end{bmatrix} = \left[ \mathcal{C}^T \ \mathcal{D}^T \right]^T \in \mathbb{C}^{\beta_1 \times \dots \times \beta_M \times \mathcal{I}^N}$$

where  $\mathcal{I}^N = I_1 \times I_2 \times \dots \times I_N$ , and  $\beta_i = J_i + K_i$  for  $i = 1, \dots, M$ .

- Let  $\mathcal{T}_1 = [\mathcal{A}_1 \ \mathcal{B}_1]$ ,  $\mathcal{T}_2 = [\mathcal{A}_2 \ \mathcal{B}_2]$  be two row block tensors, where  $\mathcal{A}_1 \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ ,  $\mathcal{B}_1 \in \mathbb{R}^{I_1 \times \dots \times I_N \times K_1 \times \dots \times K_M}$ ,  $\mathcal{A}_2 \in \mathbb{R}^{L_1 \times \dots \times L_N \times J_1 \times \dots \times J_M}$  and  $\mathcal{B}_2 \in \mathbb{R}^{L_1 \times \dots \times L_N \times K_1 \times \dots \times K_M}$ . We define the following block tensor accordingly,

$$\begin{bmatrix} \mathcal{T}_1 \\ \mathcal{T}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{B}_1 \\ \mathcal{A}_2 & \mathcal{B}_2 \end{bmatrix} \in \mathbb{C}^{\rho_1 \times \dots \times \rho_N \times \beta_1 \times \dots \times \beta_M}$$

where  $\rho_i = I_i + L_i$ ,  $i = 1, \dots, N$  and  $\beta_j = J_j + K_j$ ,  $j = 1, \dots, M$ .

We end this subsection by the following useful proposition which provide properties of the Einstein product between block tensor, see [29] for more details.

**Proposition 6.** Let the tensors  $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{A}_i$  and  $\mathcal{B}_i$  for  $i = 1, 2$  are defined in the same way given by Definition 7 and  $\mathcal{F} \in \mathbb{C}^{\mathcal{I}^N \times \mathcal{I}^N}$  where  $\mathcal{I}^N = I_1 \times I_2 \times \dots \times I_N$  as before. Then, the following statements hold:

- $\mathcal{F} *_N [\mathcal{A} \ \mathcal{B}] = [\mathcal{F} *_N \mathcal{A} \ \mathcal{F} *_N \mathcal{B}] \in \mathbb{C}^{\mathcal{I}^N \times \beta_1 \times \dots \times \beta_M}$ .
- $\begin{bmatrix} \mathcal{C} \\ \mathcal{D} \end{bmatrix} *_N \mathcal{F} = \begin{bmatrix} \mathcal{C} *_N \mathcal{F} \\ \mathcal{D} *_N \mathcal{F} \end{bmatrix} \in \mathbb{C}^{\beta_1 \times \dots \times \beta_M \times \mathcal{I}^N}$ .
- $[\mathcal{A} \ \mathcal{B}] *_M \begin{bmatrix} \mathcal{C} \\ \mathcal{D} \end{bmatrix} = \mathcal{A} *_M \mathcal{C} + \mathcal{B} *_M \mathcal{D} \in \mathbb{C}^{\mathcal{I}^N \times \mathcal{I}^N}$ .
- $\begin{bmatrix} \mathcal{A}_1 & \mathcal{B}_1 \\ \mathcal{A}_2 & \mathcal{B}_2 \end{bmatrix} *_M \begin{bmatrix} \mathcal{C} \\ \mathcal{D} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_1 *_M \mathcal{C} + \mathcal{B}_1 *_M \mathcal{D} \\ \mathcal{A}_2 *_M \mathcal{C} + \mathcal{B}_2 *_M \mathcal{D} \end{bmatrix} \in \mathbb{C}^{\rho_1 \times \dots \times \rho_M \times \mathcal{I}^N}$ .

#### 4.2. Block Arnoldi process via the Einstein product

In this part, we present the block Arnoldi process for tensors via the Einstein product which can be seen a generalization of the well-know block Arnoldi process for matrices.

Let  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times I_1 \times \dots \times I_N}$  be a square tensor and  $\mathcal{V} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ . The  $m$ -th tensor block Krylov subspace is defined by

$$\mathcal{K}_m^{Block}(\mathcal{A}, \mathcal{V}) = \text{Range} \left( [\mathcal{V}, \mathcal{A} *_N \mathcal{V}, \dots, \mathcal{A}^{m-1} *_N \mathcal{V}] \right) \subseteq \mathbb{R}^{I_1 \times \dots \times I_N} \tag{34}$$

where  $\mathcal{A}^i = \mathcal{A} *_N \mathcal{A}^{i-1}$  for  $i = 1, \dots, m - 1$  where  $\mathcal{A}^0$  is the identity tensor and for any  $\mathcal{Z} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ , the Range of  $\mathcal{Z}$  is defined by

$$\text{Range}(\mathcal{Z}) = \left\{ \mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_N} \mid \mathcal{Y} = \mathcal{Z} *_M \mathcal{X} \text{ for some } \mathcal{X} \in \mathbb{R}^{J_1 \times \dots \times J_M} \right\}.$$

To apply the block version of the Arnoldi process via the Einstein product, we need to present QR factorization via the Einstein product in the following theorem.

**Theorem 1.** Let  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ , then there exists an orthogonal tensor  $\mathcal{Q} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  ( $\mathcal{Q}^T *_N \mathcal{Q} = \mathcal{I}_M$ ) and the upper triangular tensor  $\mathcal{R} \in \mathbb{R}^{J_1 \times \dots \times J_M \times J_1 \times \dots \times J_M}$  such that

$$\mathcal{A} = \mathcal{Q} *_M \mathcal{R}. \tag{35}$$

The decomposition (35) is called the QR factorization of the tensor  $\mathcal{A}$ .

