



## ON TENSOR GMRES AND GOLUB-KAHAN METHODS VIA THE T-PRODUCT FOR COLOR IMAGE PROCESSING\*

MOHAMED EL GUIDE<sup>†</sup>, ALAA EL ICHI<sup>‡</sup>, KHALIDE JBILOU<sup>§</sup>, AND RACHID SADAKA<sup>¶</sup>

**Abstract.** The present paper is concerned with developing tensor iterative Krylov subspace methods to solve large multi-linear tensor equations. We use the T-product for two tensors to define tensor tubal global Arnoldi and tensor tubal global Golub-Kahan bidiagonalization algorithms. Furthermore, we illustrate how tensor-based global approaches can be exploited to solve ill-posed problems arising from recovering blurry multichannel (color) images and videos, using the so-called Tikhonov regularization technique, to provide computable approximate regularized solutions. We also review a generalized cross-validation and discrepancy principle type of criterion for the selection of the regularization parameter in the Tikhonov regularization. Applications to image sequence processing are given to demonstrate the efficiency of the algorithms.

**Key words.** Krylov subspaces, Linear tensor equations, Tensors, T-product.

**AMS subject classifications.** 65F10, 65F22.

**1. Introduction.** The aim of this paper is to solve the following tensor problem:

$$(1.1) \quad \mathcal{M}(\mathcal{X}) = \mathcal{C},$$

where  $\mathcal{M}$  is a linear operator that could be described as

$$(1.2) \quad \mathcal{M}(\mathcal{X}) = \mathcal{A} \star \mathcal{X},$$

or as

$$(1.3) \quad \mathcal{M}(\mathcal{X}) = \mathcal{A} \star \mathcal{X} \star \mathcal{B},$$

where  $\mathcal{A}$ ,  $\mathcal{X}$ ,  $\mathcal{B}$ , and  $\mathcal{C}$  are three-way tensors, leaving the specific dimensions to be defined later, and  $\star$  is the T-product to be also defined later. To mention but a few applications, problems of these types arise in engineering [34], signal processing [30], data mining [31], tensor complementarity problems [32], computer vision [37, 38], and graph analysis [23]. For those applications, and so many more, one has to take advantage of this multidimensional structure to build rapid and robust iterative methods for solving large-scale problems. We will then be interested in developing robust and fast iterative tensor Krylov subspace methods under tensor-tensor product framework between third-order tensors, to solve regularized 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, 9, 11]. However, these methods might be time consuming

---

\*Received by the editors on June 10, 2020. Accepted for publication on June 27, 2021. Handling Editor: Valeria Simoncini. Corresponding Author: Khalide Jbilou.

<sup>†</sup>Africa Institute for Research in Economics and Social Sciences (AIRESS), Research in Behavioral Economics and Risk Analysis (BERA), FGSES, Mohammed VI Polytechnic University, Green City, Morocco

<sup>‡</sup>Department of Mathematics, University Mohammed V Rabat, Morocco and University Littoral Cote d'Opale, Calais France

<sup>§</sup>LMPA, 50 rue F. Buisson, University Littoral Cote d'Opale Calais, France and Mohammed VI Polytechnic University, Green City, Morocco (jbilou@univ-littoral.fr)

<sup>¶</sup>Ecole Normale Supérieure Takaddoum, Département de Mathématiques, B.P. 5118, Av. Oued Akreuch, Takaddoum, Rabat, Morocco.

to numerically solve problems related to multichannel images (e.g. color images, hyperspectral images and videos).

For the Einstein product, both the Einstein tensor global Arnoldi and Einstein tensor global Golub-Kahan bidiagonalization algorithms have been established [12], which makes so natural to generalize these methods using the T-product. In this paper, we will show that the tensor–tensor product between third-order tensors allows the application of the global iterative methods, such as the global Arnoldi and global Golub-Kahan algorithms. The tensor form of the proposed Krylov methods, together with using the fast Fourier transform (FFT) to compute the T-product between third-order tensors can be efficiently implemented on many modern computers and allows to significantly reduce the overall computational complexity. It is also worth mentioning that our approaches can be naturally generalized to higher-order tensors in a recursive manner.

The paper is organized as follows. We shall first present in Section 2 some symbols and notations used throughout paper. We also recall the concept of the T-product between two tensors. In Section 3, we define tensor global Arnoldi and tensor global Golub-Kahan algorithms that allow the use of the T-product. Section 4 reviews on the adaptation of Tikhonov regularization for the tensor equation (1.1) and then proposing a restarting strategy of the tensor global GMRES and tensor global Golub-Kahan approaches in connection with Gauss-type quadrature rules to inexpensively compute solutions of the regularization of (1.1). In Section 5, we give a tensor formulation in the form of (1.1) that describes the cross-blurring of color image and then we present a few numerical examples on restoring blurred and noisy color images and videos. Concluding remarks can be found in Section 6.

**2. Definitions and notations.** A tensor is a multidimensional array of data. The number of indices of a tensor is called modes or ways. Notice that a scalar can be regarded as a zero mode tensor, first mode tensors are vectors, and matrices are second mode tensor. The order of a tensor is the dimensionality of the array needed to represent it, also known as ways or modes. For a given N-mode (or order-N) tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_N}$ , the notation  $x_{i_1, \dots, i_N}$  (with  $1 \leq i_j \leq n_j$  and  $j = 1, \dots, N$ ) stands for the element  $(i_1, \dots, i_N)$  of the tensor  $\mathcal{X}$ . The norm of a tensor  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_N}$  is specified by

$$\|\mathcal{A}\|_F^2 = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_N=1}^{n_N} a_{i_1 i_2 \dots i_N}^2.$$

Corresponding to a given tensor  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_N}$ , the notation

$$\underbrace{\mathcal{A}:: \dots ::}_k, \quad k = 1, 2, \dots, n_N,$$

*(N-1)-times*

denotes a tensor in  $\mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_{N-1}}$  which is obtained by fixing the last index and is called frontal slice. Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing all the indexes except one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Third-order tensors have column, row, and tube fibers. An element  $c \in \mathbb{R}^{1 \times 1 \times n}$  is called a tubal scalar of length  $n$ . More details are found in [24, 22].

In the present paper, we will consider only third-order tensors and show how to use them in color image and video processing.

**2.1. Discrete Fourier Transformation.** In this subsection, we recall some definitions and properties of the discrete Fourier transformation and the T-product. The Discrete Fourier Transformation (DFT) plays

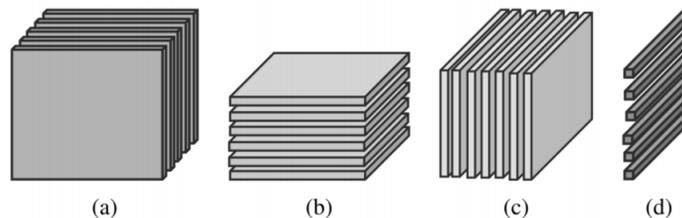


Figure 1: (a) Frontal, (b) horizontal, and (c) lateral slices of a third-order tensor. (d) A mode-3 tube fibers.

a very important role in the definition of the T-product of tensors. The DFT on a vector  $v \in \mathbb{R}^n$  is defined by

$$(2.1) \quad \tilde{v} = F_n v \in \mathbb{C}^n,$$

where  $F_n$  is the Fourier matrix defined by

$$(2.2) \quad F_n = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{pmatrix} \in \mathbb{C}^{n \times n},$$

where  $\omega = e^{\frac{-2\pi i}{n}}$  with  $i^2 = -1$ . It is not difficult to show that (see [13])

$$(2.3) \quad F_n^* = \overline{F_n}, \text{ and } F_n^* F_n = F_n F_n^* = nI_n.$$

Then  $F_n^{-1} = \frac{1}{n} \overline{F_n}$  and  $\frac{1}{\sqrt{n}} F_n$  is a unitary matrix. The cost of computing the vector  $\tilde{v}$  directly from (2.1) is  $O(n^2)$ . Using the fast Fourier Transform (**fft**), it will costs  $O(n \log(n))$ . It is known that

$$(2.4) \quad F_n \text{circ}(v) F_n^{-1} = \text{Diag}(\tilde{v}),$$

which is equivalent to

$$(2.5) \quad F_n \text{circ}(v) F_n^* = n \text{Diag}(\tilde{v}),$$

where

$$\text{circ}(v) = \begin{pmatrix} v_1 & v_2 & \dots & v_n \\ v_2 & v_1 & \dots & v_3 \\ \vdots & \vdots & \dots & \vdots \\ v_n & v_{n-1} & \dots & v_1 \end{pmatrix},$$

and  $\text{Diag}(\tilde{v})$  is the diagonal matrix whose  $i$ -th diagonal element is  $(\tilde{v})_i$ . The decomposition (2.4) shows that the columns of  $F_n$  are the eigenvectors of  $(\text{circ}(v))^T$ .

