Electronic Transactions on Numerical Analysis.
Volume 55, pp. 92–111, 2022.

ISSN 1068–9613.
DOI: 10.1553/etna_vol55s92

ETNA
Kent State University and
Johann Radon Institute (RICAM)

THE LSQR METHOD FOR SOLVING TENSOR LEAST-SQUARES PROBLEMS*

ABDESLEM H. BENTBIB†, ASMAA KHOUIA‡, AND HASSANE SADOK§

Abstract. In this paper, we are interested in finding an approximate solution $\hat{\mathcal{X}}$ of the tensor least-squares minimization problem $\min_{\mathcal{X}} \left\| \mathcal{X} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} - \mathcal{G} \right\|$, where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ and $A^{(i)} \in \mathbb{R}^{J_i \times I_i}$ $(i = 1, \ldots, N)$ are known and $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the unknown tensor to be approximated. Our approach is based on two steps. Firstly, we apply the CP or HOSVD decomposition to the right-hand side tensor $\mathcal{G}$. Secondly, we perform the well-known Golub-Kahan bidiagonalization for each coefficient matrix $A^{(i)}(i = 1, \ldots, N)$ to obtain a reduced tensor least-squares minimization problem. This type of equations may appear in color image and video restorations as we described below. Some numerical tests are performed to show the effectiveness of our proposed method.

Key words. HOSVD, CP decomposition, color image restoration, video restoration, LSQR

AMS subject classifications. 15A69, 65F

1. Introduction.

The LSQR algorithm of Paige and Sanders [18] is one of the most efficient algorithm for solving the linear system

$$Ax = b$$

or the linear least-squares problem,

$$\min_x \|Ax - b\|_2 \,, \tag{1.1}$$

where A is a matrix of size $m \times n$ and b is a vector of size m. The LSQR method is analytically equivalent to the conjugate gradient method applied to the normal equations associated to (1.1). The LSQR algorithm is based on the Golub-Kahan bidiagonalization procedure [7]. In the past few years, many researchers have employed the LSQR algorithm for solving various equations. For instance, in [19] the authors proposed a global version of LSQR (GL-LSQR) to obtain an approximate solution of the matrix equation $AX = B$, with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times s}$. Recently, a Golub-Kahan bidiagonalization process based on a tensor format was presented in [1, 13] to solve the tensor equation

$$\mathcal{A}(\mathcal{X}) = \mathcal{B},$$

where $\mathcal{X}$ is a tensor of size $n_d \times n_{d-1} \times \cdots \times n_1$ and $\mathcal{A}$ is a linear operator defined by

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{n_d \times n_{d-1} \times \cdots \times n_1} &\longrightarrow \mathbb{R}^{n_d \times n_{d-1} \times \cdots \times n_1} \\ \mathcal{X} &\longrightarrow \mathcal{A}(\mathcal{X}) = \sum_{i=1}^{I} \mathcal{X} \times_1 A_{i,d} \times_2 A_{i,d-1} \times_3 \cdots \times_d A_{i,1}. \end{aligned}$$

Here, $\times_i$, for $i = 1, \ldots, d$, denotes the i-mode product; see below. In those references, the authors used the Golub-Kahan bidiagonalization of the linear operator $\mathcal{A}$. For an extensive survey on the subject of higher-order tensors we refer to [5, 15].

*Received January 15, 2021. Accepted August 16, 2021. Published online on November 5, 2021. Recommended by L. Reichel.
†LAMAI, University Cadi Ayyaad, Marrakech, Morocco (a.bentbib@uca.ac.ma).
‡LAMAI, University Cadi Ayyaad, Marrakech, Morocco. LMPA, University Littoral Cote d'Opale Calais, France (asmaakhouia97@gmail.com).
§LMPA, University Littoral Cote d'Opale Calais, France (sadok@univ-littoral.fr).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

DEFINITION 1.1 ([5, 17]). *Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ be an Nth-order tensor, let $\mathcal{X}_{i_1 i_3 \cdots i_N}$ denote the element $(i_1, i_2, \ldots, i_N)$ of $\mathcal{X}$, and let $U \in \mathbb{R}^{J \times I_n}$ be a matrix. Then, the n-mode product of $\mathcal{X}$ by U, denoted by $\mathcal{X} \times_n U$, is a tensor of size $I_1 \times I_2 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$, whose entries are given by*

$$(\mathcal{X} \times_n U)_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} U_{j i_n}.$$

This paper is concerned with the numerical solution of a tensor least-squares problem of the form

$$\min_{\mathcal{X}} \left\| \mathcal{L}\left(\mathcal{X}\right) - \mathcal{S} \times_1 G^{(1)} \times_2 G^{(2)} \times_3 \cdots \times_N G^{(N)} \right\|. \tag{1.2}$$

Here $\mathcal{L}$ is a linear tensor operator defined by

$$\begin{aligned} \mathcal{L} : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} &\longrightarrow \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N} \\ \mathcal{X} &\longrightarrow \mathcal{X} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)}, \end{aligned}$$

where $\mathcal{S} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_N}$, $G^{(i)} \in \mathbb{R}^{J_i \times m_i}$, and $A^{(i)} \in \mathbb{R}^{J_i \times I_i}$ $(i = 1, \ldots, N)$ are known and $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the unknown tensor to be approximated.

We point out that our approach can be applied to solve general least-squares problems

$$\min_{\mathcal{X}} \|\mathcal{L}(\mathcal{X}) - \mathcal{G}\| \tag{1.3}$$

for an arbitrary right-hand side tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ by decomposing the tensor $\mathcal{G}$ using the CP [11] or the Tucker decomposition [21, 22], better known as the higher-order SVD (HOSVD) [6]. It is easy to verify that, if the right-hand side tensor $\mathcal{G}$ is written in CP or HOSVD format, then the solution $\mathcal{X}$ can also be written in CP or HOSVD format. The tensor least-squares minimization problem (1.3) is a generalization of the equations arising in color image and video restoration; see Section 6. It is not difficult to verify that (1.3) is equivalent to the minimization problem

$$\min_{\mathcal{X}} \|\mathbb{A}\, \mathrm{vec}(\mathcal{X}) - \mathrm{vec}(\mathcal{G})\|,$$

where $\mathbb{A} = A^{(N)} \otimes \cdots \otimes A^{(2)} \otimes A^{(1)}$ and $\otimes$ denotes the Kronecker product. The operator "vec" stacks the columns of a matrix or a tensor to form a vector. If $I_i = J_i$, for $i = 1, \ldots, N$, then the eigenvalues of the matrix $\mathbb{A}$ arise as a product of eigenvalues of the matrices $A^{(i)}$ $(i = 1, \ldots, N)$. The spectrum of $\mathbb{A}$ denoted by $\lambda(\mathbb{A})$ is given by the set

$$\lambda(\mathbb{A}) = \{\lambda_1 \lambda_2 \cdots \lambda_N \ \text{ such that } \ \lambda_i \in \lambda(A^{(i)}),\ i = 1, \ldots, N\};$$

see [12]. This leads to the following result:

LEMMA 1.2. *If $I_i = J_i$, for $i = 1, \ldots, N$, then the solution of the tensor problem (1.3) is unique if and only if*

$$\lambda_1 \lambda_2 \cdots \lambda_N \neq 0,$$

for all $\lambda_i \in \lambda(A^{(i)}),\ i = 1, \ldots, N$.

When $\mathcal{X}$ is an order-2 tensor, that is a matrix X, then the tensor least-squares minimization (1.3) becomes

$$\min_{X} \left\| A^{(1)} X {A^{(2)}}^T - G \right\|.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In this work, we are interested in finding an approximate solution of the tensor least-squares problem (1.2). Our approach is based on performing the well-known Golub-Kahan bidiagonalization for the matrix pair $\left(A^{(i)}, G^{(i)}\right)$, for $i = 1, \ldots, N$. More generally, we are interested in solving the tensor problem (1.3) by writing the right-hand side tensor $\mathcal{G}$ in CP or HOSVD format. Using this approach we can solve the problem (1.3) for higher orders since we are dealing with matrices instead of tensors. In fact, when the dimension increases, the problem becomes harder to handle since the data size of a tensor increases exponentially with the dimensionality of the tensor itself. As a consequence, tensor computations can be extremely expensive and require a large amount of memory. For example, the n-mode product given in Definition 1.1 has a computational complexity of $O\left(J \prod_{\substack{i=1\\ i\neq n}}^{N} I_i\right)$. In addition, by writing the approximate solution in CP or HOSVD decomposition format, we reduce the required memory. For instance, the CP decomposition transforms the storage complexity of an I^N tensors to $O(NRI)$, where R is the CP rank.

The remainder of the paper is organized as follows. In the next section we introduce the notation adopted in this paper and some basic definitions and properties related to tensors. In Section 3, we give a brief introduction to CP and HOSVD decompositions. In Section 4, we construct an approximate solution of the minimization problem (1.3) based on Golub-Kahan bidiagonalization. We work on the coefficient matrices $A^{(i)} (i = 1, \ldots, N)$ by taking the right-hand side tensor $\mathcal{G}$ in rank-one format, and then we generalized it to the case where the right-hand side tensor is approximated using the HOSVD decomposition in Section 5. An example of an application to image and video restoration is given in Section 6. Finally, numerical examples are presented in Section 7 that show the effectiveness of the proposed approach.

2. Notation and preliminary concepts.

In this section, we summarize some of the basic facts about tensors and their computations that will be used in the paper.