**Proof.** Let  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ , assume that the QR factorization of the unfolding matrix  $A = \Psi_{IJ}(\mathcal{A})$  is:  $A = QR$ , we have  $A = QR$ . Since  $\Psi$  is a bijection, we get

$$\begin{aligned} \mathcal{A} &= \Psi_{IJ}^{-1}(A) = \Psi_{IJ}^{-1}(QR) \\ &= \mathcal{Q} *_M \mathcal{R} \quad \square \end{aligned}$$

The algorithm for constructing an orthonormal basis of the tensor block Krylov subspace (34) is given by Algorithm 5. Notice that in the case  $N = M = 1$ , Algorithm 5 reduces to the well known block Arnoldi process.

---

**Algorithm 5** Bock Arnoldi process via the Einstein product.

---

1. Inputs: A tensor  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ , a tensor  $\mathcal{V} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  and the integer  $m$ .
  2. Set  $\mathcal{V} = \mathcal{Q} *_M \mathcal{R}$ ,  $\mathcal{V}_1^b = \mathcal{Q}$  and  $\mathcal{H}_{1,0} = \mathcal{R}$ .
  3. For  $j = 1, \dots, m$
  4.  $\mathcal{W} = \mathcal{A} *_N \mathcal{V}_j^b$
  5. for  $i = 1, \dots, j$ .
    - $\mathcal{H}_{ij} = \mathcal{V}_i^{bT} *_N \mathcal{W}$ ,
    - $\mathcal{W} = \mathcal{W} - \mathcal{V}_i^b *_M \mathcal{H}_{ij}$
  6. endfor
  7.  $\mathcal{W} = \mathcal{Q} *_M \mathcal{R}$ ,  $\mathcal{V}_{j+1}^b = \mathcal{Q}$  and  $\mathcal{H}_{j+1,j} = \mathcal{R}$ .
  8. EndFor
- 

For notational simplicity, we set

$$\begin{aligned} \mathbb{V}_m^b &= [\mathcal{V}_1^b, \mathcal{V}_2^b, \dots, \mathcal{V}_m^b] \in \mathbb{R}^{\mathcal{I}^N \times \gamma_1 \times \dots \times \gamma_M} \\ \mathbb{V}_{m+1}^b &= [\mathbb{V}_m^b, \mathcal{V}_{m+1}^b] \in \mathbb{R}^{\mathcal{I}^N \times \lambda_1 \times \dots \times \lambda_M} \\ \mathbb{E}_m^T &= [\mathcal{O}_M, \mathcal{O}_M, \dots, \mathcal{I}_M] \in \mathbb{R}^{J^M \times \gamma_1 \times \dots \times \gamma_M} \\ \mathbb{H}_m &= (\mathcal{H}_{i,j})_{1 \leq i, j \leq m} \in \mathbb{R}^{\gamma_1 \times \dots \times \gamma_M \times \gamma_1 \times \dots \times \gamma_M} \\ \tilde{\mathbb{H}}_m &= \begin{bmatrix} & \mathbb{H}_m \\ \mathcal{H}_{m+1,m} *_M \mathbb{E}_m^T & \end{bmatrix} \in \mathbb{R}^{\lambda_1 \times \dots \times \lambda_M \times \gamma_1 \times \dots \times \gamma_M} \end{aligned}$$

where  $\mathcal{I}^N = I_1 \times \dots \times I_N$  and  $J^M = J_1 \times \dots \times J_M$ ,  $\gamma_i = m J_i$  and  $\lambda_i = (m + 1) J_i$  for  $i = 1, \dots, M$ . We end this section by the following proposition which could be directly deduced from the steps of Algorithm 5.

**Proposition 7.** We can state now the following useful relations that could be easily proved.

$$\mathcal{A} *_N \mathbb{V}_m^b = \mathbb{V}_{m+1}^b *_M \tilde{\mathbb{H}}_m, \tag{36}$$

and

$$\mathcal{A} *_N \mathbb{V}_m^b = \mathbb{V}_m^b *_M \mathbb{H}_m + \mathcal{V}_{m+1}^b *_M \mathcal{H}_{m+1,m} *_M \mathbb{E}_m^T. \tag{37}$$

4.3. Block GMRES via the Einstein product

Here, we consider the tensor equation  $\mathcal{A} *_N \mathcal{X} = \mathcal{C}$  where  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  and  $\mathcal{C} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ . Let  $\mathcal{V} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ . If the sub-columns of a tensor  $\mathcal{U}$  of size  $(I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M)$  are in  $\mathcal{K}_m^{Block}(\mathcal{A}, \mathcal{V})$ , then  $\mathcal{U}$  can be expressed as

$$\mathcal{U} = \sum_{i=1}^m \mathcal{A}^{i-1} *_N \mathcal{V} *_M \mathcal{S}$$

where  $\mathcal{S} \in \mathbb{R}^{J_1 \times \dots \times J_M \times J_1 \times \dots \times J_M}$ . Let  $\mathcal{X}_0 \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$  be an initial guess tensor and  $\mathcal{R}_0 = \mathcal{C} - \mathcal{A} *_N \mathcal{X}_0$  be the corresponding residual. The  $m$ -th step of block GMRES method determines the approximation  $\mathcal{X}_m$  to the exact solution  $\mathcal{X}^*$  of (1) such that

$$\mathcal{X}_m - \mathcal{X}_0 \in \mathcal{K}_m^{Block}(\mathcal{A}, \mathcal{R}_0) \tag{38}$$

and

$$\mathcal{R}_{i,m} \perp \mathcal{K}_m^{Block}(\mathcal{A}, \mathcal{A} *_N \mathcal{R}_0) \tag{39}$$