**2.2. Definitions and properties of the T-product.** In this subsection, we briefly review some concepts and notations, which play a central role for the elaboration of the tensor global iterative methods based on the T-product; see [3, 17, 26, 24] for more details on the T-product. Let  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  be a third-order tensor, then the operations `bcirc`, `unfold`, and `fold` are defined by

$$\text{bcirc}(\mathcal{A}) = \begin{pmatrix} A_1 & A_{n_3} & A_{n_3-1} & \dots & A_2 \\ A_2 & A_1 & A_{n_3} & \dots & A_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A_{n_3} & A_{n_3-1} & \ddots & A_2 & A_1 \end{pmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3},$$

$$\text{unfold}(\mathcal{A}) = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_{n_3} \end{pmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2}, \quad \text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A}.$$

Let  $\tilde{\mathcal{A}}$  be the tensor obtained by applying the DFT on all the tubes of the tensor  $\mathcal{A}$ . With the MATLAB command `fft` as

$$\tilde{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3), \text{ and } \text{ifft}(\tilde{\mathcal{A}}, [], 3) = \mathcal{A},$$

where `ifft` denotes the Inverse Fast Fourier Transform.

Let  $\mathbf{A}$  be the matrix

$$(2.6) \quad \mathbf{A} = \begin{pmatrix} A^{(1)} & & & \\ & A^{(2)} & & \\ & & \ddots & \\ & & & A^{(n_3)} \end{pmatrix},$$

where the matrices  $A^{(i)}$ 's are the frontal slices of the tensor  $\tilde{\mathcal{A}}$ . The block circulant matrix `bcirc`( $\mathcal{A}$ ) can also be block diagonalized by using the DFT and this gives

$$(2.7) \quad (F_{n_3} \otimes I_{n_1}) \text{bcirc}(\mathcal{A}) (F_{n_3}^* \otimes I_{n_2}) = \mathbf{A},$$

As noticed in [24, 29], the diagonal blocks of the matrix  $\mathbf{A}$  satisfy the following property:

$$(2.8) \quad \begin{cases} A^{(1)} \in \mathbb{R}^{n_1 \times n_2} \\ \text{conj}(A^{(i)}) = A^{(n_3-i+2)}, \end{cases}$$

where  $\text{conj}(A^{(i)})$  is the complex conjugate of the matrix  $A^{(i)}$ . Next we recall the definition of the T-product.

**DEFINITION 1.** The **T-product** between two tensors  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and  $\mathcal{B} \in \mathbb{R}^{n_2 \times m \times n_3}$  is an  $n_1 \times m \times n_3$  tensor given by:

$$\mathcal{A} \star \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A})\text{unfold}(\mathcal{B})).$$

Notice that from the relation (2.6), we can show that the the product  $\mathcal{C} = \mathcal{A} \star \mathcal{B}$  is equivalent to  $\mathbf{C} = \mathbf{A} \mathbf{B}$ . So, the efficient way to compute the T-product is to use Fast Fourier Transform (FFT).

Using the relation (2.8), the following algorithm allows us to compute, in an efficient way, the T-product of the tensors  $\mathcal{A}$  and  $\mathcal{B}$ , see [29].

**Algorithm 1** Computing the T-product via FFT

Inputs:  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and  $\mathcal{B} \in \mathbb{R}^{n_2 \times m \times n_3}$

Output:  $\mathcal{C} = \mathcal{A} \star \mathcal{B} \in \mathbb{R}^{n_1 \times m \times n_3}$

1. Compute  $\tilde{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$  and  $\tilde{\mathcal{B}} = \text{fft}(\mathcal{B}, [], 3)$ .
2. Compute each frontal slices of  $\tilde{\mathcal{C}}$  by

$$C^{(i)} = \begin{cases} A^{(i)}B^{(i)}, & i = 1, \dots, \lfloor \frac{n_3 + 1}{2} \rfloor \\ \text{conj}(C^{(n_3+i-2)}), & i = \lfloor \frac{n_3 + 1}{2} \rfloor + 1, \dots, n_3. \end{cases}$$

3. Compute  $\mathcal{C} = \text{ifft}(\tilde{\mathcal{C}}, [], 3)$ .

For the T-product, we have the following definitions and remarks:

DEFINITION 2. The identity tensor  $\mathcal{J}_{n_1 n_1 n_3}$  is the tensor whose first frontal slice is the identity matrix  $I_{n_1 n_1}$  and the other frontal slices are all zeros.

DEFINITION 3.

1. An  $n_1 \times n_1 \times n_3$  tensor  $\mathcal{A}$  is invertible, if there exists a tensor  $\mathcal{B}$  of order  $n_1 \times n_1 \times n_3$  such that

$$\mathcal{A} \star \mathcal{B} = \mathcal{J}_{n_1 n_1 n_3} \quad \text{and} \quad \mathcal{B} \star \mathcal{A} = \mathcal{J}_{n_1 n_1 n_3}.$$

In that case, we set  $\mathcal{B} = \mathcal{A}^{-1}$ . It is clear that  $\mathcal{A}$  is invertible if and only if  $\text{bcirc}(\mathcal{A})$  is invertible.

2. The transpose of  $\mathcal{A}$  is obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through  $n_3$ .

REMARK 2.1.

- If  $\mathcal{A}$ ,  $\mathcal{B}$ , and  $\mathcal{C}$  are tensors of appropriate sizes, then

$$(\mathcal{A} \star \mathcal{B}) \star \mathcal{C} = \mathcal{A} \star (\mathcal{B} \star \mathcal{C}).$$

- Suppose  $\mathcal{A}$  and  $\mathcal{B}$  are two tensors such  $\mathcal{A} \star \mathcal{B}$  and  $\mathcal{B}^T \star \mathcal{A}^T$  are defined. Then

$$(\mathcal{A} \star \mathcal{B})^T = \mathcal{B}^T \star \mathcal{A}^T.$$

DEFINITION 4. Let  $\mathcal{A}$  and  $\mathcal{B}$  be two tensors in  $\mathbb{R}^{n_1 \times n_2 \times n_3}$ . Then

1. The scalar inner product is defined by

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} a_{i_1 i_2 i_3} b_{i_1 i_2 i_3}.$$

2. The norm of  $\mathcal{A}$  is defined by

$$\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}.$$

REMARK 2.2. Another interesting way for computing the scalar product and the associated norm is as follows:

$$\langle \mathcal{A}, \mathcal{B} \rangle = \frac{1}{n_3} \langle \mathbf{A}, \mathbf{B} \rangle; \quad \|\mathcal{A}\|_F = \frac{1}{\sqrt{n_3}} \|\mathbf{A}\|_F,$$

where the block diagonal matrix  $\mathbf{A}$  is defined by (2.6).

DEFINITION 5. An  $n_1 \times n_1 \times n_3$  tensor  $\mathcal{Q}$  is orthogonal if

$$\mathcal{Q}^T \star \mathcal{Q} = \mathcal{Q} \star \mathcal{Q}^T = \mathcal{J}_{n_1 n_1 n_3}.$$

LEMMA 6. If  $\mathcal{Q}$  is an orthogonal tensor, then

$$\|\mathcal{Q} \star \mathcal{A}\|_F = \|\mathcal{A}\|_F.$$

DEFINITION 7 ([24]). A tensor is called *f-diagonal* if its frontal slices are diagonal matrices. It is called *upper triangular* if all its frontal slices are upper triangular.

DEFINITION 8 ([33]). (Block tensor based on T-product) Suppose  $\mathcal{A} \in \mathbb{R}^{n_1 \times m_1 \times n_3}$ ,  $\mathcal{B} \in \mathbb{R}^{n_1 \times m_2 \times n_3}$ ,  $\mathcal{C} \in \mathbb{R}^{n_2 \times m_1 \times n_3}$  and  $\mathcal{D} \in \mathbb{R}^{n_2 \times m_2 \times n_3}$  are four tensors. The block tensor

$$\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2) \times (m_1+m_2) \times n_3}$$

is defined by compositing the frontal slices of the four tensors.

Now we introduce the T-diamond tensor product.