DEFINITION 2.1. *The inner product of two same-size tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is given by*

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{A}_{i_1 \cdots i_N} \mathcal{B}_{i_1 \cdots i_N}.$$

It follows immediately that

$$\langle \mathcal{A}, \mathcal{A} \rangle = \| \mathcal{A} \|^2 = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{A}^2_{i_1 \ldots i_N}.$$

DEFINITION 2.2 ([5, 15]). *The n-mode matrix of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, denoted by $\mathcal{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdots I_{n-1} I_{n+1} \cdots I_N)}$, arranges the mode-$n$ fibers into the columns of a matrix. More specifically, we have for, $j = 1 + \sum_{k=1, k\neq n}^{N} (i_k - 1) J_k$ and $J_k = \prod_{m=1, m\neq n}^{k-1} I_m$,*

$$\mathcal{A}_{(n)}(i_n, j) = \mathcal{A}(i_1, i_2, ..., i_N).$$

REMARK 2.3. Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. We can express the n-mode matrix using the slices

$$\begin{aligned}
\mathcal{A}_{(1)} &= \left[\mathcal{A}(:, 1, :), \mathcal{A}(:, 2, :), \ldots, \mathcal{A}(:, I_2, :)\right], \\
\mathcal{A}_{(2)} &= \left[\mathcal{A}(:, :, 1)^T, \mathcal{A}(:, :, 2)^T, \ldots, \mathcal{A}(:, :, I_3)^T\right], \\
\mathcal{A}_{(3)} &= \left[\mathcal{A}(1, :, :), \mathcal{A}(2, :, :), \ldots, \mathcal{A}(I_1, :, :)\right].
\end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

DEFINITION 2.4. *The vectorization of the matrix* $Y \in \mathbb{R}^{I\times T}$ *is defined by*

$$y = \mathrm{vec}(Y) = \left[Y(:,1)^T, Y(:,2)^T, \ldots, Y(:,T)^T\right]^T \in \mathbb{R}^{IT}.$$

Analogously, the vectorization of a tensor $\mathcal{Y}$ *is defined as the vectorization of the associated* 1*-mode unfolded matrix* $\mathcal{Y}_{(1)}$

$$\mathrm{vec}(\mathcal{Y}) = \mathrm{vec}(\mathcal{Y}_{(1)}).$$

PROPOSITION 2.5 ([5, 15]). *Let* $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ *be an* N*th-order tensor,* $V \in \mathbb{R}^{J\times I_n}$*,* $U \in \mathbb{R}^{K\times I_m}$*, and* $W \in \mathbb{R}^{I_n\times I_n}$*. For distinct modes in a series of multiplication, the order of the multiplication is irrelevant, i.e.,*

$$\mathcal{X} \times_m U \times_n V = \mathcal{X} \times_n V \times_m U.$$

If the modes are the same, then

$$\mathcal{X} \times_n V \times_n W = \mathcal{X} \times_n WV.$$

If U *is an orthonormal matrix, then*

$$\|\mathcal{X} \times_m U\| = \|\mathcal{X}\|.$$

PROPOSITION 2.6 ([5, 15]). *Let* $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ *be a* N*th-order tensor and* $\{U_i\}_{1\le i\le N}$ *a set of matrices with* $U_i \in \mathbb{R}^{J_i\times I_i}$*,* $i = 1,\ldots,N$*.*

1. $(\mathcal{X} \times_{i=1}^N U_i)_{(n)} = U_n \mathcal{X}_{(n)} (U_N \otimes \ldots \otimes U_{n+1} \otimes U_{n-1} \otimes \ldots \otimes U_1)^T$.
2. $\mathrm{vec}(\mathcal{X} \times_{i=1}^N U_i) = (U_N \otimes U_{N-1} \otimes \cdots \otimes U_1)\,\mathrm{vec}(\mathcal{X})$.

DEFINITION 2.7 ([5, 15]). *The outer product of the tensors* $\mathcal{Y} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ *and* $\mathcal{X} \in \mathbb{R}^{J_1\times J_2\times\cdots\times J_M}$ *is given by*

$$\mathcal{Z} = \mathcal{Y} \circ \mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N\times J_1\times J_2\times\cdots\times J_M},$$

with

$$\mathcal{Z}_{i_1,i_2,\ldots,i_N,j_1,j_2,\ldots,j_M} = \mathcal{Y}_{i_1,i_2,\ldots,i_N}\mathcal{X}_{j_1,j_2,\ldots,j_M}.$$

As special cases, the outer product of two vectors $a \in \mathbb{R}^I$ *and* $b \in \mathbb{R}^J$ *yields a rank-one matrix*

$$A = a \circ b = ab^T \in \mathbb{R}^{I\times J},$$

and the outer product of three vectors $a \in \mathbb{R}^I$*,* $b \in \mathbb{R}^J$*, and* $c \in \mathbb{R}^Q$ *yields a third-order rank-one tensor*

$$\mathcal{Z} = a \circ b \circ c \in \mathbb{R}^{I\times J\times Q}, \quad \textit{with} \quad z_{i,j,k} = a_i b_j c_k.$$

Using the outer product definition, a tensor of rank one can be defined as follows.

DEFINITION 2.8 ([5, 15]). *A tensor* $\mathcal{X} \in R^{I_1\times I_2\times\cdots\times I_N}$ *of order* N *has rank-one if it can be written as an outer product of* N *vectors, i.e.,*

$$\mathcal{X} = x^{(1)} \circ x^{(2)} \circ \cdots \circ x^{(N)},$$

with $x^{(i)} \in \mathbb{R}^{I_i}$*, for* $i = 1,\ldots,N$*.*

The following definition generalizes the matrix Kronecker product to tensors.

DEFINITION 2.9 ([17]). *The Kronecker product of two tensors* $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *and* $\mathcal{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ *is given by*

$$\mathcal{Z} = \mathcal{Y} \otimes \mathcal{X} \in \mathbb{R}^{I_1 J_1 \times I_2 J_2 \times \cdots \times I_N J_N},$$

with

$$\mathcal{Z}_{k_1,\ldots,k_N} = \mathcal{Y}_{i_1,\ldots,i_N} \mathcal{X}_{j_1,\ldots,j_N}, \quad k_n = j_n + (i_n - 1) J_n, \quad n = 1, \ldots, N.$$

PROPOSITION 2.10. *Let* $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *and* $\mathcal{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ *be two* N*th-order tensors. We have the following result:*

$$\|\mathcal{Y} \otimes \mathcal{X}\| = \|\mathcal{Y}\| \, \|\mathcal{X}\|.$$

Proof. The thesis is easy to verify using Definition 2.9. ∎

PROPOSITION 2.11 ([5]). *Let* $(a_i)_{1 \le i \le N}$ *be a family of* N *vectors of sizes* I_i*, with* $i = 1, \ldots, N$*. Then we have the relation*

$$\operatorname{vec}(a_1 \circ a_2 \circ \cdots \circ a_N) = a_N \otimes a_{N-1} \otimes \cdots \otimes a_1.$$

PROPOSITION 2.12 ([2]). *Let* $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *and* $\mathcal{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ *be two tensors, and let* $U_n \in \mathbb{R}^{K \times I_n}$ *and* $V_n \in \mathbb{R}^{L \times J_n}$*. We have*

$$(\mathcal{Y} \otimes \mathcal{X}) \times_n (U_n \otimes V_n) = (\mathcal{Y} \times_n U_n) \otimes (\mathcal{X} \times_n V_n).$$

3. Tensor decomposition.

In this section, we give a brief introduction to higher-order decompositions. In particular, we focus on two tensor decompositions, the CP decomposition that approximates a tensor as sum of rank-one tensors and the higher-order SVD (HOSVD) decomposition.

3.1. The CP decomposition.

Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an Nth-order tensor. The CP decomposition [5, 11, 14, 15] of $\mathcal{A}$ is given by

$$\mathcal{A} = \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)},$$

where $a_r^{(k)}$ are vectors of size I_k, for $1 \le k \le N$, and R is a positive integer. A CP decomposition of a tensor $\mathcal{A}$ is called an exact CP decomposition if $R = \operatorname{rank}(\mathcal{A})$, where $\operatorname{rank}(\mathcal{A})$ [15] represents the rank of the tensor $\mathcal{A}$ defined as the smallest number of rank-one tensors that generate $\mathcal{A}$ as their sum. Unlike for matrices, where the best rank-R approximation is given by the leading R factors of the SVD, the rank of a specific given tensor is hard to define [10]. In practice, the rank of a tensor is determined numerically by fitting various rank-R CP models. However, an interesting property associated with the CP decomposition for higher-order tensors is the uniqueness under some conditions; see [11, 16].

If we define $A_n = \left[a_1^{(n)} a_2^{(n)} \cdots a_R^{(n)}\right]$, for $n \in \{1, \ldots, N\}$, then the CP decomposition can be symbolically written as

$$\mathcal{A} = A_1 \circ A_2 \circ \cdots \circ A_N,$$

where the matrices $A_n \in \mathbb{R}^{I_n \times R}$ are called factor matrices. Often, the vectors $a_r^{(n)}$ are chosen such that $\|a_r^{(n)}\| = 1$. In this case, the CP decomposition is written as

$$\mathcal{A} = \sum_{r=1}^{R} \lambda_r a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)},$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where λ_r is a scalar that compensates for the magnitudes of the vectors $a_r^{(n)}$. Using the n-mode multiplication of a tensor by a matrix, we obtain the representation

$$\mathcal{A} = \Lambda \times_1 A_1 \times_2 \cdots \times_N A_N,$$

where $\Lambda \in \mathbb{R}^{R \times R \times \cdots \times R}$ is defined by

$$\Lambda_{i_1,\cdots,i_N} = \begin{cases} \lambda_r & \text{for} \quad i_1 = i_2 = \cdots = i_N = r, \\ 0 & \text{otherwise.} \end{cases}$$

For a given integer R, there are many algorithms to compute the CP decomposition. The most popular approach is to apply the alternating least-squares method (ALS); see [4, 11, 15].