where  $\mathcal{R}_{i,m}$  denote the  $i$ -th sub-column of  $\mathcal{R}_m = \mathcal{C} - \mathcal{A} *_N \mathcal{X}_m$ . Consider the block column tensor  $\mathbb{L}_m = [\mathcal{R}_0, \mathcal{A} *_N \mathcal{R}_0, \dots, \mathcal{A}^{m-1} *_N \mathcal{R}_0] \in \mathbb{R}^{I_1 \times \dots \times I_N \times \gamma_1 \times \dots \times \gamma_M}$  and  $\mathbb{W}_m = \mathcal{A} *_N \mathbb{L}_m$ , where  $\gamma_i = m J_i, i = 1, \dots, M$ . The relation (38) can be expressed as follows

$$\mathcal{X}_m = \mathcal{X}_0 + \mathbb{L}_m *_M \mathbb{Y}_m \tag{40}$$

where  $\mathbb{Y}_m = \begin{bmatrix} \mathcal{Y}_1 \\ \vdots \\ \mathcal{Y}_m \end{bmatrix} \in \mathbb{R}^{\gamma_1 \times \dots \times \gamma_M \times J_1 \times \dots \times J_M}, \mathcal{Y}_i \in \mathbb{R}^{J_1 \times \dots \times J_M \times J_1 \times \dots \times J_M}$ .

The orthogonality condition (39) implies that

$$(\mathbb{W}_m^T *_N \mathbb{W}_m) *_M \mathbb{Y}_m = \mathbb{W}_m^T *_N \mathcal{R}_0 \tag{41}$$

Using the relations (40) and (41), it is not difficult to conclude the following theorem.

**Theorem 2.** Assuming that the tensor  $(\mathbb{W}_m^T *_N \mathbb{W}_m)$  is nonsingular, then the approximation  $\mathcal{X}_m$  and the corresponding residual  $\mathcal{R}_m$  are expressed as follows

$$\mathcal{X}_m = \mathcal{X}_0 + \mathbb{L}_m *_M (\mathbb{W}_m^T *_N \mathbb{W}_m)^{-1} *_M (\mathbb{W}_m^T *_N \mathcal{R}_0) \tag{42}$$

and

$$\mathcal{R}_m = \mathcal{R}_0 - \mathbb{W}_m *_M (\mathbb{W}_m^T *_N \mathbb{W}_m)^{-1} *_M (\mathbb{W}_m^T *_N \mathcal{R}_0) \tag{43}$$

As we are dealing with an orthogonal projection, the residual norm of the iterates produced by the tensor block GMRES satisfies the following minimization property.

$$\|\mathcal{R}_m\|_F = \min_{\mathcal{Z} \in \mathcal{K}_m(\mathcal{A}, \mathcal{R}_0)} \|\mathcal{C} - \mathcal{A} *_N \mathcal{Z}\|_F \tag{44}$$

Notice that  $\mathcal{Z} = \mathcal{X}_0 + \mathbb{L}_m *_M \mathbb{Y}$  with  $\mathbb{Y} \in \mathbb{R}^{\gamma_1 \times \dots \times \gamma_M \times J^M}$  and then, by using the relation (36) and the fact that  $\mathcal{R}_0 = \mathcal{V}_1^b *_M \mathcal{H}_{1,0} = \mathbb{V}_{m+1}^b *_M \mathbb{E}_1 *_M \mathcal{H}_{1,0}$ , we observe that

$$\begin{aligned} \mathcal{R}_m &= \mathcal{R}_0 - \mathcal{A} *_N \mathbb{L}_m *_M \mathbb{Y} \\ &= \mathbb{V}_{m+1}^b *_M \mathbb{E}_1 *_M \mathcal{H}_{1,0} - \mathbb{V}_{m+1}^b *_M \tilde{\mathbb{H}}_m *_M \mathbb{Y} \\ &= \mathbb{V}_{m+1}^b *_M (\mathbb{E}_1 *_M \mathcal{H}_{1,0} - \tilde{\mathbb{H}}_m *_M \mathbb{Y}). \end{aligned}$$

Consequently, one can conclude that

$$\|\mathcal{R}_m\|_F = \min_{\mathbb{Y}} \|\mathbb{E}_1 *_M \mathcal{H}_{1,0} - \tilde{\mathbb{H}}_m *_M \mathbb{Y}\|_F. \tag{45}$$

As a result, the approximation  $\mathcal{X}_m$  is given by

$$\mathcal{X}_m = \mathcal{X}_0 + \mathbb{L}_m *_M \mathbb{Y}_m, \tag{46}$$

where  $\mathbb{Y}_m$  solves the problem (45).

#### 4.4. Block Golub–Kahan method

In this part, we briefly mention a possible implementation for the block version of Golub–Kahan algorithm using the Einstein product. Consider the tensor equation

$$\mathcal{A} *_P \mathcal{X} = \mathcal{C} \tag{47}$$

where  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N \times K_1 \times \dots \times K_P}$  and  $\mathcal{C} \in \mathbb{R}^{I_1 \times \dots \times I_N \times J_1 \times \dots \times J_M}$ . The block version of the Golub–Kahan bidiagonalization method via Einstein product produces column block tensors  $\mathbb{U}_{m+1} \in \mathbb{R}^{I_1 \times \dots \times I_N \times \lambda_1 \times \dots \times \lambda_M}$  and  $\mathbb{V}_m \in \mathbb{R}^{K_1 \times \dots \times K_P \times \gamma_1 \times \dots \times \gamma_M}$  with orthonormal column tensors  $\mathcal{U}_1, \dots, \mathcal{U}_{m+1}$  and  $\mathcal{V}_1, \dots, \mathcal{V}_m$  where  $\lambda_i = (m + 1) J_i$  and  $\gamma_i = m J_i$  for  $i = 1, \dots, M$ .

The block Golub–Kahan bidiagonalization algorithm via Einstein product is summarized in Algorithm 6.