DEFINITION 9. Let  $\mathcal{A} = [\mathcal{A}_1, \dots, \mathcal{A}_p] \in \mathbb{R}^{n_1 \times ps \times n_3}$ , where  $\mathcal{A}_i \in \mathbb{R}^{n_1 \times s \times n_3}$ ,  $i = 1, \dots, p$  and let  $\mathcal{B} = [\mathcal{B}_1, \dots, \mathcal{B}_l] \in \mathbb{R}^{n_1 \times ls \times n_3}$  with  $\mathcal{B}_j \in \mathbb{R}^{n_1 \times s \times n_3}$ ,  $j = 1, \dots, l$ . Then, the product  $\mathcal{A}^T \diamond \mathcal{B}$  is the  $p \times l$  matrix given by :

$$(\mathcal{A}^T \diamond \mathcal{B})_{i,j} = \langle \mathcal{A}_i, \mathcal{B}_j \rangle.$$

### 3. Tensor T-global GMRES and tensor T-global Golub-Kahan algorithms.

**3.1. The tensor T-global GMRES .** Consider the following tensor linear system of equations:

$$(3.1) \quad \mathcal{A} \star \mathcal{X} = \mathcal{C},$$

where  $\mathcal{A} \in \mathbb{R}^{n \times n \times p}$ ,  $\mathcal{C}$  and  $\mathcal{X} \in \mathbb{R}^{n \times s \times p}$ . We introduce the tensor Krylov subspace  $\mathcal{TK}_m(\mathcal{A}, \mathcal{V})$  associated with the T-product, defined for the pair  $(\mathcal{A}, \mathcal{V})$  as follows:

$$(3.2) \quad \mathcal{TK}_m(\mathcal{A}, \mathcal{V}) = \text{Tspan}\{\mathcal{V}, \mathcal{A} \star \mathcal{V}, \dots, \mathcal{A}^{m-1} \star \mathcal{V}\} = \left\{ \mathcal{Z} \in \mathbb{R}^{n \times s \times p}, \mathcal{Z} = \sum_{i=1}^m \alpha_i (\mathcal{A}^{i-1} \star \mathcal{V}) \right\},$$

where  $\alpha_i \in \mathbb{R}$ ,  $\mathcal{A}^{i-1} \star \mathcal{V} = \mathcal{A}^{i-2} \star \mathcal{A} \star \mathcal{V}$ , for  $i = 2, \dots, m$  and  $\mathcal{A}^0$  is the identity tensor. In the following algorithm, we define the Tensor T-global Arnoldi algorithm.

---

#### Algorithm 2 Tensor T-global Arnoldi

---

1. **Input.**  $\mathcal{A} \in \mathbb{R}^{n \times n \times p}$ ,  $\mathcal{V} \in \mathbb{R}^{n \times s \times p}$  and the positive integer  $m$ .
  2. Set  $\beta = \|\mathcal{V}\|_F$ ,  $\mathcal{V}_1 = \frac{\mathcal{V}}{\beta}$
  3. For  $j = 1, \dots, m$ 
    - (a)  $\mathcal{W} = \mathcal{A} \star \mathcal{V}_j$
    - (b) for  $i = 1, \dots, j$ 
      - i.  $h_{i,j} = \langle \mathcal{V}_i, \mathcal{W} \rangle$
      - ii.  $\mathcal{W} = \mathcal{W} - h_{i,j} \mathcal{V}_i$
    - (c) End for
    - (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. End
-

It is not difficult to show that after  $m$  steps of Algorithm 2, the tensors  $\mathcal{V}_1, \dots, \mathcal{V}_m$ , form an orthonormal basis of the tensor global Krylov subspace  $\mathcal{TK}_m(\mathcal{A}, \mathcal{V})$ . Let  $\mathbb{V}_m$  be the  $(n \times (sm) \times p)$  tensor with frontal slices  $\mathcal{V}_1, \dots, \mathcal{V}_m$  and let  $\tilde{H}_m$  be the  $(m+1) \times m$  upper Hessenberg matrix whose elements are the  $h_{i,j}$ 's defined by Algorithm 2. Let  $H_m$  be the matrix obtained from  $\tilde{H}_m$  by deleting its last row;  $H_{\cdot,j}$  will denote the  $j$ -th column of the matrix  $H_m$  and  $\mathcal{A} \star \mathbb{V}_m$  is the  $(n \times (sm) \times p)$  tensor with frontal slices  $\mathcal{A} \star \mathcal{V}_1, \dots, \mathcal{A} \star \mathcal{V}_m$ :

$$(3.3) \quad \mathbb{V}_m := [\mathcal{V}_1, \dots, \mathcal{V}_m] \quad \text{and} \quad \mathcal{A} \star \mathbb{V}_m := [\mathcal{A} \star \mathcal{V}_1, \dots, \mathcal{A} \star \mathcal{V}_m].$$

We introduce the product  $\otimes$  defined by

$$\mathbb{V}_m \otimes y = \sum_{j=1}^m y_j \mathcal{V}_j, \quad y = (y_1, \dots, y_m)^T \in \mathbb{R}^m,$$

and we set

$$\mathbb{V}_m \otimes H_m = [\mathbb{V}_m \otimes H_{\cdot,1}, \dots, \mathbb{V}_m \otimes H_{\cdot,m}].$$

Then, it is easy to see that for all vectors  $u$  and  $v$  in  $\mathbb{R}^m$ , we have

$$(3.4) \quad \mathbb{V}_m \otimes (u + v) = \mathbb{V}_m \otimes u + \mathbb{V}_m \otimes v \quad \text{and} \quad (\mathbb{V}_m \otimes H_m) \otimes u = \mathbb{V}_m \otimes (H_m u).$$

With these notations, we can show the following result that will be useful later on.

PROPOSITION 10. *Let  $\mathbb{V}_m$  be the tensor defined by  $[\mathcal{V}_1, \dots, \mathcal{V}_m]$  where  $\mathcal{V}_i \in \mathbb{R}^{n \times s \times p}$  are defined by the Tensor  $T$ -global Arnoldi algorithm. Then, we have*

$$(3.5) \quad \|\mathbb{V}_m \otimes y\|_F = \|y\|_2, \quad \forall y = (y_1, \dots, y_m)^T \in \mathbb{R}^m.$$

*Proof.* From the definition of the product  $\otimes$ , we have  $\sum_{j=1}^m y_j \mathcal{V}_j = \mathbb{V}_m \otimes y$ . Therefore,

$$\|\mathbb{V}_m \otimes y\|_F^2 = \left\langle \sum_{j=1}^m y_j \mathcal{V}_j, \sum_{j=1}^m y_j \mathcal{V}_j \right\rangle_F.$$

But, since the tensors  $\mathcal{V}_i$ 's are orthonormal, it follows that

$$\|\mathbb{V}_m \otimes y\|_F^2 = \sum_{j=1}^m y_j^2 = \|y\|_2^2,$$

which shows the result.

With the above notations, we can easily prove the results of the following proposition.

PROPOSITION 11. *Suppose that  $m$  steps of Algorithm 2 have been run. Then, the following statements hold:*

$$(3.6) \quad \mathcal{A} \star \mathbb{V}_m = \mathbb{V}_m \otimes H_m + h_{m+1,m} [\mathcal{O}_{n \times s \times p}, \dots, \mathcal{O}_{n \times s \times p}, \mathcal{V}_{m+1}],$$

$$(3.7) \quad \mathcal{A} \star \mathbb{V}_m = \mathbb{V}_{m+1} \otimes \tilde{H}_m,$$

$$(3.8) \quad \mathbb{V}_m^T \diamond \mathcal{A} \star \mathbb{V}_m = H_m,$$

$$(3.9) \quad \mathbb{V}_{m+1}^T \diamond \mathcal{A} \star \mathbb{V}_m = \tilde{H}_m,$$

$$(3.10) \quad \mathbb{V}_m^T \diamond \mathbb{V}_m = I_m,$$

where  $I_m$  the identity matrix and  $\mathcal{O}_{n \times s \times p}$  is the tensor having all its entries equal to zero.

*Proof.* From Algorithm 2, we have  $\mathcal{A} \star \mathcal{V}_j = \sum_{i=1}^{j+1} h_{i,j} \mathcal{V}_i$ . Using the fact that  $\mathcal{A} \star \mathbb{V}_m = [\mathcal{A} \star \mathcal{V}_1, \dots, \mathcal{A} \star \mathcal{V}_m]$ , the  $j$ -th frontal slice of  $\mathcal{A} \star \mathbb{V}_m$  is given by

$$(\mathcal{A} \star \mathbb{V}_m)_j = \mathcal{A} \star \mathcal{V}_j = \sum_{i=1}^{j+1} h_{i,j} \mathcal{V}_i.$$

Furthermore, from the definition of the  $\otimes$  product, we have

$$\begin{aligned} (\mathbb{V}_{m+1} \otimes \tilde{H}_m)_j &= \mathbb{V}_{m+1} \otimes H_{:,j}, \\ &= \sum_{i=1}^{j+1} h_{i,j} \mathcal{V}_i, \end{aligned}$$

which proves the first two relations. The other relations follow from the definition of T-diamond product.  $\square$