3.2. The HOSVD decomposition. There are many decompositions associated with higher-order tensors that generalize the matrix SVD, for example, the higher-order SVD (HOSVD).

DEFINITION 3.1. *Let $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an Nth-order tensor. The Tucker decomposition (often referred to as the higher-order SVD (HOSVD) [6, 22]) of $\mathcal{A}$ is defined by*

$$\mathcal{A} = \mathcal{V} \times_1 U_1 \times_2 U_2 \times_3 \cdots \times_N U_N,$$

where $\mathcal{V} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is the core tensor, $U_n \in \mathbb{R}^{I_n \times R_n}$ are factor matrices, and $(R_1, \ldots, R_N)$ is the multi-linear rank of the tensor $\mathcal{A}$, where $R_n = \operatorname{rank}(\mathcal{A}_{(n)})$. In the case of the HOSVD decomposition, the factor matrices U_n, for $n = 1, \ldots, N$, are orthonormal.

For computing the orthonormal factors U_n, for $n = 1, \ldots, N$, and the tensor $\mathcal{V}$, we compute the SVD associated with each n-mode matrix of the tensor

$$\mathcal{A}_{(n)} = U_n \Sigma_n V_n^T$$

and put

$$\mathcal{V} = \mathcal{A} \times_1 U_1^T \times_2 U_2^T \times_3 \cdots \times_N U_N^T.$$

We point out that a $\operatorname{rank}(r_1, \cdots, r_N)$ approximation, with $r_n \leq R_n$, for $n = 1, \ldots, N$, can be obtained easily by restricting the factor matrices U_n to the first r_n columns (truncated SVD), for $n = 1, \ldots, N$, and by restricting the core tensor $\mathcal{V}$.

4. Rank-one approximation. In the following section we assume that the right-hand side tensor $\mathcal{G}$ in (1.3) is of rank one, which means, it can be written as

$$\mathcal{G} = g^{(1)} \circ g^{(2)} \circ \cdots \circ g^{(N)},$$

where $g^{(i)} \in \mathbb{R}^{J_i}$, for $i = 1, \ldots, N$. Applying a Golub-Kahan bidiagonalization to the pairs $(A^{(i)}, g^{(i)})$, for $i = 1, \ldots, N$, leads to the following relations, for $i = 1, \ldots, N$,

$$U_{k+1}^{(i)}(\beta_1^{(i)} e_1) = g^{(i)}, \quad A^{(i)} V_k^{(i)} = U_{k+1}^{(i)} B_k^{(i)}, \tag{4.1}$$

where $U_{k+1}^{(i)} = [u_1^{(i)}, \ldots, u_{k+1}^{(i)}]$ and $V_k^{(i)} = [v_1^{(i)}, \ldots, v_k^{(i)}]$ are orthonormal bases, e_1 is the first unit vector of $\mathbb{R}^k$, and $B_k^{(i)}$ is a bidiagonal matrix defined as

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$$B_k^{(i)} = \begin{bmatrix} \alpha_1^{(i)} & & & & \\ \beta_2^{(i)} & \alpha_2^{(i)} & & & \\ & \beta_3^{(i)} & \alpha_3^{(i)} & & \\ & & \ddots & \ddots & \\ & & & \beta_k^{(i)} & \alpha_k^{(i)} \\ & & & & \beta_{k+1}^{(i)} \end{bmatrix}.$$

PROPOSITION 4.1. *Let* $\mathcal{X}^{(k)} = \mathcal{Y}_k \times_1 V_k^{(1)} \times_2 V_k^{(2)} \times_3 \cdots \times_N V_k^{(N)}$ *be an approximate solution of* (1.3) *with* $\mathcal{Y}_k \in \mathbb{R}^{k\times k\times\cdots\times k}$. *The corresponding residual* $\mathcal{R}_k$ *can be expressed as*

$$\mathcal{R}_k = (\beta_1^{(1)} e_1 \circ \cdots \circ \beta_1^{(N)} e_1 - \mathcal{Y}_k \times_1 B_k^{(1)} \times_2 \cdots \times_N B_k^{(N)}) \times_1 U_{k+1}^{(1)} \times_2 \cdots \times_N U_{k+1}^{(N)}. \tag{4.2}$$

The residual norm is given by

$$\|\mathcal{R}_k\| = \left\| \beta_1^{(1)} e_1 \circ \cdots \circ \beta_1^{(N)} e_1 - \mathcal{Y}_k \times_1 B_k^{(1)} \times_2 \cdots \times_N B_k^{(N)} \right\|. \tag{4.3}$$

Proof. Using the relations (4.1), we get

$$\begin{aligned} \mathcal{R}_k &= \mathcal{G} - \mathcal{X}_k \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} \\ &= \mathcal{G} - \mathcal{Y}_k \times_1 V_k^{(1)} \times_2 V_k^{(2)} \times_3 \cdots \times_N V_k^{(N)} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} \\ &= \mathcal{G} - \mathcal{Y}_k \times_1 A^{(1)} V_k^{(1)} \times_2 A^{(2)} V_k^{(2)} \times_3 \cdots \times_N A^{(N)} V_k^{(N)} \\ &= \mathcal{G} - \mathcal{Y}_k \times_1 U_{k+1}^{(1)} B_k^{(1)} \times_2 U_{k+1}^{(2)} B_k^{(2)} \times_3 \cdots \times_N U_{k+1}^{(N)} B_k^{(N)} \\ &= \mathcal{G} - \mathcal{Y}_k \times_1 B_k^{(1)} \times_2 B_k^{(2)} \times_3 \cdots \times_N B_k^{(N)} \times_1 U_{k+1}^{(1)} \times_2 U_{k+1}^{(2)} \times_3 \cdots \times_N U_{k+1}^{(N)}. \end{aligned}$$

On the other hand, we have

$$\begin{aligned} \mathcal{G} &= g^{(1)} \circ g^{(2)} \circ \cdots \circ g^{(N)} \\ &= U_{k+1}^{(1)} (\beta_1^{(1)} e_1) \circ U_{k+1}^{(2)} (\beta_1^{(2)} e_1) \circ \cdots \circ U_{k+1}^{(N)} (\beta_1^{(N)} e_1) \\ &= (\beta_1^{(1)} e_1 \circ \beta_1^{(2)} e_1 \circ \cdots \circ \beta_1^{(N)} e_1) \times_1 U_{k+1}^{(1)} \times_2 U_{k+1}^{(2)} \times_3 \cdots \times_N U_{k+1}^{(N)}, \end{aligned}$$

which shows (4.2). Moreover, $U_{k+1}^{(i)}$, for $i = 1, \ldots, N$, are orthonormal matrices, which proves (4.3). ∎

The method determines the tensor $\mathcal{Y}_k$ that minimizes $\|\mathcal{R}_k\|$,

$$\begin{aligned} \mathcal{Y}_k = \arg\min_{\mathcal{Y}} \Big\| & \beta_1^{(1)} e_1 \circ \beta_1^{(2)} e_1 \circ \cdots \circ \beta_1^{(N)} e_1 \\ & - \mathcal{Y} \times_1 B_k^{(1)} \times_2 B_k^{(2)} \times_3 \cdots \times_N B_k^{(N)} \Big\|. \end{aligned} \tag{4.4}$$

This minimization problem is solved using the QR decomposition of each matrix $B_k^{(i)}$, for $i = 1, \ldots, N$,

$$Q_k^{(i)} B_k^{(i)} = \begin{bmatrix} R_k^{(i)} \\ 0 \end{bmatrix}, \qquad Q_k^{(i)} (\beta_1^{(i)} e_1) = \begin{bmatrix} f_k^{(i)} \\ \bar{\phi}_{k+1}^{(i)} \end{bmatrix},$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where the matrix $Q_k^{(i)}$ is a product of k Givens rotation chosen to eliminate the subdiagonal elements $\beta_2^{(i)}, \beta_3^{(i)}, \ldots, \beta_{k+1}^{(i)}$,

$$R_k^{(i)} = \begin{bmatrix} \rho_1^{(i)} & \theta_2^{(i)} & & & \\ & \rho_2^{(i)} & \theta_3^{(i)} & & \\ & & \ddots & \ddots & \\ & & & \rho_{k-1}^{(i)} & \theta_k^{(i)} \\ & & & & \rho_k^{(i)} \end{bmatrix} \quad \text{and} \quad f_k^{(i)} = \begin{bmatrix} \phi_1^{(i)} \\ \phi_2^{(i)} \\ \vdots \\ \phi_{k-1}^{(i)} \\ \phi_k^{(i)} \end{bmatrix}.$$

The minimizer $\mathcal{Y}_k$ of the minimization problem (4.4) can be obtained from the equation

$$\mathcal{Y}_k \times_1 R_k^{(1)} \times_2 R_k^{(2)} \times_3 \cdots \times_N R_k^{(N)} = f_k^{(1)} \circ f_k^{(2)} \circ \cdots \circ f_k^{(N)}.$$

Therefore, an approximate solution is given by

$$\begin{aligned} \mathcal{X}^{(k)} &= \mathcal{Y}_k \times_1 V_k^{(1)} \times_2 V_k^{(2)} \times_3 \cdots \times_N V_k^{(N)} \\ &= V_k{}^{(1)} R_k{}^{(1)^{-1}} f_k{}^{(1)} \circ V_k^{(2)} R_k^{(2)^{-1}} f_k{}^{(2)} \circ \cdots \circ V_k^{(N)} R_k^{(N)^{-1}} f_k^{(N)} \\ &= x_k^{(1)} \circ x_k^{(2)} \circ \cdots \circ x_k^{(N)}, \end{aligned}$$

where $x_k{}^{(i)} = V_k^{(i)} R_k^{(i)^{-1}} f_k^{(i)}$, for $i = 1, \ldots, N$, which can also be written in the form (see [18])

$$\begin{cases} x_k^{(i)} = x_{k-1}^{(i)} + \phi_k^{(i)} d_k^{(i)}, \\ x_0^{(i)} = 0, \end{cases}$$

and $d_k^{(i)}$ can be updated using the expression

$$\begin{cases} d_k^{(i)} = \dfrac{1}{\rho_k^{(i)}} (v_k^{(i)} - \theta_k^{(i)} d_{k-1}^{(i)}), \\ d_0^{(i)} = 0. \end{cases}$$

The following lemma is used to prove Theorem 4.3.