**Algorithm 6** Block Golub–Kahan algorithm via Einstein product.

1. **Inputs** The tensors  $\mathcal{A}$  and  $\mathcal{C}$ , and an integer  $m$ .
2. Set  $\mathcal{V}_0 = 0$  and  $\mathcal{R}_0 = \mathcal{C} - \mathcal{A} *_{\mathcal{P}} \mathcal{X}_0 = \mathcal{Q} *_{\mathcal{M}} \mathcal{R}$ ,  $\mathcal{U}_1 = \mathcal{Q}$ ,  $\mathcal{A}_1 = \mathcal{R}$ .
3. For  $j = 1, \dots, m$ 
  - (a)  $\tilde{\mathcal{V}} = \mathcal{A}^T *_{\mathcal{N}} \mathcal{U}_j - \mathcal{V}_{j-1} *_{\mathcal{M}} \mathcal{A}_j$
  - (b)  $\tilde{\mathcal{V}} = \mathcal{Q} *_{\mathcal{M}} \mathcal{R}$ , (QR factorization); and set  $\mathcal{V}_j = \mathcal{Q}$ ,  $\mathcal{B}_j = \mathcal{R}$ .
  - (c)  $\tilde{\mathcal{U}} = \mathcal{A} *_{\mathcal{P}} \mathcal{V}_j - \mathcal{U}_j *_{\mathcal{M}} \mathcal{B}_j$ .
  - (d)  $\tilde{\mathcal{U}} = \mathcal{Q} *_{\mathcal{M}} \mathcal{R}$ , (QR factorization); and set  $\mathcal{U}_{j+1} = \mathcal{Q}$ ,  $\mathcal{A}_{j+1} = \mathcal{R}$ .
4. End for

Let  $\tilde{\mathcal{C}}_m \in \mathbb{R}^{\lambda_1 \times \dots \times \lambda_M \times \gamma_1 \times \dots \times \gamma_M}$  and  $\mathcal{C}_m, \mathcal{D}_m \in \mathbb{R}^{\gamma_1 \times \dots \times \gamma_M \times \gamma_1 \times \dots \times \gamma_M}$  as follows

$$\tilde{\mathcal{C}}_m = \begin{bmatrix} \mathcal{B}_1 & & & & & \\ & \mathcal{A}_2 & \mathcal{B}_2 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & \mathcal{A}_m & \mathcal{B}_m \\ & & & & & \mathcal{A}_{m+1} \end{bmatrix}, \mathcal{C}_m = \begin{bmatrix} \mathcal{B}_1 & & & & & \\ & \mathcal{A}_2 & \mathcal{B}_2 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & \mathcal{A}_m & \mathcal{B}_m \end{bmatrix}, \mathcal{D}_m = \begin{bmatrix} \mathcal{B}_1 & \mathcal{A}_2 & & & & \\ & \mathcal{B}_2 & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & & \mathcal{A}_m & \mathcal{B}_m \\ & & & & & \mathcal{B}_m \end{bmatrix}.$$

The following proposition provides some relations that can be directly concluded from steps of Algorithm 6.

**Proposition 8.** The tensors produced by the tensor block Golub-Kahan algorithm satisfy the following relations

$$\mathcal{A} *_{\mathcal{P}} \mathbb{V}_m = \mathbb{U}_{m+1} *_{\mathcal{M}} \tilde{\mathcal{C}}_m, \tag{48}$$

$$= \mathbb{U}_m *_{\mathcal{M}} \mathcal{C}_m + \mathcal{U}_{m+1} *_{\mathcal{M}} \mathcal{A}_{m+1} *_{\mathcal{M}} \mathbb{E}_m^T, \tag{49}$$

$$\mathcal{A}^T *_{\mathcal{N}} \mathbb{U}_m = \mathbb{V}_m *_{\mathcal{M}} \mathcal{D}_m, \tag{50}$$

$$\mathcal{R}_0 = \mathbb{U}_{m+1} *_{\mathcal{M}} \mathbb{E}_1 *_{\mathcal{M}} \mathcal{A}_1, \tag{51}$$

where  $\mathbb{E}_m^T = [\mathcal{O}_M, \mathcal{O}_M, \dots, \mathcal{I}_M] \in \mathbb{R}^{J_1 \times \dots \times J_M \times \gamma_1 \times \dots \times \gamma_M}$ ,  $\mathbb{E}_1 = \begin{bmatrix} \mathcal{I}_M \\ \vdots \\ \mathcal{O}_M \end{bmatrix} \in \mathbb{R}^{\lambda_1 \times \dots \times \lambda_M \times J_1 \times \dots \times J_M}$  and  $\mathbb{U}_m = [\mathcal{U}_1, \dots, \mathcal{U}_m] \in \mathbb{R}^{\mathcal{I}_N \times \gamma_1 \times \dots \times \gamma_M}$ .

**Proposition 9.** Assume that  $\mathbb{V}_m$  is a column block tensor whose block columns  $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_m$  are computed by Algorithm 6. Let  $\mathcal{X}_m = \mathcal{X}_0 + \mathbb{V}_m *_{\mathcal{M}} \mathcal{Y}_m$  with  $\mathcal{Y}_m \in \mathbb{R}^{\gamma_1 \times \dots \times \gamma_M \times J_1 \times \dots \times J_M}$  be an approximation for the exact solution  $\mathcal{X}^*$  of (47). Then

$$\mathcal{X}_m = \arg \min_{\mathcal{X} = \mathcal{X}_0 + \mathbb{V}_m *_{\mathcal{M}} \mathcal{Y}} \|\mathcal{A} *_{\mathcal{P}} \mathcal{X} - \mathcal{C}\|_F \tag{52}$$

where  $\mathcal{Y}_m$  solves the minimization problem

$$\min_{\mathcal{Y}} \|\mathbb{E}_1 *_{\mathcal{M}} \mathcal{A}_1 - \tilde{\mathcal{C}}_m *_{\mathcal{M}} \mathcal{Y}\|_F \tag{53}$$