In the sequel, we develop the tensor T-global GMRES algorithm for solving the problem (3.1). It could be considered as generalization of the global GMERS algorithm [19]. Let  $\mathcal{X}_0 \in \mathbb{R}^{n \times s \times p}$  be an arbitrary initial guess with the corresponding residual  $\mathcal{R}_0 = \mathcal{C} - \mathcal{A} \star \mathcal{X}_0$ . The aim of tensor T-global GMRES method is to find and approximate solution  $\mathcal{X}_m$  approximating the exact solution  $\mathcal{X}^*$  of (3.1) such that

$$(3.11) \quad \mathcal{X}_m - \mathcal{X}_0 \in \mathcal{TK}_m(\mathcal{A}, \mathcal{R}_0),$$

with the classical minimization property

$$(3.12) \quad \|\mathcal{R}_m\|_F = \min_{\mathcal{X} \in \mathcal{X}_0 + \mathcal{TK}_m(\mathcal{A}, \mathcal{R}_0)} \|\mathcal{C} - \mathcal{A} \star \mathcal{X}\|_F.$$

Let  $\mathcal{X}_m = \mathcal{X}_0 + \mathbb{V}_m \otimes y$  with  $y \in \mathbb{R}^m$  be the approximate solution satisfying (3.11). Then,

$$\|\mathcal{R}_m\|_F = \min_{y \in \mathbb{R}^m} \|\mathcal{R}_0 - (\mathcal{A} \star \mathbb{V}_m) \otimes y\|_F,$$

where  $\mathcal{A} \star \mathbb{V}_m := [\mathcal{A} \star \mathcal{V}_1, \dots, \mathcal{A} \star \mathcal{V}_m]$  is the  $(n \times sm \times p)$  tensor defined earlier. Using Proposition 10 and the fact that  $\mathcal{R}_0 = \|\mathcal{R}_0\|_F \mathcal{V}_1$  with  $\mathcal{V}_1 = \mathbb{V}_{m+1} \otimes e_1$ , where  $e_1$  the first canonical basis vector in  $\mathbb{R}^{m+1}$ , we get

$$(3.13) \quad y = \arg \min_{y \in \mathbb{R}^m} \|\|\mathcal{R}_0\|_F e_1 - \tilde{H}_m y\|_2.$$

**3.2. The Tensor T-global Golub-Kahan algorithm.** Instead of using the tensor T-global Arnoldi to generate a basis for the projection subspace, we can define T-version of the tensor global Lanczos process. Here, we will use the tensor Golub-Kahan algorithm related to the T-product.

Let  $\mathcal{A} \in \mathbb{R}^{n \times \ell \times p}$  be a tensor and let  $\mathcal{U} \in \mathbb{R}^{\ell \times s \times p}$  and  $\mathcal{V} \in \mathbb{R}^{n \times s \times p}$  two other given tensors. The Tensor T-global Golub-Kahan bidiagonalization algorithm (associated with the T-product) is defined as follows:

Let  $\tilde{C}_m$  be the upper bidiagonal  $((m+1) \times m)$  matrix

$$\tilde{C}_m = \begin{bmatrix} \alpha_1 & & & & & \\ \beta_2 & \alpha_2 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \beta_m & \alpha_m \\ & & & & & \beta_{m+1} \end{bmatrix},$$

---

**Algorithm 3** The Tensor T-global Golub-Kahan algorithm

---

1. **Input.** The tensors  $\mathcal{A}$ ,  $\mathcal{V}$ , and  $\mathcal{U}$  and an integer  $m$ .
  2. Set  $\beta_1 = \|\mathcal{V}\|_F$ ,  $\alpha_1 = \|\mathcal{U}\|_F$ ,  $\mathcal{V}_1 = \mathcal{V}/\beta_1$  and  $\mathcal{U}_1 = \mathcal{U}/\alpha_1$ .
  3. for  $j = 2, \dots, m$ 
    - (a)  $\tilde{\mathcal{V}} = \mathcal{A} \star \mathcal{U}_{j-1} - \alpha_{j-1} \mathcal{V}_{j-1}$
    - (b)  $\beta_j = \|\tilde{\mathcal{V}}\|_F$  if  $\beta_j = 0$  stop, else
    - (c)  $\mathcal{V}_j = \tilde{\mathcal{V}}/\beta_j$
    - (d)  $\tilde{\mathcal{U}} = \mathcal{A}^T \star \mathcal{V}_j - \beta_j \mathcal{U}_{j-1}$
    - (e)  $\alpha_j = \|\tilde{\mathcal{U}}\|_F$
    - (f) if  $\alpha_j = 0$  stop, else
    - (g)  $\mathcal{U}_j = \tilde{\mathcal{U}}/\alpha_j$ .
- 

and let  $C_m$  be the  $(m \times m)$  matrix obtained by deleting the last row of  $\tilde{C}_m$ . We denote by  $C_{.,j}$  the  $j$ -th column of the matrix  $C_m$ . Let  $\mathbb{U}_m$  and  $\mathcal{A} \star \mathbb{U}_m$  be the  $(\ell \times (sm) \times p)$  and  $(n \times (sm) \times p)$  tensors with frontal slices  $\mathcal{U}_1, \dots, \mathcal{U}_m$  and  $\mathcal{A} \star \mathcal{U}_1, \dots, \mathcal{A} \star \mathcal{U}_m$ , respectively, and let  $\mathbb{V}_m$  and  $\mathcal{A}^T \star \mathbb{V}_m$  be the  $(n \times (sm) \times p)$  and  $(\ell \times (sm) \times p)$  tensors with frontal slices  $\mathcal{V}_1, \dots, \mathcal{V}_m$  and  $\mathcal{A}^T \star \mathcal{V}_1, \dots, \mathcal{A}^T \star \mathcal{V}_m$ , respectively. We set

$$(3.14) \quad \mathbb{U}_m := [\mathcal{U}_1, \dots, \mathcal{U}_m], \quad \text{and} \quad \mathcal{A} \star \mathbb{U}_m := [\mathcal{A} \star \mathcal{U}_1, \dots, \mathcal{A} \star \mathcal{U}_m],$$

$$(3.15) \quad \mathbb{V}_m := [\mathcal{V}_1, \dots, \mathcal{V}_m], \quad \text{and} \quad \mathcal{A}^T \star \mathbb{V}_m := [\mathcal{A}^T \star \mathcal{V}_1, \dots, \mathcal{A}^T \star \mathcal{V}_m].$$

Then, the following proposition can be established.

**PROPOSITION 12.** *The tensors produced by the tensor T-global Golub-Kahan algorithm satisfy the following relations:*

$$(3.16) \quad \mathcal{A} \star \mathbb{U}_m = \mathbb{V}_{m+1} \otimes \tilde{C}_m,$$

$$(3.17) \quad = \mathbb{V}_m \otimes C_m + \beta_{m+1} [\mathcal{O}_{n \times s \times p}, \dots, \mathcal{O}_{n \times s \times p}, \mathcal{V}_{m+1}],$$

$$(3.18) \quad \mathcal{A}^T \star \mathbb{V}_m = \mathbb{U}_m \otimes \tilde{C}_m^T.$$

*Proof.* Using  $\mathcal{A} \star \mathbb{U}_m = [\mathcal{A} \star \mathcal{U}_1, \dots, \mathcal{A} \star \mathcal{U}_m] \in \mathbb{R}^{n \times (sm) \times n_3}$ , the  $(j-1)$ -th frontal slice of  $(\mathcal{A} \star \mathbb{U}_m)$  is given by

$$(\mathcal{A} \star \mathbb{U}_m)_{j-1} = \mathcal{A} \star \mathcal{U}_{j-1} = \alpha_{j-1} \mathcal{V}_{j-1} + \beta_j \mathcal{V}_j.$$

Furthermore, from the definition of the  $\otimes$  product, we have

$$\begin{aligned} (\mathbb{V}_{m+1} \otimes \tilde{C}_m)_{j-1} &= \mathbb{V}_{m+1} \otimes C_{.,j-1}, \\ &= \alpha_{j-1} \mathcal{V}_{j-1} + \beta_j \mathcal{V}_j \end{aligned}$$

and for  $j = m$ ,  $\mathbb{U}_m \otimes \mathcal{C}_{.,m} = \mathcal{A} \star \mathcal{U}_m + \beta_{m+1} \mathcal{V}_{m+1}$  and the result follows. The other relations are proven in a similar way.  $\square$

**4. Application to discrete-ill-posed tensor problems.** In what follows, we will apply the tensor global GMRES and tensor Golub-Kahan algorithms to some discrete-ill-posed problems. We consider the following discrete-ill-posed tensor equation:

$$(4.1) \quad \mathcal{A} \star \mathcal{X} = \mathcal{C}, \quad \mathcal{C} = \widehat{\mathcal{C}} + \mathcal{N},$$

where  $\mathcal{A} \in \mathbb{R}^{n \times n \times p}$ ,  $\mathcal{X}$ ,  $\mathcal{N}$  (additive noise), and  $\mathcal{C}$  are tensors in  $\mathbb{R}^{n \times s \times p}$ .