LEMMA 4.2. *Let $\mathcal{F}$ and $\hat{\mathcal{F}}$ be two Nth-order tensors of size $k+1 \times \cdots \times k+1$ defined by*

$$\begin{aligned} \mathcal{F} &= \begin{bmatrix} f_k^{(1)} \\ \bar{\phi}_{k+1}^{(1)} \end{bmatrix} \circ \begin{bmatrix} f_k^{(2)} \\ \bar{\phi}_{k+1}^{(2)} \end{bmatrix} \circ \cdots \circ \begin{bmatrix} f_k^{(N)} \\ \bar{\phi}_{k+1}^{(N)} \end{bmatrix}, \\ \hat{\mathcal{F}} &= \begin{bmatrix} f_k^{(1)} \\ 0 \end{bmatrix} \circ \begin{bmatrix} f_k^{(2)} \\ 0 \end{bmatrix} \circ \cdots \circ \begin{bmatrix} f_k^{(N)} \\ 0 \end{bmatrix}. \end{aligned}$$

Then $\left\langle \mathcal{F}, \hat{\mathcal{F}} \right\rangle = \left\| \hat{\mathcal{F}} \right\|^2$.

Proof. By construction it holds that

$$\left\langle \mathcal{F}, \hat{\mathcal{F}} \right\rangle = \sum_{i_1=1}^{k+1} \sum_{i_2=1}^{k+1} \cdots \sum_{i_N=1}^{k+1} \mathcal{F}_{i_1 \ldots i_N} \hat{\mathcal{F}}_{i_1 \ldots i_N}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

From Definition 2.7, it is easy to verify that

$$\hat{\mathcal{F}}_{i_1\dots i_N} = \begin{cases} \mathcal{F}_{i_1\dots i_N} & \text{for} \quad 1 \le i_1, i_2, \dots, i_N \le k, \\ 0 & \text{otherwise.} \end{cases}$$

Then we have

$$\left\langle \mathcal{F}, \hat{\mathcal{F}} \right\rangle = \sum_{i_1}^{k} \sum_{i_2}^{k} \cdots \sum_{i_N}^{k} \mathcal{F}_{i_1\dots i_N} \mathcal{F}_{i_1\dots i_N} = \sum_{i_1}^{k} \sum_{i_2}^{k} \cdots \sum_{i_N}^{k} \mathcal{F}^2_{i_1\dots i_N} = \left\| \hat{\mathcal{F}} \right\|^2. \qquad \square$$

In the next theorem, we give an upper bound for the residual norm.

THEOREM 4.3. *The residual norm* $||\mathcal{R}_k||^2$ *satisfies the following inequality:*

$$\|\mathcal{R}_k\|^2 \le P_k \sum_{i=1}^{N} \frac{\bar{\phi}^{(i)2}_{k+1}}{\left\| \hat{f}^{(i)}_k \right\|^2},$$

where $\hat{f}^{(i)}_k = \begin{bmatrix} f_k{}^{(i)} \\ \bar{\phi}^{(i)}_{k+1} \end{bmatrix}$ *and* $P_k = \prod_{i=1}^{N} \left\| \hat{f}^{(i)}_k \right\|^2$.

Proof. We have

$$\begin{aligned} \|\mathcal{R}_k\|^2 &= \left\| \mathcal{Y}_k \times_1 B^{(1)}_k \times_2 \cdots \times_N B_k{}^{(N)} - \beta_1{}^{(1)} e_1 \circ \cdots \circ \beta_1{}^{(N)} e_1 \right\|^2 \\ &= \Big\| \Big(\mathcal{Y}_k \times_1 B^{(1)}_k \times_2 \cdots \times_N B_k{}^{(N)} - \beta^{(1)}_1 e_1 \circ \cdots \circ \beta^{(N)}_1 e_1 \Big) \times_1 \\ &\qquad Q_k{}^{(1)} \times_2 \cdots \times_N Q_k{}^{(N)} \Big\|^2 \\ &= \left\| \mathcal{Y}_k \times_1 \begin{bmatrix} R^{(1)}_k \\ 0 \end{bmatrix} \times_2 \cdots \times_N \begin{bmatrix} R^{(N)}_k \\ 0 \end{bmatrix} - \begin{bmatrix} f^{(1)}_k \\ \bar{\phi}^{(1)}_{k+1} \end{bmatrix} \circ \cdots \circ \begin{bmatrix} f^{(N)}_k \\ \bar{\phi}^{(N)}_{k+1} \end{bmatrix} \right\|^2. \end{aligned}$$

Let $\mathcal{Y}_k$ be the solution of the problem

$$\mathcal{Y}_k \times_1 R^{(1)}_k \times_2 R^{(2)}_k \times_3 \cdots \times_N R^{(N)}_k = f^{(1)}_k \circ f^{(2)}_k \circ \cdots \circ f^{(N)}_k.$$

Then

$$\|\mathcal{R}_k\|^2 = \left\| \begin{bmatrix} f^{(1)}_k \\ \bar{\phi}^{(1)}_{k+1} \end{bmatrix} \circ \begin{bmatrix} f^{(2)}_k \\ \bar{\phi}^{(2)}_{k+1} \end{bmatrix} \circ \cdots \circ \begin{bmatrix} f^{(N)}_k \\ \bar{\phi}^{(N)}_{k+1} \end{bmatrix} - \begin{bmatrix} f^{(1)}_k \\ 0 \end{bmatrix} \circ \begin{bmatrix} f^{(2)}_k \\ 0 \end{bmatrix} \circ \cdots \circ \begin{bmatrix} f^{(N)}_k \\ 0 \end{bmatrix} \right\|^2.$$

From Lemma 4.2, the residual norm can be expressed as

$$\begin{aligned} \|\mathcal{R}_k\|^2 &= \left\| \mathcal{F} - \hat{\mathcal{F}} \right\|^2 = \|\mathcal{F}\|^2 - 2 \left\langle \mathcal{F}, \hat{\mathcal{F}} \right\rangle + \left\| \hat{\mathcal{F}} \right\|^2 = \|\mathcal{F}\|^2 - \left\| \hat{\mathcal{F}} \right\|^2 \\ &= \sum_{i_1=1}^{k+1} \sum_{i_2=1}^{k+1} \cdots \sum_{i_N=1}^{k+1} \mathcal{F}^2_{i_1 i_2 \dots i_N} - \sum_{i_1=1}^{k} \sum_{i_2=1}^{k} \cdots \sum_{i_N=1}^{k} \mathcal{F}^2_{i_1 i_2 \dots i_N} \\ &\le \sum_{i_2=1}^{k+1} \cdots \sum_{i_N=1}^{k+1} \mathcal{F}^2_{k+1 i_2 \dots i_N} + \sum_{i_1=1}^{k+1} \cdots \sum_{i_N=1}^{k+1} \mathcal{F}^2_{i_1 k+1 \dots i_N} \\ &\qquad + \cdots + \sum_{i_1=1}^{k+1} \sum_{i_2=1}^{k+1} \cdots \sum_{i_{N-1}=1}^{k+1} \mathcal{F}^2_{i_1 i_2 \dots k+1}. \end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Using Matlab notation, we have

$$\left\|\mathcal{R}_k\right\|^2 \leq \|\mathcal{F}(k+1,:,\ldots,:)\|^2 + \|\mathcal{F}(:,k+1,\ldots,:)\|^2 + \cdots + \|\mathcal{F}(:,:,\ldots,k+1)\|^2 .$$

Since $\mathcal{F}(:,\ldots,\underbrace{k+1}_{\text{i-th index}},\ldots,:)$, for $i=1,\ldots,N$, can be written as

$$\mathcal{F}(:,\ldots,k+1,\ldots,:) = \bar{\phi}_{k+1}^{(i)} \hat{f}_k^{(1)} \circ \cdots \circ \hat{f}_k^{(i-1)} \circ \hat{f}_k^{(i+1)} \circ \cdots \circ \hat{f}_k^{(N)},$$

using Proposition 2.11, we have

$$\operatorname{vec}(\mathcal{F}(:,\ldots,k+1,\ldots,:)) = \bar{\phi}_{k+1}^{(i)} \hat{f}_k^{(N)} \otimes \cdots \otimes \hat{f}_k^{(i+1)} \otimes \hat{f}_k^{(i-1)} \cdots \otimes \hat{f}_k^{(1)}.$$

Therefore, we obtain

$$||\mathcal{R}_k||^2 \leq \sum_{i=1}^{N} \bar{\phi}_{k+1}^{(i)2} \left\|\hat{f}_k^{(1)}\right\|^2 \cdots \left\|\hat{f}_k^{(i-1)}\right\|^2 \left\|\hat{f}_k^{(i+1)}\right\|^2 \cdots \left\|\hat{f}_k^{(N)}\right\|^2 . \qquad \square$$

5. Approximation in the HOSVD format. In this section, we assume that the right-hand side tensor $\mathcal{G}$ of (1.3) is written in the HOSVD format:

$$\mathcal{G} = \mathcal{S} \times_1 G^{(1)} \times_2 \cdots \times_N G^{(N)},$$

where $\mathcal{S} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_N}$ and $G^{(i)} \in \mathbb{R}^{J_i \times m_i}$, for $i=1,\ldots,N$. The CP decomposition is a particular case of an HOSVD, when $m_1 = m_2 = \cdots = m_N = R$ and $\mathcal{S} = \mathcal{I}_R$, where $\mathcal{I}_R$ is the tensor identity. Applying a global Golub-Kahan bidiagonalization [19] to the pairs $(A^{(i)}, G^{(i)})$, for $i=1,\ldots,N$, leads to the relations

$$\mathbb{U}_{k+1}{}^{(i)}(\beta_1{}^{(i)} e_1 \otimes I_{m_i}) = G^{(i)}, \quad A^{(i)} \mathbb{V}_k^{(i)} = \mathbb{U}_{k+1}^{(i)} (B_k^{(i)} \otimes I_{m_i}), \tag{5.1}$$

with $\mathbb{U}_{k+1}^{(i)} = \left[U_1^{(i)}\, U_2^{(i)} \cdots U_{k+1}^{(i)}\right]$ and $\mathbb{V}_k^{(i)} = \left[V_1^{(i)}\, V_2^{(i)} \cdots V_k^{(i)}\right]$ being F-orthonormal.