**Proof.** Using the results of Proposition 8, we get

$$\begin{aligned} \| \mathcal{C} - \mathcal{A} *_{\mathcal{P}} \mathcal{X}_m \|_F &= \| \mathbb{U}_{m+1} *_{\mathcal{M}} \mathbb{E}_1 *_{\mathcal{M}} \mathcal{A}_1 - \mathbb{U}_{m+1} *_{\mathcal{M}} \tilde{\mathcal{C}}_m *_{\mathcal{M}} \mathcal{Y}_m \|_F \\ &= \| \mathbb{U}_{m+1} *_{\mathcal{M}} (\mathbb{E}_1 *_{\mathcal{M}} \mathcal{A}_1 - \tilde{\mathcal{C}}_m *_{\mathcal{M}} \mathcal{Y}_m) \|_F = \| \mathbb{E}_1 *_{\mathcal{M}} \mathcal{A}_1 - \tilde{\mathcal{C}}_m *_{\mathcal{M}} \mathcal{Y}_m \|_F, \end{aligned}$$

which ends the proof.

The low dimensional minimization tensor problem (53) is then solved by a QR factorization of the tensor  $\tilde{\mathcal{C}}_m$ .  $\square$

**5. Numerical results**

This section provides some numerical results to show the performance of Algorithms 2 and 4 when applied to the restoration of blurred and noisy color images and videos. For clarity and definiteness, we first focus on the formulation of a tensor model, describing the blurring that is taking place in the process of going from the exact to the blurred RGB image (or video). Notwithstanding what has just been said, recovering RGB (or video) from their blurry and noisy observations can be seen as a tensor problem of the form (1). Therefore, it is very important to understand how the model (1) can be constructed for RGB images and color video deblurring problems. In what follows, we concentrate only on the formulation

of the tensor model for RGB image deblurring problems and comment at the end of this section how a similar one can be formulated for color video deblurring problems. We recall that an RGB image is just multidimensional array of dimension  $M \times N \times 3$  whose entries are the light intensity. Throughout this paper, we assume that the original RGB image has the same dimensions as the blurred one, and we refer to it as  $N \times N \times 3$  tensor. Let  $\mathcal{C}$  represent the available blurred RGB image,  $\mathcal{X}$  denote the desired unknown blurred RGB, and  $\mathcal{A}$  be the tensor describing the blurring that is taking place in the process of going from  $\mathcal{X}$  to  $\mathcal{C}$ . It is well-known in the literature of image processing that all the blurring operators can be characterized by a Point Spread Function (PSF) describing the blurring process and the boundary conditions outside the image, see [18]. Let  $P$  stand for the two-dimensional PSF array. Once the array  $P$  is specified, we can as well build the blurring tensor  $\mathcal{A}$ . By using the fact that the blurring process of an RGB image is simply a multi-dimensional convolution operation of the PSF array  $P$  and the original three-dimensional image  $\mathcal{X}$ , the blurring tensor  $\mathcal{A}$  can be easily constructed by placing the elements of  $P$  in the appropriate positions. Note that the PSF is a two-dimensional array  $P$  describing the image of a single white pixel, which makes its dimensions much smaller than  $N$ . Therefore,  $P$  contains all the required information about the blurring throughout the RGB image  $\mathcal{C}$ . To illustrate this, the discrete operation for multi-dimensional convolution using a  $3 \times 3$  local and spatially invariant PSF array  $P$  with  $p_{22}$  is its center, and assuming zero boundary conditions, is given by:

$$C_{ijk} = p_{33}\mathcal{X}_{i-1j-1k} + p_{32}\mathcal{X}_{i-1jk} + p_{31}\mathcal{X}_{i-1j+1k} + p_{23}\mathcal{X}_{ij-1k} + p_{22}\mathcal{X}_{ijk} \tag{54}$$

$$+ p_{21}\mathcal{X}_{ij+1k} + p_{13}\mathcal{X}_{i+1j-1k} + p_{12}\mathcal{X}_{i+1jk} + p_{11}\mathcal{X}_{i+1j+1k}, \tag{55}$$

for  $i, j = 1, \dots, N$  and  $k = 1, 2, 3$ . Here the zero boundary conditions are imposed so the values of  $\mathcal{X}$  are zero outside the RGB image, i.e.,  $\mathcal{X}_{i0k} = \mathcal{X}_{iN+1k} = \mathcal{X}_{0jk} = \mathcal{X}_{N+1jk} = 0$  for  $0 < i, j < N + 1$  and  $k = 1, 2, 3$ . The fourth order tensor  $\mathcal{A} \in \mathbb{R}^{N \times N \times N \times N}$  associated with (54), with partition  $(1, N, 1, N)$ , can be partitioned into matrix blocks of size  $N \times N$ . Each block is denoted by  $\mathcal{A}_{i_2, i_4}^{(2,4)} = \mathcal{A}(:, i_2, :, i_4) \in \mathbb{R}^{N \times N}$  with  $i_2, i_4 = 1, 2, \dots, N$ . The nonzero entries of the matrix block  $\mathcal{A}_{a,b}^{(2,4)} \in \mathbb{R}^{N \times N}$  are given by

$$\begin{aligned} (\mathcal{A}_{a,b}^{(2,4)})_{a-1b-1} &= p_{33}; & (\mathcal{A}_{a,b}^{(2,4)})_{ab+1} &= p_{21} \\ (\mathcal{A}_{a,b}^{(2,4)})_{a-1b} &= p_{32}; & (\mathcal{A}_{a,b}^{(2,4)})_{a+1b-1} &= p_{13} \\ (\mathcal{A}_{a,b}^{(2,4)})_{a-1b+1} &= p_{31}; & (\mathcal{A}_{a,b}^{(2,4)})_{a+1b} &= p_{12} \\ (\mathcal{A}_{a,b}^{(2,4)})_{ab-1} &= p_{23}; & (\mathcal{A}_{a,b}^{(2,4)})_{a+1b+1} &= p_{11} \\ (\mathcal{A}_{a,b}^{(2,4)})_{ab} &= p_{22} \end{aligned}$$

for  $a, b = 2, \dots, N - 1$ .