In color image processing,  $p = 3$ ,  $\mathcal{A}$  represents the blurring tensor,  $\mathcal{C}$  the blurry and noisy observed image,  $\mathcal{X}$  is the image that we would like to restore, and  $\mathcal{N}$  is an unknown additive noise. Therefore, to stabilize the recovered image, regularization techniques are needed. There are several techniques to regularize the linear inverse problem given by equation (4.1); for the matrix case, see, for example, [1, 9, 14, 15]. All of these techniques stabilize the restoration process by adding a regularization term, depending on some priori knowledge of the unknown image. One of the most regularization method is due to Tikhonov and is given as follows:

$$(4.2) \quad \min_{\mathcal{X}} \{ \|\mathcal{A} \star \mathcal{X} - \mathcal{C}\|_F^2 + \mu \|\mathcal{X}\|_F^2 \}.$$

As problem (4.1) is large, Tikhonov regularization (4.2) may be very expensive to solve. One possibility is instead of regularizing the original problem, we apply the Tikhonov technique to the projected problem (3.13) which leads to the following problem:

$$(4.3) \quad y_{m,\mu} = \arg \min_{y \in \mathbb{R}^m} \left( \|\mathcal{R}_0\|_2 - \widetilde{H}_m y\|_2^2 + \mu \|y\|_2^2 \right),$$

$$(4.4) \quad = \arg \min_{y \in \mathbb{R}^m} \left\| \begin{pmatrix} \widetilde{H}_m \\ \mu I_m \end{pmatrix} y - \begin{pmatrix} \beta e_1 \\ 0 \end{pmatrix} \right\|_2.$$

The minimizer  $y_{m,\mu}$  can also be computed as the solution of the following normal equations associated with (4.4)

$$(4.5) \quad \widetilde{H}_{m,\mu} y = \widetilde{H}_m^T \beta e_1, \quad \widetilde{H}_{m,\mu} = (\widetilde{H}_m^T \widetilde{H}_m + \mu^2 I_m).$$

Note that since the Tikhonov problem (4.5) is now a matrix one with small dimension as  $m$  is generally small, the vector  $y_{m,\mu}$ , can thereby be inexpensively computed by some techniques such as the GCV method [14] or the L-curve criterion [9, 11, 15, 16].

In terms of practical implementations, it's more convenient to introduce a restarted version of the tensor Global GMRES. This strategy is essentially based on restarting the tensor T-global Arnoldi algorithm. Therefore, at each restart, the initial guess  $\mathcal{X}_0$  and the regularization parameter  $\mu$  are updated employing the last values computed when the the number of inner iterations required is fulfilled. We note that as the number outer iterations increases it is possible to compute the  $m$ th residual without having to compute extra T-products. At step  $m$ , the residual  $\mathcal{R}_m = \mathcal{C} - \mathcal{A} \star \mathcal{X}_m$  produced by the tensor Global GMRES method for tensor equation (1.1) has the following expression:

$$(4.6) \quad \mathcal{R}_m = \mathbb{V}_{m+1} \circledast (\gamma_{m+1} Q_m e_{m+1}),$$

where  $Q_m$  is the unitary matrix obtained from the QR decomposition of the upper Hessenberg matrix  $\widetilde{H}_m$  and  $\gamma_{m+1}$  is the last component of the vector  $\|\mathcal{R}_0\|_F Q_m^T e_1$  and  $e_{m+1} = (0, 0, \dots, 1)^T \in \mathbb{R}^{m+1}$ . Furthermore, it is easy to show that

$$(4.7) \quad \|\mathcal{R}_m\|_F = |\gamma_{m+1}|.$$

The tensor T-global GMRES method is summarized in the following algorithm.

---

**Algorithm 4** Implementation of Tensor T-global GMRES(m)

---

1. **Input.**  $\mathcal{A} \in \mathbb{R}^{n \times n \times p}$ ,  $\mathcal{V}, \mathcal{B}, \mathcal{X}_0 \in \mathbb{R}^{n \times s \times p}$ , the maximum number of iteration  $\text{Iter}_{\max}$  and a tolerance  $\text{tol} > 0$ .
  2. **Output.**  $\mathcal{X}_m \in \mathbb{R}^{n \times s \times n_3}$  approximate solution of the system (3.1).
  3.  $k = 1, \dots, \text{Iter}_{\max}$ 
    - (a) Compute  $\mathcal{R}_0 = \mathcal{C} - \mathcal{A} \star \mathcal{X}_0$ .
    - (b) Apply Algorithm 2 to compute  $\mathbb{V}_m$  and  $\tilde{H}_m$ .
    - (c) Determine  $\mu_k$  as the parameter minimizing the GCV function [14].
    - (d) Compute the regularized solution  $y_{m_k, \mu}$  of the problem (4.4).
    - (e) Compute the approximate solution  $\mathcal{X}_m = \mathcal{X}_0 + \mathbb{V}_m \otimes y_{m, \mu_k}$
  4. If  $\|\mathcal{R}_m\|_F < \text{tol}$ , stop, else
  5. Set  $\mathcal{X}_0 = \mathcal{X}_m$  and go to 3-a.
  6. End
- 

We turn now to the tensor T-global Golub-Kahan approach for the solving the Tikhonov regularization of the problem (1.1). Here, we apply the following Tikhonov regularization approach and solve the new problem:

$$(4.8) \quad \min_{\mathcal{X}} \{ \|\mathcal{A} \star \mathcal{X} - \mathcal{C}\|_F^2 + \mu^{-1} \|\mathcal{X}\|_F^2 \}.$$

The use of  $\mu^{-1}$  in (4.8) instead of  $\mu$  will be justified below. In the what follows, we briefly review the discrepancy principle approach to determine a suitable regularization parameter, given an approximation of the norm of the additive error. We then assume that a bound  $\epsilon$  for  $\|\mathcal{N}\|_F$  is available. This priori information suggests that  $\mu$  has to be determined as soon as

$$(4.9) \quad \phi(\mu) \leq \eta \epsilon,$$

where  $\phi(\mu) = \|\mathcal{A} \star \mathcal{X} - \mathcal{C}\|_F^2$  and  $\eta \gtrsim 1$  is referred to as the safety factor for the discrepancy principle. A zero-finding method can be used to solve (4.9) in order to find a suitable regularization parameter which also implies that  $\phi(\mu)$  has to be evaluated for several  $\mu$ -values. When the tensor  $\mathcal{A}$  is of moderate size, the quantity  $\phi(\mu)$  can be easily evaluated. This evaluation becomes expensive when the matrix  $\mathcal{A}$  is large, which means that its evaluation by a zero-finding method can be very difficult and computationally expensive. We will approximate  $\phi$  to be able to determine an estimate of  $\|\mathcal{A} \star \mathcal{X} - \mathcal{C}\|_F^2$ . Our approximation is obtained by using T-global Golub-Kahan bidiagonalization (T-GGKB) and Gauss-type quadrature rules. This connection provides approximations of moderate sizes to the quantity  $\phi$ , and therefore gives a solution method to inexpensively solve (4.9) by evaluating these small quantities that can successfully and inexpensively be employed to compute  $\mu$  as well as defining a stopping criterion for the T-GGKB iterations; see [1, 2] for discussion on this method.

Introduce the functions (of  $\mu$ )

$$(4.10) \quad f_\mu(t) := (\mu t + 1)^{-2},$$

$$(4.11) \quad \mathcal{G}_{m, \mu} := \|\mathcal{C}\|_F^2 e_1^T f_\mu(C_m C_m^T) e_1,$$

$$(4.12) \quad \mathcal{R}_{m+1, \mu} := \|\mathcal{C}\|_F^2 e_1^T f_\mu(\tilde{C}_m \tilde{C}_m^T) e_1,$$

The quantities  $\mathcal{G}_{m, \mu}$  and  $\mathcal{R}_{m+1, \mu}$  are referred to as Gauss and Gauss–Radau quadrature rules, respectively, and can be obtained after  $m$  steps of T-GGKB (Algorithm 3) applied to tensor  $\mathcal{A}$  with initial tensor  $\mathcal{C}$ . These quantities approximate  $\phi(\mu)$  as follows:

$$(4.13) \quad \mathcal{G}_{m, \mu} \leq \phi(\mu) \leq \mathcal{R}_{m+1, \mu}.$$

Similarly to the approaches proposed in [1, 2], we therefore instead solve for  $\mu$  the low-dimensional nonlinear equation

$$(4.14) \quad \mathcal{G}_{m,\mu} = \epsilon^2.$$

We apply the Newton's method to solve (4.14) that requires repeated evaluation of the function  $\mathcal{G}_m f_\mu$  and its derivative, which are inexpensive computations for small  $m$ .