The method consists in searching an approximate solution of the form

$$\mathcal{X}^{(k)} = \mathcal{Y}_k \times_1 \mathbb{V}_k^{(1)} \times_2 \mathbb{V}_k^{(2)} \times_3 \cdots \times_N \mathbb{V}_k^{(N)},$$

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

$$\begin{aligned} \min_{\mathcal{Y}} \Big\| & \mathcal{S} \times_1 (\beta_1^{(i)} e_1 \otimes I_{m_1}) \times_2 \cdots \times_N (\beta_1^{(N)} e_1 \otimes I_{m_N}) \\ & - \mathcal{Y} \times_1 (B_k^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N (B_k^{(N)} \otimes I_{m_N}) \Big\| . \end{aligned} \tag{5.2}$$

In particular, when $\mathcal{S}$ reduces to $\mathcal{I}_R$ and $m_1 = \ldots = m_N = R$, we obtain the next proposition.

PROPOSITION 5.1. *Let*

$$\mathcal{X}^{(k)} = (\mathcal{Y}_k \otimes \mathcal{I}_R) \times_1 \mathbb{V}_k^{(1)} \times_2 \mathbb{V}_k^{(2)} \cdots \times_N \mathbb{V}_k^{(N)},$$

where $\mathcal{Y}_k \in \mathbb{R}^{k \times k \times \cdots \times k}$ is an approximate solution of (1.3) and where the right-hand side tensor $\mathcal{G}$ is written in CP decomposition format. Then the corresponding residual $\mathcal{R}_k$ can be expressed as

$$\begin{aligned} \mathcal{R}_k = (\beta_1^{(1)} e_1 \circ \cdots \circ \beta_1^{(N)} e_1 - \mathcal{Y}_k \times_1 B_k^{(1)} \times_2 \cdots \times_N B_k^{(N)}) \otimes \\ \mathcal{I}_R \times_1 \mathbb{U}_{k+1}^{(1)} \times_2 \cdots \times_N \mathbb{U}_{k+1}^{(N)}. \end{aligned} \tag{5.3}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In this case, the method determines the tensor $\mathcal{Y}_k$ that solves the minimization problem

$$\mathcal{Y}_k = \arg\min_{\mathcal{Y}} \left\| \beta_1^{(1)} e_1 \circ \beta_1^{(2)} e_1 \circ \cdots \circ \beta_1^{(N)} e_1 - \mathcal{Y} \times_1 B_k^{(1)} \times_2 B_k^{(2)} \times_3 \cdots \times_N B_k^{(N)} \right\|.$$

Proof. Using the relations (5.1), we get

$$\begin{aligned}
\mathcal{R}_k &= \mathcal{G} - \mathcal{X} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} \\
&= \mathcal{G} - (\mathcal{Y}_k \otimes \mathcal{I}_R) \times_1 \mathbb{V}_k^{(1)} \times_2 \mathbb{V}_k^{(2)} \times_3 \cdots \\
&\quad \cdots \times_N \mathbb{V}_k^{(N)} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \cdots \times_N A^{(N)} \\
&= \mathcal{G} - (\mathcal{Y}_k \otimes \mathcal{I}_R) \times_1 \mathbb{U}_{k+1}^{(1)} (B_k^{(1)} \otimes I_R) \times_2 \mathbb{U}_{k+1}^{(2)} (B_k^{(2)} \otimes I_R) \times_3 \cdots \\
&\quad \cdots \times_N \mathbb{U}_{k+1}^{(N)} (B_k^{(N)} \otimes I_R) \\
&= \mathcal{G} - \mathcal{Y}_k \times_1 B_k^{(1)} \times_2 B_k^{(2)} \times_3 \cdots \\
&\quad \cdots \times_N B_k^{(N)} \otimes \mathcal{I}_{\mathcal{R}} \times_1 \mathbb{U}_{k+1}^{(1)} \times_2 \mathbb{U}_{k+1}^{(2)} \times_3 \cdots \times_N \mathbb{U}_{k+1}^{(N)}.
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
\mathcal{G} &= \mathcal{I}_R \times_1 G^{(1)} \times_2 \cdots \times_N G^{(N)} \\
&= \mathcal{I}_R \times_1 \mathbb{U}_{k+1}^{(1)} (\beta_1^{(1)} e_1 \otimes I_R) \times_2 \cdots \times_N \mathbb{U}_{k+1}^{(N)} (\beta_1^{(N)} e_1 \otimes I_R) \\
&= (\beta_1^{(1)} e_1 \circ \cdots \circ \beta_1^{(N)} e_1 \otimes \mathcal{I}_{\mathcal{R}}) \times_1 \mathbb{U}_{k+1}^{(1)} \times_2 \cdots \times_N \mathbb{U}_{k+1}^{(N)},
\end{aligned}$$

which proves (5.3). ∎

The minimization problem (5.2) is solved using the QR decomposition of the matrices $B_k^{(i)}$, for $i = 1, \ldots, N$. Then, the minimizer $\mathcal{Y}_k$ of the problem (5.2) can be obtained from the following equation:

$$\mathcal{Y}_k \times_1 (R_k^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N (R_k^{(N)} \otimes I_{m_N}) = \mathcal{S} \times_1 (f_k^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N (f_k^{(N)} \otimes I_{m_N}).$$

Therefore, an approximate solution is found by

$$\begin{aligned}
\mathcal{X}^{(k)} &= \mathcal{S} \times_1 \mathbb{V}_k^{(1)} \times_2 \cdots \times_N \mathbb{V}_k^{(N)} \\
&= \mathcal{S} \times_1 {\mathbb{V}_k}^{(1)} ({R_k}^{(1)-1} {f_k}^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N \mathbb{V}_k^{(N)} ({R_k}^{(N)-1} {f_k}^{(N)} \otimes I_{m_N}) \\
&= \mathcal{S} \times_1 X_k^{(1)} \times_2 \cdots \times_N X_k^{(N)},
\end{aligned} \tag{5.4}$$

where ${X_k}^{(i)} = {\mathbb{V}_k}^{(i)} ({R_k^{(i)}}^{-1} {f_k}^{(i)} \otimes I_{m_i})$, for $i = 1, \ldots, N$. The matrix $X_k^{(i)}$ can be expressed in the form

$$\begin{cases} X_k^{(i)} = X_{k-1}^{(i)} + \phi_k^{(i)} D_k^{(i)}, \\ X_0^{(i)} = 0, \end{cases}$$

where $D_k^{(i)}$ can be updated using the expression

$$\begin{cases} D_k^{(i)} = \dfrac{1}{\rho_k^{(i)}} (V_k^{(i)} - \theta_k^{(i)} D_{k-1}^{(i)}), \\ D_0^{(i)} = 0. \end{cases}$$

LEMMA 5.2. *Let* $\mathbb{V}_k^{(i)} = \left[V_1^{(i)}\, V_2^{(i)} \cdots V_k^{(i)}\right]$, *with* $V_l^{(i)} \in \mathbb{R}^{I_i \times m_i}$ *for* $l = 1, \ldots, k$, *be an F-orthonormal basis, and let* $\mathcal{X}$ *be an* N*th-order tensors of size* $km_1 \times \cdots \times km_N$, *and* $\mathcal{Z} \in \mathbb{R}^{k \times \cdots \times k}$. *Then*

$$\left\|\mathcal{X} \times_i \mathbb{V}_k^{(i)}\right\| \leqslant \|\mathcal{X}\|, \qquad \left\|(\mathcal{Z} \otimes \mathcal{I}_{m_i}) \times_i \mathbb{V}_k^{(i)}\right\| = \|\mathcal{Z}\|.$$

Proof. See [2]. □

THEOREM 5.3. *The residual norm* $||\mathcal{R}_k||^2$ *satisfies the inequality*

$$\|\mathcal{R}_k\|^2 \leq \|\mathcal{S}\|^2 P_k \sum_{i=1}^{N} \frac{\bar{\phi}_{k+1}^{(i)2}}{\left\|\hat{f}_k^{(i)}\right\|^2},$$

where $\hat{f}_k^{(i)} = \begin{bmatrix} f_k^{(i)} \\ \bar{\phi}_{k+1}^{(i)} \end{bmatrix}$ *and* $P_k = \prod_{i=1}^{N} ||\hat{f}_k^{(i)}||^2$.