The first following examples apply Algorithms 2 and 4 to the restoration of blurred color image and video that have been contaminated by Gaussian blur and by additive zero-mean white Gaussian noise. We consider the blurring to be local and spatially invariant. In this case the entries of the Gaussian PSF array  $P$  are given by

$$p_{ij} = \exp\left(-\frac{1}{2}\left(\frac{(i-k)}{\sigma}\right)^2 - \frac{1}{2}\left(\frac{(j-\ell)}{\sigma}\right)^2\right),$$

where  $\sigma$  controls the width of the Gaussian PSF and  $(k, \ell)$  is its center, see [18]. Note that  $\sigma$  controls the amount of smoothing, i.e. the larger the  $\sigma$ , the more ill posed the problem. The original tensor image is denoted by  $\hat{\mathcal{X}}$  in each example and  $\mathcal{A}$  represents the blurring tensor. The tensor  $\hat{\mathcal{C}} = \mathcal{A} *_{\mathcal{N}} \hat{\mathcal{X}}$  represents the associated blurred and noise-free multichannel image. We generated a blurred and noisy tensor image  $\mathcal{C} = \hat{\mathcal{C}} + \mathcal{N}$ , where  $\mathcal{N}$  is a noise tensor with normally distributed random entries with zero mean and with variance chosen to correspond to a specific noise level  $\nu := \|\mathcal{N}\|_F / \|\hat{\mathcal{C}}\|_F$ . To determine the effectiveness of our solution methods, we evaluate

$$RE = \frac{\|\hat{\mathcal{X}} - X_{\text{restored}}\|_F}{\|\hat{\mathcal{X}}\|_F}$$

and the Signal-to-Noise Ratio (SNR) defined by

$$SNR(X_{\text{restored}}) = 10 \log_{10} \frac{\|\hat{\mathcal{X}} - E(\hat{\mathcal{X}})\|_F^2}{\|X_{\text{restored}} - \hat{\mathcal{X}}\|_F^2}$$

where  $E(\hat{\mathcal{X}})$  denotes the mean gray-level of the uncontaminated image  $\hat{\mathcal{X}}$ . All computations were carried out using the MATLAB environment on an Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz (8 CPUs) computer with 12 GB of RAM. The computations were done with approximately 15 decimal digits of relative accuracy.



Fig. 1. Example 1: Original image (left), blurred and noisy image (right).

Table 1  
Results for Example 1.

Noise level	Method	PSNR	RE	CPU-time (seconds)
$10^{-3}$	Algorithm 2	21.76	$6.09 \times 10^{-2}$	8.28
	Algorithm 4	24.37	$4.51 \times 10^{-2}$	7.29
	GKB	24.22	$4.51 \times 10^{-2}$	18.45
$10^{-2}$	Algorithm 2	20.60	$6.96 \times 10^{-2}$	3.31
	Algorithm 4	20.97	$6.67 \times 10^{-2}$	1.58
	GKB	20.08	$6.66 \times 10^{-2}$	5.52



Fig. 2. Example 1: Restored image by Algorithm 4 (left), and restored image by Algorithm 2 (right).

5.1. Example 1

This example illustrates the performance of Algorithms 2 and 4 when applied to the restoration of 3-channel RGB color image that have been contaminated by Gaussian blur and additive noise. The original (unknown) RGB image  $\hat{\mathcal{X}} \in \mathbb{R}^{256 \times 256 \times 3}$  is the **papav256** image from **MATLAB**. It is shown on the left-hand side of Fig. 1. For the blurring tensor  $\mathcal{A}$ , we consider a PSF array  $P$  with  $\sigma = 2$  under zero boundary conditions. The associated blurred and noisy RGB image  $\hat{\mathcal{C}} = \mathcal{A} *_N \hat{\mathcal{X}}$  is shown on the right-hand side of Fig. 1 with the noise level  $\nu = 10^{-3}$ . Given the contaminated RGB image  $\mathcal{C}$ , we would like to recover an approximation of the original RGB image  $\hat{\mathcal{X}}$ . Table 1 compares, the computing time (in seconds), the relative errors and the PSNR of the computed restorations. The restoration for noise level  $\nu = 10^{-3}$  is shown on the left-hand side of Fig. 2 and it is obtained by applying Einstein tensor global GMRES method (Algorithm 2) with input  $\mathcal{A}, \mathcal{C}, \mathcal{X}_0 = \mathcal{O}, \varepsilon = 10^{-6}, m = 10$  and **Maxit** = 10. Using GCV, the computed optimal value for the projected problem in Algorithm 2 was  $\mu = 9.44 \times 10^{-4}$ . The strategy of performing restarts with GCV is essentially based on restarting the global Arnoldi algorithm: at each restart the regularization parameter, the initial guess are updated employing the last values computed when  $m$  steps of global Arnoldi are used, where  $m$  is a chosen parameter; we keep restarting the global Arnoldi algorithm to a certain number of outer iterations that guarantees that the obtained solution and the computed regularization parameter satisfy the discrepancy principle. The restoration obtained with Algorithm 4 is shown on the right-hand side of Fig. 2. The discrepancy principle with  $\eta = 1.1$  is satisfied when  $\ell = 61$  steps of the method have been carried out, producing a regularization parameter given by  $\mu_\ell = 2.95 \times 10^{-4}$ . For comparison with existing approaches in the literature, we report in Table 1 the results obtained with the method proposed in [7] for solving the corresponding matrix form of (4). This method utilizes the connection between (standard) Golub–Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations obtained by matricization of problem (4). We refer to this method as GKB. It is a solution method based on first reducing the blurring matrix to a small bidiagonal matrix with the aid of Golub–Kahan bidiagonalization (GKB) and then applying the connection between GKB and Gauss-type quadrature rules (the same as the ones in (23)–(24)) to determine an approximation of  $x_{\mu}$  that satisfies the discrepancy principle associated to the matrix problem. It determines the regularization parameter analogously to Algorithm 4, and uses a similar stopping criterion. The results in Table 1 with GKB method have been obtained after having carried out  $\ell = 61$  steps of GKB algorithm. From





Fig. 3. Frame no. 5: Original frame (left), blurred and noisy frame (right).