We now comment on the use of  $\mu$  in (4.8) instead of  $1/\mu$ , implies that the left-hand side of (4.9) 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 (4.14) for  $m$  sufficiently large; see [1, 2] for analyses. We accept  $\mu_m$  that solve (4.9) as an approximation of  $\mu$ , whenever we have

$$(4.15) \quad \mathcal{R}_{m+1,\mu} \leq \eta^2 \epsilon^2.$$

If (4.15) does not hold for  $\mu_m$ , we carry out one more GGKB steps, replacing  $m$  by  $m + 1$  and solve the nonlinear equation

$$(4.16) \quad \mathcal{G}_{m+1,\mu} = \epsilon^2;$$

see [1, 2] for more details. Assume now that (4.15) holds for some  $\mu_m$ . The corresponding regularized solution is then computed by

$$(4.17) \quad \mathcal{X}_{m,\mu_m} = \mathbb{U}_m \otimes y_{m,\mu_m},$$

where  $y_{m,\mu_m}$  solves

$$(4.18) \quad (\tilde{C}_m^T \tilde{C}_m + \mu_m^{-1} I_m) y = \alpha_1 \tilde{C}_m^T e_1, \quad \alpha_1 = \|\mathcal{C}\|_F.$$

It is also computed by solving the least-squares problem

$$(4.19) \quad \min_{y \in \mathbb{R}^m} \left\| \begin{bmatrix} \mu_m^{1/2} \tilde{C}_m \\ I_m \end{bmatrix} y - \alpha_1 \mu_m^{1/2} e_1 \right\|_2.$$

The following result shows an important property of the approximate solution (4.17). We include a proof for completeness.

PROPOSITION 13. *Let  $\mu_m$  solve (4.14) and let  $y_{m,\mu_m}$  solve (4.19). Then the associated approximate solution (4.17) of (4.8) satisfies*

$$\|\mathcal{A} \star \mathcal{X}_{m,\mu_m} - \mathcal{C}\|_F^2 = \mathcal{R}_{m+1,\mu_m}.$$

*Proof.* From the items of Proposition 12, we get the following matrix low-dimensional least squares problem:

$$\|\mathcal{A} \star \mathcal{X}_{m,\mu_m} - \mathcal{C}\|_F^2 = \left\| \tilde{C}_\ell y_{m,\mu_m} - \alpha_1 e_1 \right\|_2^2.$$

where  $\alpha_1 = \|\mathcal{C}\|_F$ . We now express  $y_{m,\mu_m}$  with the aid of (4.18) and by using the following matrix identity:

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

with  $A$  replaced by  $\tilde{C}_m$ , to obtain

$$\begin{aligned} \|\mathcal{A} \star \mathcal{X}_{m,\mu_m} - \mathcal{C}\|_F^2 &= \alpha_1^2 \left\| e_1 - \tilde{C}_m \left( \tilde{C}_m^T \tilde{C}_m + \mu_m^{-1} I_m \right)^{-1} \tilde{C}_m^T e_1 \right\|_F^2, \\ &= \alpha_1^2 e_1^T \left( \mu_m \tilde{C}_m \tilde{C}_m^T + I_{m+1} \right)^{-2} e_1, \\ &= \mathcal{R}_{m+1,\mu_m}. \end{aligned}$$

The following algorithm summarizes the main steps to compute a regularization parameter and a corresponding regularized solution of (1.1), using Tensor T-GGKB and quadrature rules method for Tikhonov regularization.

---

**Algorithm 5** Tensor T-GGKB and quadrature rules method for Tikhonov regularization

---

1. **Input.**  $\mathcal{A} \in \mathbb{R}^{n \times n \times n_3}$ ,  $\mathcal{C}$ ,  $\eta \gtrsim 1$  and  $\varepsilon$ .
  2. **Output.** T-GGKB steps  $m$ ,  $\mu_m$  and  $X_{m,\mu_m}$ .
  3. Determine the orthonormal bases  $\mathbb{U}_{m+1}$  and  $\mathbb{V}_m$  of tensors, and the bidiagonal  $C_m$  and  $\tilde{C}_m$  matrices with Algorithm 3.
  4. Determine  $\mu_m$  that satisfies (4.14) with Newton's method.
  5. Determine  $y_{m,\mu_m}$  by solving (4.19) and then compute  $X_{m,\mu_m}$  by (4.17).
- 

We comment on the complexity of Algorithms 4 and 5. First, note that the overall computational cost for Algorithm 4 is dominated by the work required to determine  $\mathbb{V}_m$  in Algorithm 2. The computational effort required to determine  $\mathbb{V}_m$  is dominated by the evaluation of  $m$  T-products, which demands approximately,  $O(n_1 n_2 n_3 \log n_3 m)$  flops by using Algorithm 1. Concerning Algorithm 5, the computational complexity is dominated by the work needed to determine  $\mathbb{U}_m$  and  $\mathbb{V}_m$  in Algorithm 3, which demands approximately  $O(n_1 n_2 n_3 \log n_3 m)$  flops. We show in the following section that Algorithms 4 and 5 solve problem (1.1) in a time and cost less than the ones required when using the method proposed in [10] that uses the connection between (standard) Golub-Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations of form (1.1).

**5. Numerical results.** This section performs some numerical tests on the methods of Tensor T-Global GMRES( $m$ ) and Tensor T-Global Golub-Kahan algorithm given by Algorithms 4 and 5, respectively, when applied to the restoration of blurred and noisy color images and videos. For clarity, we only focus on the formulation of a tensor model (4.1), describing the blurring that is taking place in the process of going from the exact to the blurred RGB image. 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 section, we assume that the three channels of the RGB image has the same dimensions, and we refer to it as  $n \times n \times 3$  tensor. Let  $\hat{X}^{(1)}$ ,  $\hat{X}^{(2)}$ , and  $\hat{X}^{(3)}$  be the  $n \times n$  matrices that constitute the three channels of the original error-free color image  $\hat{\mathcal{X}}$ , and  $\hat{C}^{(1)}$ ,  $\hat{C}^{(2)}$ , and  $\hat{C}^{(3)}$  the  $n \times n$  matrices associated with error-free blurred color image  $\hat{\mathcal{C}}$ . Because of some unique features in images, we seek an image restoration model that utilizes blur information, exploiting the spatially invariant properties. Let us also consider that both cross-channel and within-channel blurring take place in the blurring process of the original image. Let  $\text{vec}$  be the operator that transforms a matrix to a vector by stacking the columns of the matrix from left to right. Then, the full blurring model is described by the following form:

$$(5.1) \quad \left( \mathbf{A}_{\text{color}} \otimes \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \right) \hat{\mathbf{x}} = \hat{\mathbf{c}},$$

where,

$$\hat{\mathbf{c}} = \begin{bmatrix} \text{vec}(\hat{\mathbf{C}}^{(1)}) \\ \text{vec}(\hat{\mathbf{C}}^{(2)}) \\ \text{vec}(\hat{\mathbf{C}}^{(3)}) \end{bmatrix}, \quad \hat{\mathbf{x}} = \begin{bmatrix} \text{vec}(\hat{\mathbf{X}}^{(1)}) \\ \text{vec}(\hat{\mathbf{X}}^{(2)}) \\ \text{vec}(\hat{\mathbf{X}}^{(3)}) \end{bmatrix},$$

and

$$\mathbf{A}_{\text{color}} = \begin{bmatrix} a_{rr} & a_{rg} & a_{rb} \\ a_{gr} & a_{gg} & a_{gb} \\ a_{br} & a_{bg} & a_{bb} \end{bmatrix}.$$

$\mathbf{A}_{\text{color}}$  is the  $3 \times 3$  matrix that models the cross-channel blurring, where each row sums to one. This matrix is obtained from [18]. We consider the special case where  $a_{rr} = a_{gg} = a_{bb}$ ,  $a_{gr} = a_{rg}$ ,  $a_{br} = a_{rb}$ , and  $a_{bg} = a_{gb}$ , which gives rise to a cross-channel circular mixing.  $\mathbf{A}^{(1)} \in \mathbb{R}^{n \times n}$  and  $\mathbf{A}^{(2)} \in \mathbb{R}^{n \times n}$  define within-channel blurring and they model the horizontal within blurring and the vertical within blurring matrices, respectively; for more details, see [18]. The notation  $\otimes$  denotes the Kronecker product of matrices; i.e. the Kronecker product of a  $n \times p$  matrix  $A = (a_{ij})$  and a  $(s \times q)$  matrix  $B = (b_{ij})$  is defined as the  $(ns) \times (pq)$  matrix  $A \otimes B = (a_{ij}B)$ . By exploiting the circulant structure of the cross-channel blurring matrix  $\mathbf{A}_{\text{color}}$  and the operators unfold and fold, it can be easily shown that (5.1) can be written in the following tensor form:

$$(5.2) \quad \mathcal{A} \star \hat{\mathbf{X}} \star \mathcal{B} = \hat{\mathbf{C}},$$

where  $\mathcal{A}$  is a 3-way tensor such that  $\mathcal{A}(:, :, 1) = \alpha \mathbf{A}^{(2)}$ ,  $\mathcal{A}(:, :, 2) = \beta \mathbf{A}^{(2)}$ , and  $\mathcal{A}(:, :, 3) = \gamma \mathbf{A}^{(2)}$  and  $\mathcal{B}$  is a 3-way tensor with  $\mathcal{B}(:, :, 1) = (\mathbf{A}^{(1)})^T$ ,  $\mathcal{B}(:, :, 2) = 0$ , and  $\mathcal{B}(:, :, 3) = 0$ . To test the performance of algorithms, the within blurring matrices  $A^{(i)}$  have the following entries:

$$a_{k\ell} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(k-\ell)^2}{2\sigma^2}\right), & |k-\ell| \leq r \\ 0, & \text{otherwise} \end{cases}.$$