Proof. Using the first relation in Lemma 5.2, we have

$$\|\mathcal{R}_k\|^2 \leqslant \Big\|\mathcal{Y}_k \times_1 (B_k^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N (B_k^{(N)} \otimes I_{m_N}) \\ -\mathcal{S} \times_1 (\beta_1^{(i)} e_1 \otimes I_{m_1}) \times_2 \cdots \times_N (\beta_1^{(N)} e_1 \otimes I_{m_N})\Big\|^2.$$

Let

$$Z = \Big\|\mathcal{Y}_k \times_1 (B_k^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N (B_k^{(N)} \otimes I_{m_N}) \\ -\mathcal{S} \times_1 (\beta_1^{(i)} e_1 \otimes I_{m_1}) \times_2 \cdots \times_N (\beta_1^{(N)} e_1 \otimes I_{m_N})\Big\|^2.$$

Thus, we have

$$Z = \left\|\mathcal{Y}_k \times_{i=1}^{N} \left(\begin{bmatrix} R_k^{(i)} \\ 0 \end{bmatrix} \otimes I_{m_i}\right) - \mathcal{S} \times_{i=1}^{N} \left(\begin{bmatrix} f_k^{(i)} \\ \bar{\phi}_{k+1}^{(i)} \end{bmatrix} \otimes I_{m_i}\right)\right\|^2.$$

Let $\mathcal{Y}_k$ be the solution to the following equation:

$$\mathcal{Y} \times_1 (R_k^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N (R_k^{(N)} \otimes I_{m_N}) \\ = \mathcal{S}_k \times_1 (f_k^{(1)} \otimes I_{m_1}) \times_2 \cdots \times_N (f_k^{(N)} \otimes I_{m_N}).$$

Then we have

$$Z = \left\|\mathcal{S} \times_{i=1}^{N} \left(\begin{bmatrix} f_k^{(i)} \\ \bar{\phi}_{k+1}^{(i)} \end{bmatrix} \otimes I_{m_i}\right) - \mathcal{S} \times_{i=1}^{N} \left(\begin{bmatrix} f_k^{(i)} \\ 0 \end{bmatrix} \otimes I_{m_i}\right)\right\|^2.$$

Let

$$\hat{f}^{(i)} = \begin{bmatrix} f_k^{(i)} \\ \bar{\phi}_{k+1}^{(i)} \end{bmatrix} \quad \text{and} \quad f^{(i)} = \begin{bmatrix} f_k^{(i)} \\ 0 \end{bmatrix}, \qquad i = 1, \ldots, N.$$

Using Proposition 2.6, we have

$$
\begin{aligned}
Z &\leq \|\mathcal{S}\|^2 \left\| \left(\hat{f}^{(N)} \otimes I_{m_N}\right) \otimes \cdots \otimes \left(\hat{f}^{(1)} \otimes I_{m_1}\right) \right. \\
&\qquad\qquad\qquad \left. - \left(f^{(N)} \otimes I_{m_N}\right) \otimes \cdots \otimes \left(f^{(1)} \otimes I_{m_1}\right) \right\|_2^2 \\
&= \|\mathcal{S}\|^2 \left\| \left(\hat{f}^{(N)} \otimes \cdots \otimes \hat{f}^{(1)}\right) \otimes \left(I_{m_N} \otimes \cdots \otimes I_{m_1}\right) \right. \\
&\qquad\qquad\qquad \left. - \left(f^{(N)} \otimes \cdots \otimes f^{(1)}\right) \otimes \left(I_{m_N} \otimes \cdots \otimes I_{m_1}\right) \right\|_2^2 \\
&= \|\mathcal{S}\|^2 \left\| \left(\hat{f}^{(N)} \otimes \cdots \otimes \hat{f}^{(1)} - f^{(N)} \otimes \cdots \otimes f^{(1)}\right) \otimes \underbrace{\left(I_{m_N} \otimes \cdots \otimes I_{m_1}\right)}_{I_M} \right\|_2^2 \\
&= \|\mathcal{S}\|^2 \left\| \left(\hat{f}^{(N)} \otimes \cdots \otimes \hat{f}^{(1)} - f^{(N)} \otimes \cdots \otimes f^{(1)}\right) \otimes I_M \right\|_2^2 \\
&= \|\mathcal{S}\|^2 \left\| \hat{f}^{(N)} \otimes \cdots \otimes \hat{f}^{(1)} - f^{(N)} \otimes \cdots \otimes f^{(1)} \right\|_2^2 \\
&= \|\mathcal{S}\|^2 \left\| \operatorname{vec}\left(\hat{f}^{(1)} \circ \cdots \circ \hat{f}^{(N)}\right) - \operatorname{vec}\left(f^{(1)} \circ \cdots \circ f^{(N)}\right) \right\|_2^2 \\
&= \|\mathcal{S}\|^2 \left\| \hat{f}^{(1)} \circ \cdots \circ \hat{f}^{(N)} - f^{(1)} \circ \cdots \circ f^{(N)} \right\|^2 .
\end{aligned}
$$

The proof is completed by applying Theorem 4.3. ∎

The discussed approach is summarized in Algorithm 1 given below.

REMARK 5.4. In line 23 of Algorithm 1, we compute the upper bound of the residual norm $||\mathcal{R}_k||$ given in Theorem 5.3, where $\hat{n}_j^{(i)}$ denotes $\left\|\hat{f}_j^{(i)}\right\|^2$.

6. An example of an application to image and video restoration. In this section, we describe a degradation model associated with color images and videos in the form (1.3). Image restoration is the process of removing blur and noise from a degraded image to recover an approximation of the original image. The well-known mathematical model associated with gray-scale image restoration [3, 8, 9] is formulated as follows:

$$
Kx = b, \quad \text{with} \quad x = \operatorname{vec}(X), \quad \text{and} \quad b = \operatorname{vec}(B),
$$

where $B \in \mathbb{R}^{m\times n}$ is the blurred image, $X \in \mathbb{R}^{m\times n}$ is the true image, and $K \in \mathbb{R}^{mn\times mn}$ is the blurring matrix. The blurring matrix can be determined using both the point spread function (PSF) and the imposed boundary conditions [9]. If the blur is separable, then the blurring matrix can be decomposed as a Kronecker product of two matrices: $K = K_r \otimes K_c$. In this case, the blurring model associated to the restoration of a gray-scale image can be formulated in the form

$$
K_c X K_r^T = B.
$$

In the non-separable case, one can approximate the matrix K by solving the Kronecker product approximation (KPA) problem [23]

$$
(\hat{K_r}, \hat{K_c}) = \underset{K_r, K_c}{\arg\min} \, \|K - K_r \otimes K_c\| . \tag{6.1}
$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 1

Input: Coefficient matrices $A^{(i)}$ for $i = 1, \ldots, N$. The right-hand side tensor $\mathcal{G}$.
Output: An approximate solution $\mathcal{X}_k$ of (1.3).
Decompose $\mathcal{G}$ as $\mathcal{S} \times_1 G^{(1)} \times_2 G^{(2)} \times_3 \cdots \times_N G^{(N)}$
Set $\beta_1{}^{(i)} = \left\|G^{(i)}\right\|$, $U_1{}^{(i)} = \dfrac{1}{\beta_1} G^{(i)}$, $\alpha_1{}^{(i)} = \left\|A^{(i)T} U_1{}^{(i)}\right\|$, and $V_1{}^{(i)} = \dfrac{1}{\alpha_1} A^{(i)T} U_1^{(i)}$
Set $W_1^{(i)} = V_1^{(i)}$, $\bar{\phi}_1^{(i)} = \beta_1^{(i)}$, $\bar{\rho}_1^{(i)} = \alpha_1^{(i)}$
for do $j = 1, 2, 3 \ldots, k$
  for do $i = 1, 2, 3 \ldots, N$
    $\bar{W}_j = A^{(i)} V_j^{(i)} - \alpha_j^{(i)} U_j^{(i)},\ \beta_{j+1}^{(i)} = ||\bar{W}_j^{(i)}||,\ U_{j+1}^{(i)} = \bar{W}_j^{(i)} / \beta_{j+1}^{(i)}$
    $\bar{S}_j^{(i)} = A^{(i)T} U_{j+1}{}^{(i)} - \beta_{j+1}{}^{(i)} V_j{}^{(i)},\ \alpha_{j+1}^{(i)} = ||\bar{S}_j^{(i)}||,\ V_{j+1}{}^{(i)} = \bar{S}_j^{(i)} / \alpha_{j+1}{}^{(i)}$
    $\rho_j{}^{(i)} = \sqrt{\bar{\rho}_j^{(i)2} + \beta_{j+1}{}^{(i)2}}$
    $c_j{}^{(i)} = \bar{\rho}_j^{(i)} / \rho_j{}^{(i)}$
    $s_j^{(i)} = \beta_j^{(i)} / \rho_j^{(i)}$
    $\theta_{j+1}^{(i)} = s_j^{(i)} \alpha_{j+1}^{(i)}$
    $\bar{\rho}_{j+1}^{(i)} = c_j^{(i)} \alpha_{j+1}^{(i)}$
    $\phi_j^{(i)} = c_j^{(i)} \bar{\phi}_j^{(i)}$
    $n_j{}^{(i)} = n_{j-1}{}^{(i)} \phi_j{}^{(i)2}$
    $\bar{\phi}_{j+1}^{(i)} = s_j^{(i)} \bar{\phi}_j^{(i)}$
    $X_j^{(i)} = X_{j-1}^{(i)} + \frac{\phi_j^{(i)}}{\rho_j^{(i)}} W_j^{(i)}$
    $W_{j+1}{}^{(i)} = V_{j+1}{}^{(i)} - \frac{\phi_{i+1}{}^{(i)}}{\rho_i{}^{(i)}} W_j{}^{(i)}$
    $\hat{n}_j^{(i)} = n_j{}^{(i)} + \bar{\phi}_{j+1}^{(i)2}$
  end for
  $\mathcal{X}_j = \mathcal{S} \times_1 X_j{}^{(1)} \times_2 \cdots \times_N X_j{}^{(N)}$
  If $\|\mathcal{S}\| \sqrt{\prod_{i=1}^N \hat{n}_j^{(i)} \sum_{i=1}^N \frac{\bar{\phi}_{j+1}^{(i)2}}{\hat{n}_j^{(i)}}}$ is small enough then stop.
end for

Using RGB format storage, color images are represented by a three-dimensional array of size $m \times n \times 3$. The degradation model in tensor format is given in the following proposition.