Table 2  
Results for Example 2.

Noise level	Method	PSNR	Relative error	CPU-time (second)
$10^{-3}$	Algorithm 2	15.48	$6.84 \times 10^{-2}$	38.93
	Algorithm 4	19.24	$4.43 \times 10^{-2}$	27.37
$10^{-2}$	Algorithm 2	14.50	$7.65 \times 10^{-2}$	15.55
	Algorithm 4	15.13	$7.11 \times 10^{-2}$	4.40

Table 1, we can state that tensor Krylov-based methods perform better than their matrix counterpart in terms of CPU-time. The GKB method performs equally well as Algorithm 4 in terms of restorations quality, but Algorithm 4 is found to require less CPU time than GKB, this is because the blurring tensor has a lower bandwidth than the blurring matrix. In fact, reducing the bandwidth of these sparse blurring matrices directly and substantially improves the number of operations and storage locations. The cost of this computation is dominated by the evaluation of one tensor-tensor product (TTP) with  $\mathcal{A}$  for Algorithm 2 and Algorithm 4. Table 1 shows that Algorithm 2 requires more CPU time than Algorithm 4. This is due to the fact that Algorithm 4 performs less tensor-tensor products than Algorithm 2. About the number of iterations to carry out when performing GCV for each restart, it is well-known that the approximate solutions computed by the Arnoldi-Tikhonov method can very quickly fulfill the discrepancy principle and deliver a regularized solution belonging to a Krylov subspace of dimension  $m$ . We can note that, for this example, the chosen  $m = 10$  satisfies the discrepancy principle. As a consequence, the generated Krylov subspaces tend to be the same and the discrepancy principle continues to be satisfied. In this situation, although the quality of the reconstruction may not be substantially improved, performing additional restarts could still be useful in order to keep updating the regularization parameter with slightly better approximations of the solution and, as a consequence, obtain slightly more accurate reconstructions.

### 5.2. Example 2

In this example, we evaluate the effectiveness of Algorithms 2 and 4 when applied to the restoration of a color video defined by a sequence of RGB images. Video restoration is the problem of restoring a sequence of  $k$  color images (frames). Each frame is represented by a tensor of  $N \times N \times 3$  pixels. In the present example, we are interested in restoring 10 consecutive frames of a contaminated video. We consider the xylophone video from MATLAB. The video clip is in MP4 format with each frame having  $240 \times 240$  pixels. The (unknown) blur- and noise-free frames are stored in the tensor  $\hat{\mathcal{C}} \in \mathbb{R}^{N \times N \times 3 \times 10}$ . These frames are blurred by a blurring tensor  $\mathcal{A}$  of the same kind and with the same parameters as in the previous example. Fig. 3 shows the 5th exact (original) frame and the contaminated version, which is to be restored. Blurred and noisy frames are generated by  $\hat{\mathcal{C}} = \mathcal{A} *_N \hat{\mathcal{X}}$  where the tensor  $\mathcal{E}$  represents white Gaussian noise of levels  $\nu = 10^{-3}$  or  $\nu = 10^{-2}$ . Table 2 displays the performance of algorithms. For Algorithm 2, we have used as an input  $\mathcal{A}$ ,  $\mathcal{C}$ ,  $\mathcal{X}_0 = \mathcal{O}$ ,  $\varepsilon = 10^{-6}$ ,  $m = 10$  and **Maxit** = 10. For the ten outer iterations, minimizing the GCV function produces  $\mu_{10} = 9.44 \times 10^{-4}$ . Using Algorithm 4, the discrepancy principle with  $\eta = 1.1$  has been satisfied after  $\ell = 59$  steps of the method, producing a regularization parameter given by  $\mu_\ell = 1.06 \times 10^{-4}$ . The restorations obtained with Algorithms 2 and 4 are shown on the left-hand and right-hand sides of Fig. 4, respectively. Similar to Example 1, Table 2 also shows that Algorithm 2 requires more CPU time than Algorithm 4. The primary reason is that factorization using Golub-Kahan bidiagonalization contains the relevant desired (spectral) information. Precisely, we can get slightly better approximations of the solution whenever we increase  $\ell$ . It is shown in [14] that for sufficiently many steps of GKB, the spaces  $\text{range}(\mathbb{U}_{\ell+1})$  and  $\text{range}(\mathbb{V}_{\ell+1})$  contain to high accuracy the subspaces of  $s$  spectral vectors, for  $s \geq 1$  fixed and not too large. Computed examples in [14] indicate that it often suffices to choose  $\ell \leq 3s$ . It is therefore unlikely that many steps of this bidiagonalization process have to be carried out in order to be able to compute useful approximations of the columns of the desired solution. On the one hand, after the first iterations of the Arnoldi algorithm, singular values of the Hessenberg matrices approximate the largest singular values of the original tensor  $\mathcal{A}$ ; by solving the projected problem using Tikhonov regularization. The computed relative errors essentially depend on the decay of the singular values of the Hessenberg matrices, and therefore they tend to have a similar behavior after a certain number of iterations have been completed.



Fig. 4. Frame no. 5: Restored frame by Algorithm 4 (left), and restored frame by Algorithm 2 (right).