Note that  $\sigma$  controls the amount of smoothing, i.e. the larger the  $\sigma$ , the more ill-posed the problem. We generated a blurred and noisy tensor image  $\mathcal{C} = \hat{\mathbf{C}} + \mathcal{N}$ , 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  $\nu := \|\mathcal{N}\|_F / \|\hat{\mathbf{C}}\|_F$ . To determine the effectiveness of our solution methods, we evaluate

$$\text{Relative error} = \frac{\|\hat{\mathbf{X}} - \mathcal{X}_{\text{restored}}\|_F}{\|\hat{\mathbf{X}}\|_F},$$

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

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

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

**5.1. Example 1.** In this example, we present the experimental results recovered by Algorithms 4 and 5 for the reconstruction of a cross-channel blurred color images that have been contaminated by both within and cross blur, and additive noise. The cross-channel blurring is determined by the matrix

$$\mathbf{A}_{\text{color}} = \begin{bmatrix} 0.8 & 0.10 & 0.10 \\ 0.10 & 0.80 & 0.10 \\ 0.10 & 0.10 & 0.80 \end{bmatrix}.$$

We consider two RGB images from **MATLAB**, **papav256** ( $\hat{\mathcal{X}} \in \mathbb{R}^{256 \times 256 \times 3}$ ) and **peppers** ( $\hat{\mathcal{X}} \in \mathbb{R}^{512 \times 512 \times 3}$ ). They are shown in Figure 2. For the within-channel blurring, we let  $\sigma = 4$  and  $r = 6$ . The considered noise levels are  $\nu = 10^{-3}$  and  $\nu = 10^{-2}$ . The associated blurred and noisy RGB images  $\mathcal{C} = \mathcal{A} * \hat{\mathcal{X}} * \mathcal{B} + \mathcal{N}$  for noise level  $\nu = 10^{-3}$  are shown in Figure 3. Given the contaminated RGB image  $\mathcal{C}$ , we would like to recover an approximation of the original RGB image  $\hat{\mathcal{X}}$ . The restorations for noise level  $\nu = 10^{-3}$  are shown in Figure 4 and they are obtained by applying Algorithm 4 implementing the Tensor T-Global GMRES method, with  $\mathcal{X}_0 = \mathcal{O}$ ,  $tol = 10^{-6}$ ,  $m = 10$ , and  $Iter_{\max} = 10$ . Using GCV, the computed optimal value for the projected problem was  $\mu_{10} = 3.82 \times 10^{-5}$ . Table 1 compares the computing time (in seconds), the relative errors, and the SNR of the computed restorations. Note that in this table, the allowed maximum number of outer iterations for Algorithm 4 with noise level  $\nu = 10^{-2}$  was  $Iter_{\max} = 4$  and the maximum number of inner iterations was  $m = 4$ . The restorations obtained with Algorithm 5 are shown in Figure 5. For the **papav256** color image, the discrepancy principle with  $\eta = 1.1$  is satisfied when  $m = 64$  steps of the



Figure 2: Example 1: Original RGB images: **peppers** (left), **papav256** (right).



Figure 3: Example 1: Blurred and noisy images, **peppers** (left), **papav256** (right).



Figure 4: Example 1: Restored images by Algorithm 4, peppers (left), papav256 (right).

Table 1: Results for Example 1.

RGB images	Noise level	Method	SNR	Relative error	CPU time (sec)
papav256	10 <sup>-3</sup>	Algorithm 4	21.01	6.64 × 10 <sup>-2</sup>	6.62
		Algorithm 5	20.41	7.12 × 10 <sup>-2</sup>	5.87
		GKB	20.99	7.12 × 10 <sup>-2</sup>	18.61
	10 <sup>-2</sup>	Algorithm 4	18.00	9.40 × 10 <sup>-2</sup>	1.18
		Algorithm 5	17.78	9.64 × 10 <sup>-2</sup>	1.11
		GKB	17.78	9.64 × 10 <sup>-2</sup>	5.79
peppers	10 <sup>-3</sup>	Algorithm 4	19.39	5.50 × 10 <sup>-2</sup>	24.32
		Algorithm 5	19.11	5.68 × 10 <sup>-2</sup>	25.63
		GKB	19.11	5.68 × 10 <sup>-2</sup>	78.13
	10 <sup>-2</sup>	Algorithm 4	16.23	7.92 × 10 <sup>-2</sup>	4.59
		Algorithm 5	15.61	8.50 × 10 <sup>-2</sup>	3.39
		GKB	15.61	8.50 × 10 <sup>-2</sup>	15.16

Tensor T-GGKB method (Algorithm 3) have been carried out, producing a regularization parameter given by  $\mu_m = 5.57 \times 10^{-5}$ . For comparison with existing approaches in the literature, we report in Table 1 the results obtained by the method proposed in [10]. This method utilizes the connection between (standard) Golub-Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations (5.1), which is equivalent to the tensor problem (5.2). We refer to this method as GKB. It is a solution method based on first reducing  $\mathbf{A}_{\text{color}} \otimes \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)}$  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 (4.10)) to determine an approximation of  $x_\mu$  that satisfies the discrepancy principle associated with the linear problem (5.1). It determines the regularization parameter analogously to Algorithm 5, and uses a similar stopping criterion. The FFT-based computation of the T-product is the only difference in Algorithms 4 and 5. We can see that the methods yield restorations of the same quality, but the new proposed methods perform significantly better in terms of CPU time. This is due to the fact that the T-product operations need fewer flops (approximately  $36mn$  flops) than structure-ignoring evaluation of matrix–vector products with the large matrix  $\mathbf{A}_{\text{color}} \otimes \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)}$  in (5.1) and its transpose, which requires approximately  $2(3m)^2(3n)^2$  flops.



Figure 5: Example 1: Restored images by Algorithm 5, peppers (left), papav256 (right).



Figure 6: Example 2: Original frame no. 5 (left), blurred and noisy frame no. 5 (right).

**5.2. Example 2.** In this example, we evaluate the effectiveness of Algorithms 4 and 5 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. Note that the processing of such given frames, one at a time, is extremely time consuming. 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  $\hat{\mathcal{X}} \in \mathbb{R}^{240 \times 240 \times 30}$ , obtained by stacking the grayscale images that constitute the three channels of each blurred color frame. These frames are blurred by  $\mathcal{A} \star \hat{\mathcal{X}} \star \mathcal{B} = \hat{\mathcal{C}}$ , where  $\mathcal{A}$  and  $\mathcal{B}$  are a 3-way tensors such that  $\mathcal{A}(:, :, 1) = \mathbf{A}^{(2)}$ ,  $\mathcal{B}(:, :, 1) = (\mathbf{A}^{(1)})^T$ , and  $\mathcal{A}(:, :, i) = \mathcal{B}(:, :, i) = 0$ , for  $i = 2, \dots, 30$ , using  $\sigma = 2$  and  $r = 4$  to build the blurring matrices with periodic boundary conditions. This gives rises to block circulant with circulant blocks matrices. We consider white Gaussian noise of levels  $\nu = 10^{-3}$  or  $\nu = 10^{-2}$ . Figure 6 shows the fifth exact (original) frame and the contaminated version with noise level  $\nu = 10^{-3}$ , which is to be restored. Table 2 displays the performance of Algorithms 4 and 5. In Algorithm 4, we have used as an input for noise level  $\nu = 10^{-3}$ ,  $\mathcal{C}$ ,  $\mathcal{X}_0 = \mathcal{O}$ ,  $tol = 10^{-6}$ ,  $m = 10$  and  $Iter_{\max} = 10$ . The chosen inner and outer iterations for noise level  $\nu = 10^{-2}$  were  $m = 4$ , and  $Iter_{\max} = 4$ , respectively. For the 10 outer iterations, minimizing the GCV function produces  $\mu_{10} = 1.15 \times 10^{-5}$ . Using Algorithm 5, the discrepancy principle with  $\eta = 1.1$  have been satisfied after  $m = 59$  steps of T-GGKB method (Algorithm 3), producing a regularization parameter given by  $\mu_m = 1.06 \times 10^{-4}$ . For comparison with existing approaches in the literature, we report in Table 2 the results obtained by the method proposed in [2]. This method utilizes the connection between the Global

Table 2: Results for Example 2.