PROPOSITION 6.1. *Under the assumption that the blur is the same in all channels, the blurring model associated with color images is given by the tensorial equation*

$$\mathcal{X} \times_1 K_c \times_2 K_r = \mathcal{G}, \tag{6.2}$$

where $\mathcal{X}, \mathcal{G} \in \mathbb{R}^{m \times n \times 3}$ denote the original and the degraded image, respectively.

Proof. The blurring model is described by the equation

$$(I_3 \otimes K) x = g, \tag{6.3}$$

where x and g are defined by

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix}, \quad g = \begin{bmatrix} g^{(1)} \\ g^{(2)} \\ g^{(3)} \end{bmatrix},$$

with $x^{(i)} = \mathrm{vec}(\mathcal{X}(:,:,i))$ and $g^{(i)} = \mathrm{vec}(\mathcal{G}(:,:,i))$, for $i = 1, 2, 3$.
Using the Kronecker approximation (6.1) of the blurring matrix K, (6.3) can be written as

$$(I_3 \otimes K_r \otimes K_c)x = g,$$

which is equivalent to

$$K_c[\mathcal{X}(:,:,1), \mathcal{X}(:,:,2), \mathcal{X}(:,:,3)](I_3 \otimes K_r^T) = [\mathcal{G}(:,:,1), \mathcal{G}(:,:,2), \mathcal{G}(:,:,3)]. \tag{6.4}$$

From (6.4) and using the transpose of each frontal slice, we obtain the equation

$$\begin{aligned} &K_r[\mathcal{X}(:,:,1)^T, \mathcal{X}(:,:,2)^T, \mathcal{X}(:,:,3)^T](I_3 \otimes K_c^T) \\ &\qquad = [\mathcal{G}(:,:,1)^T, \mathcal{G}(:,:,2)^T, \mathcal{G}(:,:,3)^T]. \end{aligned} \tag{6.5}$$

Using Remark 2.3, equation (6.5) can be rewritten in the form

$$K_r \mathcal{X}_{(2)}(I_3 \otimes K_c^T) = \mathcal{G}_{(2)}. \tag{6.6}$$

Using Proposition 2.6, equation (6.6) can be expressed as

$$(\mathcal{X} \times_1 K_c \times_2 K_r \times_3 I_3)_{(2)} = \mathcal{G}_{(2)}.$$

This leads to the following tensor equation,

$$\mathcal{X} \times_1 K_c \times_2 K_r = \mathcal{G}. \qquad \square$$

REMARK 6.2. The blurring equation (6.2) given above represents gray-scale and color videos degradation models, where $\mathcal{X}, \mathcal{G} \in \mathbb{R}^{m \times n \times p}$ are third-order tensors in case of gray-scale videos and $\mathcal{X}, \mathcal{G} \in \mathbb{R}^{m \times n \times 3 \times p}$ are fourth-order tensors in case of color videos, respectively. The dimension p represents the number of frames. Note that the restoration of these frames, one at a time, is extremely time consuming.

7. Numerical examples. In this section, we perform some numerical tests to show the effectiveness of the approach described in this paper. The first part is devoted to the solution of problem (1.3) for given matrices $A^{(i)}$, $i = 1, \dots, N$. In the second part, we present some results for an application to image and video restoration. In order to solve the problem (1.3), we decompose the right-hand side tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ using either the HOSVD or the CP decompositions. In our tables, we use Algorithm 1-CP to denote Algorithm 1, where we decompose the right-hand side tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ using the CP decomposition

$$\mathcal{G} = G^{(1)} \circ G^{(2)} \circ \cdots \circ G^{(N)},$$

where $G^{(i)} \in \mathbb{R}^{J_i \times R}$, for $i = 1, \dots, N$. In addition, we use Algorithm 1-HOSVD to denote Algorithm 1, where $\mathcal{G}$ is decomposed using the HOSVD decomposition

$$\mathcal{G} = \mathcal{S} \times_1 G^{(1)} \times_2 \cdots \times_N G^{(N)},$$

where $\mathcal{S} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_N}$ and $G^{(i)} \in \mathbb{R}^{J_i \times m_i}$. All experiments are performed on a computer with 2.7 GHz Intel(R) Core i5 and 8 GByte using Matlab 2016a. In all the tables, “Iter” stands for the number of iterations.

7.1. Part 1. In this section, we present two numerical examples in order to show the effectiveness of our approach for solving the problem (1.3) for given matrices $A^{(i)}$, $i = 1, \dots, N$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 7.1
Numerical results for Example 1.

N	n	p	R	**Iter**	$\|\mathcal{R}_k\|$	$\|\mathcal{X}_k - \mathcal{X}^\star\|$	**CPU-time(sec)**
3	200	100	5	21	3.14×10^{-8}	7.66×10^{-13}	0.18
	400	300	10	30	1.95×10^{-6}	2.46×10^{-12}	0.78
	500	400	10	30	8.40×10^{-6}	4.37×10^{-12}	5.16
4	50	50	10	23	2.53×10^{-7}	8.16×10^{-12}	0.11
	100	50	20	30	1.92×10^{-7}	3.31×10^{-12}	0.17

Example 1. In the first example, the coefficient matrices $A^{(i)}$, $i = 1, \dots, N$, are generated using the Matlab command

$$A^{(i)} = \texttt{gallery('cycol', [n p], l)},$$

with $l = 20$. In this case, $A^{(i)}$ are $n \times p$ matrices with cyclically repeating columns such that the rank cannot exceed l. We construct the right-hand side tensor so that all the entries of the exact solution $\mathcal{X}^\star$ are equal to one. Table 7.1 displays the obtained results. The used stopping criterion is

$$\|\mathcal{R}_k\| \leq \epsilon,$$

where ϵ is a given tolerance equal to 10^{-6}, and the maximum number of iterations allowed is equal to 30. In this example, we decompose the right-hand side tensor using a CP decomposition.

We point out that the CPU time includes the required time for computing the CP decomposition and the construction of the solution $\mathcal{X}^{(k)}$ in (5.4).

Example 2. In this example, we keep the same data of the previous example except for the coefficient matrices $A^{(i)}$, $i = 1, \dots, N$. They are taken from [20] and have the same size n:

$$A^{(i)} = \texttt{eye}(n) + \frac{1}{2\sqrt{n}}\texttt{rand}(n),$$

where $\texttt{eye}(n)$ and $\texttt{rand}(n)$ are Matlab functions that compute the identity matrix of order n and an $n \times n$ matrix with random entries, respectively. We compared our approach with GLS-BTF described in [19]. The numerical results, which are obtained by applying Algorithm 1, are listed in Table 7.2. For this example, the considered stopping criterion is

$$\frac{\|\mathcal{R}_k\|}{\|\mathcal{G}\|} < 10^{-10},$$

with the maximum number of allowed iterations equal to 160. We point out that the reported CPU-time covers the required time for computing the HOSVD decomposition of the right-hand side tensor and the time for constructing the solution $\mathcal{X}^{(k)}$. In Table 7.2, “error” denotes the upper bound of the residual norm described in Theorem 5.3. In this example, we set $m_i = J_i = n$, for $i = 1, \dots, N$.

Table 7.2 demonstrates the efficiency of our approach especially in terms of execution time. Note that for the GLS-BTF method we did not give the exact error and the residual norm when $N = 3,\ n = 400$ and when $N = 4,\ n = 100$ due to a high CPU-time needed to execute its associated algorithm. In order to show the quality of the bound given in Theorem 5.3, we

TABLE 7.2
Comparison of Algorithm 1 and GLS-BTF.

Method	N	n	**Iter**	$\lVert\mathcal{R}_k\rVert$	error	$\lVert\mathcal{X}_k - \mathcal{X}^\star\rVert$	**CPU-time(sec)**
Alg.1-HOSVD	3	100	27	3.42×10^{-10}	1.34×10^{-9}	3×10^{-11}	0.21
		400	30	4.72×10^{-8}	7.39×10^{-8}	1.39×10^{-9}	17.94
	4	50	27	6.12×10^{-10}	2.81×10^{-9}	4.60×10^{-11}	2.01
		100	27	6.34×10^{-8}	1.33×10^{-7}	6.55×10^{-10}	35
GLS-BTF [19]	3	100	131	4.06×10^{-6}		5.41×10^{-6}	11.71
		400	–	–		–	–
	4	50	160	8.5×10^{-5}		1.18×10^{-4}	106.18
		100	–	–		–	–

TABLE 7.3
Numerical results for Example 2 with n = 1000, 10.000.

N	n	R	**Iter**	error	**CPU-time(sec)**
3	1000	10	25	1.97×10^{-9}	0.22
	10,000	10	25	3.45×10^{-7}	10.27
4	1000	10	30	2.07×10^{-10}	0.34
	10,000	5	30	3.03×10^{-8}	16.13

compare the residual norm and the value of 'error', which represent the upper bound of the residual norm.