## 6. Conclusions

In this paper, we extended the GMRES and Golub–Kahan bidiagonalization in conjunction of Tikhonov regularization for solving (possibly) ill-conditioned multilinear systems via Einstein product with perturbed right-hand side. We gave also tensor block version of these processes and derive some theoretical results. Numerical experiments were disclosed for image and video processing to demonstrate the feasibility of the proposed iterative algorithms.

## References

- [1] A.H. Bentbib, M. El Guide, K. Jbilou, L. Reichel, Global Golub–Kahan bidiagonalization applied to large discrete ill-posed problems, *J. Comput. Appl. Math.* 322 (2017) 46–56.
- [2] A.H. Bentbib, M. El Guide, K. Jbilou, E. Onunwor, L. Reichel, Solution methods for linear discrete ill-posed problems for color image restoration, *BIT Numer. Math.* 58 (2018) 555–576.
- [3] M. Brazell, N. Li, C. Navasca, C. Tamon, Solving multilinear systems via tensor inversion, *SIAM J. Matrix Anal. Appl.* 34 (2013) 542–570.
- [4] F.P.A. Beik, F.S. Movahed, S. Ahmadi-Asl, On the Krylov subspace methods based on tensor format for positive definite Sylvester tensor equations, *Numer. Linear Algebra Appl.* 23 (2016) 444–466.
- [5] F.P.A. Beik, K. Jbilou, M. Najafi-Kalyani, L. Reichel, Golub–Kahan bidiagonalization for ill-conditioned tensor equations with applications, *Numer. Algorithms* 84 (2020) 1535–1563.
- [6] A. Bouhamidi, K. Jbilou, A Sylvester-Tikhonov regularization method for image restoration, *J. Comput. Appl. Math.* 206 (2007) 86–98.
- [7] D. Calvetti, L. Reichel, Tikhonov regularization with a solution constraint, *SIAM J. Sci. Comput.* 26 (2004) 224–239.
- [8] D. Calvetti, G.H. Golub, L. Reichel, Estimation of the L-curve via Lanczos bidiagonalization, *BIT Numer. Math.* 39 (1999) 603–619.
- [9] D. Calvetti, P.C. Hansen, L. Reichel, L-curve curvature bounds via Lanczos bidiagonalization, *Electron. Trans. Numer. Anal.* 14 (2002) 134–149.
- [10] P. Comon, Tensor decompositions, in: J.G. McWhirter, I.K. Proudler (Eds.), *Mathematics in Signal Processing V*, Clarendon Press, Oxford, UK, 2002, pp. 1–24.
- [11] W.Y. Ding, Y.M. Wei, Solving multi-linear systems with M-tensors, *J. Sci. Comput.* 68 (2016) 689–715.
- [12] M. El Guide, A. El Ichi, K. Jbilou, R. Sadaka, On tensor GMRES and Golub–Kahan methods via the T-product for color image processing, *Electron. J. Linear Algebra* 37 (2021) 524–543.
- [13] M. El Guide, A. El Ichi, K. Jbilou, Discrete cosine transform LSQR methods for multidimensional ill-posed problems, *J. Math. Model.* 10 (2022) 21–37.
- [14] S. Gazzola, E. Onunwor, L. Reichel, G. Rodriguez, On the Lanczos and Golub–Kahan reduction methods applied to discrete ill-posed problems, *Numer. Linear Algebra Appl.* 23 (2016) 187–204.
- [15] G.H. Golub, M. Heath, G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, *Technometrics* 21 (1979) 215–223.
- [16] P.C. Hansen, Analysis of discrete ill-posed problems by means of the L-curve, *SIAM Rev.* 34 (1992) 561–580.
- [17] P.C. Hansen, *Regularization tools, a MATLAB package for analysis of discrete regularization problems*, *Numer. Algorithms* 6 (1994) 1–35.
- [18] P.C. Hansen, J. Nagy, D.P. O’Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.
- [19] B. Huang, Y. Xie, C. Ma, Krylov subspace methods to solve a class of tensor equations via the Einstein product, *Numer. Linear Algebra Appl.* 26 (2019) e2254.
- [20] K. Jbilou, A. Messaoudi, H. Sadok, Global FOM and GMRES algorithms for matrix equations, *Appl. Numer. Math.* 31 (1999) 49–63.
- [21] M.E. Kimler, C.D. Martin, Factorization strategies for third-order tensors, *Linear Algebra Appl.* 435 (2011) 641–658.
- [22] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (2009) 455–500.
- [23] P. Kroonenberg, *Three-Mode Principal Component Analysis: Theory and Applications*, DSWO Press, Leiden, 1983.
- [24] L. De Lathauwer, A. de Baynast, Blind deconvolution of DS-CDMA signals by means of decomposition in rank-( $I, L, L$ ) terms, *IEEE Trans. Signal Process.* 56 (2008) 1562–1571.
- [25] M.L. Liang, B. Zheng, R.J. Zhao, *Tensor Inversion and Its Application to the Tensor Equations with Einstein Product*, 2018, pp. 843–870.
- [26] X.T. Li, M.K. Ng, Solving sparse non-negative tensor equations: algorithms and applications, *Front. Math. China* 10 (2015) 649–680.
- [27] Q. Qi, Z. Luo, *Tensor Analysis: Spectral Theory and Special Tensors*, vol. 151, SIAM, Philadelphia, 2017.
- [28] A. Smilde, R. Bro, P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*, Wiley, 2004.
- [29] L. Sun, B. Zheng, C. Bu, Y. Wei, Moore–Penrose inverse of tensors via Einstein product, *Linear Multilinear Algebra* 64 (2016) 686–698.
- [30] A.N. Tikhonov, Regularization of incorrectly posed problems, *Sov. Math.* 4 (1963) 1624–1627.
- [31] G. Wahba, Practical approximation solutions to linear operator equations when the data are noisy, *SIAM J. Numer. Anal.* 14 (1977) 651–667.