Noise level	Method	SNR	Relative error	CPU time (second)
$10^{-3}$	Algorithm 4	11.22	$1.09 \times 10^{-1}$	50.22
	Algorithm 5	16.17	$6.31 \times 10^{-2}$	45.22
	GGKB	16.01	$7.56 \times 10^{-2}$	60.10



Figure 7: Example 2: restored frame no. 5 by Algorithm 4 (left), and restored frame no. 5 by Algorithm 5 (right).

Golub-Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations (5.1). We refer to this method as GGKB. It determines the regularization parameter analogously to Algorithm 5, and uses a similar stopping criterion. We can see that Algorithm 5 yields restorations of the same quality as the GGKB method, but the new proposed methods perform significantly better in terms of CPU time. For completeness, the restorations obtained with Algorithms 4 and 5 are shown on the left-hand and the right-hand side of Figure 7, respectively.

**6. Conclusion.** In this paper, we proposed tensor versions of GMRES and Golub-Kahan bidiagonalization algorithms using the T-product, with applications to solving large-scale linear tensor equations arising in the reconstruction of blurred and noisy multichannel images and videos. We also introduced new tensor products as generalizations of other well-known matrix products. The numerical experiments that we have performed show the effectiveness of the proposed schemes to inexpensively compute regularized solutions of high quality.

**Acknowledgment** The authors would like to thank again the two referees for all the valuable remarks and helpful suggestions.

#### REFERENCES

- [1] A.H. Bentbib, M. El Guide, K. Jbilou, and L. Reichel. Global Golub–Kahan bidiagonalization applied to large discrete ill-posed problems. *J. Comput. Appl. Math.*, 322:46–56, 2017.
- [2] A.H. Bentbib, M. El Guide, K. Jbilou, E. Onunwor, and L. Reichel. Solution methods for linear discrete ill-posed problems for color image restoration. *BIT Num. Math.*, 58(3):555–576, 2018.
- [3] K. Braman. Third-order tensors as linear operators on a space of matrices. *Lin. Alg. Appl.*, 433:1241–1253, 2010.
- [4] M. Brazell, N. Li, C. Navasca, and C. Tamon. Solving multilinear systems via tensor inversion. *SIAM J. Matrix Anal. Appl.*, 34(2):542–570, 2013.

- [5] R. Bouyouli, K. Jbilou, R. Sadaka, and H. Sadok. Convergence properties of some block Krylov subspace methods for multiple linear systems. *J. Comput. Appl. Math.*, 196:498–511, 2006.
- [6] F.P.A Beik, F.S. Movahed, and S. Ahmadi-Asl. On the Krylov subspace methods based on tensor format for positive definite Sylvester tensor equations. *Numer. Lin. Alg. Appl.*, 23:444–466, 2016.
- [7] F.P.A. Beik, K. Jbilou, M. Najafi-Kalyani, and L. Reichel. Golub-Kahan bidiagonalization for ill-conditioned tensor equations with applications. *Numer. Algo.*, 84:1535–1563, 2020.
- [8] A. Bouhamidi and K. Jbilou. Sylvester-Tikhonov regularization method for image restoration. *J. Comput. Appl. Math.*, 206:86–98, 2007.
- [9] D. Calvetti, P.C. Hansen, and L. Reichel. L-curve curvature bounds via Lanczos bidiagonalization. *Electron. Trans. Numer. Anal.* 14:134–149, 2002.
- [10] D. Calvetti and L. Reichel. Tikhonov regularization with a solution constraint. *SIAM J. Sci. Comput.*, 26:224–239, 2004.
- [11] D. Calvetti, G.H. Golub, and L. Reichel. Estimation of the L-curve via Lanczos bidiagonalization. *BIT*, 39:603–619, 1999.
- [12] M. El Guide, A. El Ichi, K. Jbilou, and F.P.A Beik, Tensor GMRES and Golub-Kahan Bidiagonalization methods via the Einstein product with applications to image and video processing. arXiv preprint arXiv:2005.07458.
- [13] G.H. Golub and C. F. Van Loan. *Matrix Computations*, 3rd ed. Johns Hopkins University Press, Baltimore, 1996.
- [14] G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.
- [15] P.C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Rev.*, 34:561–580, 1992.
- [16] P.C. Hansen. Regularization tools, a MATLAB package for analysis of discrete regularization problems. *Numer. Algo.*, 6:1–35, 1994.
- [17] N. Hao, M.E. Kilmer, K. Braman, and R.C. Hoover. Facial recognition using tensor-tensor decompositions. *SIAM J. Image Sci.*, 6:437–463, 2013.
- [18] P.C. Hansen, J. Nagy, and D.P. O’Leary. *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.
- [19] K. Jbilou, A. Messaoudi, and H. Sadok. Global FOM and GMRES algorithms for matrix equations. *Appl. Numer. Math.*, 31:49–63, 1999.
- [20] K. Jbilou, H. Sadok, and A. Tinzeft. Oblique projection methods for linear systems with multiple right-hand sides. *Electron. Trans. Numer. Anal.*, 20:119–138, 2005.
- [21] M.N. Kalyani, F.P. A. Beik, and K. Jbilou. On global iterative schemes based on Hessenberg process for (ill-posed) Sylvester tensor equations. *J. Comput. Appl. Math.*, 373:112216, 2020.
- [22] T.G. Kolda and B.W. Bader. Tensor decompositions and Applications. *SIAM Rev.*, 3:455–500, 2009.
- [23] T. Kolda and B. Bader. Higher-order web link analysis using multilinear algebra. In: *Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM 2005*. IEEE Computer Society, 242–249, 2005.
- [24] M.E. Kilmer and C.D. Martin. Factorization strategies for third-order tensors. *Lin. Alg. Appl.*, 435:641–658, 2011.
- [25] M.E. Kilmer, C.D. Martin, and L. Perrone. A third-order generalization of the matrix SVD as a product of third-order tensors. Tech. Report TR-2008-4, Tufts University, Computer Science Department, 2008.
- [26] M.E. Kilmer, K. Braman, N. Hao, and R.C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl.*, 34:148–172, 2013.
- [27] M. Liang and B. Zheng. Further results on Moore-Penrose inverses of tensors with application to tensor nearness problems. *Comput. Math. Appl.*, 77(5):1282–1293, 2019.
- [28] N. Lee and A. Cichocki. Fundamental tensor operations for large-scale data analysis using tensor network formats. *Mult. Sys. Sign. Proc.*, 29:921–960, 2018.
- [29] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Patt. Anal. Mach. Intell.*, 42(4):925–938, 2020.
- [30] L. De Lathauwer and A. de Baynast. Blind deconvolution of DS-CDMA signals by means of decomposition in rank-(l, L, L) terms. *IEEE Trans. Sign. Proc.*, 56:1562–1571, 2008.
- [31] X.-T. Li and M.K. Ng. Solving sparse non-negative tensor equations: Algorithms and applications. *Front. Math. China* 10(3):649–680, 2015.
- [32] Z.-Y. Luo, L.-Q. Qi, and N.-H. Xiu. The sparsest solutions to Z-tensor complementarity problems. *Optim. Lett.*, 11:471–482, 2017.
- [33] Y. Miao, L. Qi, and Y. Wei. Generalized tensor function via the tensor singular value decomposition based on the T-product. *Lin. Alg. Appl.*, 590: 258–303, 2020.
- [34] L.-Q. Qi and Z.-Y. Luo. *Tensor Analysis: Spectral Theory and Special Tensors*. SIAM, Philadelphia, 2017.
- [35] L. Sun, B. Zheng, C. Bu, and Y. Wei. Moore Penrose inverse of tensors via Einstein product. *Lin. Mult. Alg.*, 64:686–698, 2016.
- [36] A.N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math.*, 4:1624–1627, 1963.



- [37] M.A.O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In: *ECCV 2002: Proceedings of the 7th European Conference on Computer Vision*, Lecture Notes in Comput. Sci., vol. 2350. Springer, 447–460, 2002.
- [38] M.A.O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. In: *ICPR 2002: Proceedings of the 16th International Conference on Pattern Recognition*, 511–514, 2002.
- [39] G. Wahba. Practical approximation solutions to linear operator equations when the data are noisy. *SIAM J. Numer. Anal.*, 14:651–667, 1977.