We point out that, when solving problem (1.3) for higher dimensions, the approximate solution $\mathcal{X}^{(k)}$ in (5.4) is not explicitly computed, only the coefficient matrices $X_k^{(i)}$, $i = 1, \dots, N$, are constructed. The numerical results are shown in Table 7.3. We keep the same matrices $A^{(i)}$, $i = 1, \dots, N$, defined in this example under the assumption that the right-hand side tensor is written in CP decomposition format

$$\mathcal{G} = G^{(1)} \circ G^{(2)} \circ \cdots \circ G^{(N)},$$

with $G^{(i)} = \mathtt{rand}(J_i, R)$, $i = 1 \dots, N$. The used stopping criterion is

$$\lVert\mathcal{R}_k\rVert \leq \epsilon,$$

where ϵ is a given tolerance equal to 10^{-10} and where the maximal number of iterations is equal to 25 when $N = 3$, and equal to 30 when $N = 4$.

7.2. Part 2: an application to image and video restoration. In this section, we provide some numerical results that illustrate the performance of the approach described in this work applied to the problem of image restoration. To determine the effectiveness of our methods, we evaluate the relative error defined by

$$\text{Relative error} = \frac{||\mathcal{X}^{(k)} - \mathcal{X}_{true}||}{||\mathcal{X}_{true}||},$$

where $\mathcal{X}^{(k)}$ denotes the computed restoration. In addition we evaluate the Signal-to-Noise Ratio (SNR) defined by

$$\text{SNR} = 10\log_{10} \frac{||\mathcal{X}_{true} - E(\mathcal{X}_{true})||^2}{||\mathcal{X}^{(k)} - \mathcal{X}_{true}||^2},$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

(a)

(b)

FIG. 7.1. *Example 1. (a) Exact image. (b) Blurred image.*

TABLE 7.4
Results for Example 1.

$R = m$	**Method**	**Iter**	**SNR**	**Relative error**	**CPU-time(sec)**
100	Algorithm 1-CP	40	19.51	6.32×10^{-2}	1.96
	Algorithm 1-HOSVD	40	22.44	4.51×10^{-2}	0.63
150	Algorithm 1-CP	20	21.77	4.87×10^{-2}	2.75
	Algorithm 1-HOSVD	20	24.23	3.67×10^{-2}	0.54

where $E(\mathcal{X}_{true})$ denotes the mean gray-level of the uncontaminated image $\mathcal{X}_{true}$. In the following examples, we point out that the CPU time covers both the time of the decomposition of the right-hand side tensor $\mathcal{G}$ and the construction of the solution. In the next example, we set $m = m_1 = m_2$ and $m_i = J_i$, for $i = 3, \ldots, N$, with $N = 3$ in the case of color images and gray-scale videos and $N = 4$ in the case of color videos.

7.2.1. Example 1. This example illustrates the performance of Algorithm 1 applied to the restoration of a 3-channel RGB color image that has been contaminated by Gaussian blur, whose point spread function is given by

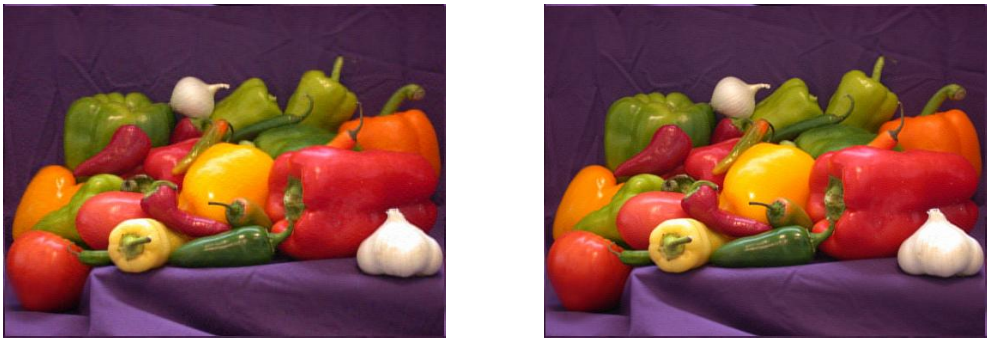
$$k(s,t) = \frac{1}{2\pi\alpha^2}\exp\left\{-\frac{1}{2\alpha^2}\left(s^2+t^2\right)\right\},$$

and by noise (generated by Matlab's `randn` function) with noise level $\nu = 10^{-3}$. This noise level is defined as $\nu = \frac{\|\mathcal{E}\|}{\|\hat{\mathcal{G}}\|}$, where $\mathcal{E}$ is a tensor that represents the noise in $\mathcal{G}$, i.e., $\mathcal{G} := \hat{\mathcal{G}} + \mathcal{E}$, and $\hat{\mathcal{G}}$ is the noise-free image. The true and blurred noisy images of size $388 \times 516 \times 3$ are presented in Figure 7.1. Table 7.4 compares the CPU-time, the relative errors, and the SNR of the computed restorations for a fixed number of iterations.

7.2.2. Example 2. In this example, we illustrate the effectiveness of our approach applied to the restoration of gray-scale and color videos, seen as third-order and fourth-order tensors, respectively. Table 7.5 gives the results obtained after 20 iterations of Algorithm 1. For completeness, Figure 7.3 reports the results obtained from the restoration of a video of size $360 \times 640 \times 30$ using Algorithm 1-HOSVD.

Note that in the last three experiments displayed in Table 7.5, we present only the results associated to the restoration obtained with algorithm 1-HOSVD, due to the time needed to build the approximate solution tensor.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) (b)

FIG. 7.2. *Example 1. Restored images for R=m=150. (a) Approach based on CP decomposition. (b) Approach based on HOSVD decomposition.*

TABLE 7.5
Comparison of the performance of the two approaches for videos restoration of different sizes.

Size	$R = m$	Method	Relative error	CPU-time(sec)
$200 \times 200 \times 30$	150 150	Alg. 1-CP Alg. 1-HOSVD	2.79×10^{-2} 3.12×10^{-2}	1.15 0.25
$360 \times 640 \times 30$	200 200	Alg. 1-CP Alg. 1-HOSVD	7.79×10^{-2} 3.13×10^{-2}	4.56 1.19
$360 \times 640 \times 100$	200	Alg. 1-HOSVD	5.75×10^{-2}	3.49
$200 \times 200 \times 3 \times 20$	150	Alg. 1-HOSVD	3.13×10^{-2}	0.68
$360 \times 640 \times 3 \times 30$	200	Alg. 1-HOSVD	3.17×10^{-2}	6.82

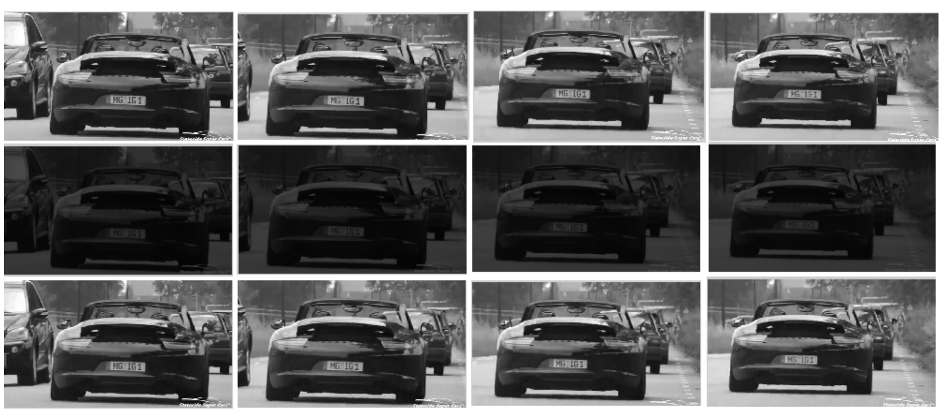


FIG. 7.3. *Example 2. First row: original frames. Second row: blurred frames. Third row: restored frames.*

8. Conclusion. In this work, we proposed a new approach to solve the tensor least-squares minimization problem (1.3). We worked under the assumption that the right-hand side tensor is written (or approximated) using either a CP or a higher-order singular value decomposition (HOSVD) format. Our goal was to solve problem (1.3) for higher dimensions by applying a Golub-Kahan bidiagonalization process to each coefficient matrix $A^{(i)}$, for $i = 1, \ldots, N$, and using an LSQR-like method to construct the approximate solution. The presented numerical examples show the effectiveness of the proposed approach.

REFERENCES

[1] F. P. A. BEIK, K. JBILOU, M. NAJAFI-KALYANI, AND L. REICHEL, *Golub-Kahan bidiagonalization for ill-conditioned tensor equations with applications*, Numer. Algorithms, 84 (2020), pp. 1535–1563.
[2] A. H. BENTBIB, S. EL-HALOUY, AND E. M. SADEK, *Krylov subspace projection method for Sylvester tensor equation with low rank right-hand side*, Numer. Algorithms, 84 (2020), pp. 1411–1430.
[3] D. CALVETTI AND L. REICHEL, *Application of ADI iterative methods to the restoration of noisy images*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 165–186.
[4] J. CARROL AND J. CHANG, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.
[5] A. CICHOCKI, R. ZDUNEK, A. H. PHAN, AND S.-I. AMARI, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*, Wiley, New York, 2009.
[6] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[7] G. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.
[8] C. W. GROETSCH, *Inverse Problems in the Mathematical Sciences*, Vieweg, Braunschweig, 1993.
[9] P. C. HANSEN, J. G. NAGY, D. P. LEARY, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.
[10] J. HÅSTAD, *Tensor rank is NP-complete*, J. Algorithms, 11 (1990), pp. 644–654.
[11] R. A. HARSHMAN, *Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-code factor analysis*, UCLA Working Papers in Phonetics 16, UCLA Linguistics Department, UCLA, Los Angeles, 1970 (84 pages).
[12] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1994.
[13] S. KARIMI AND M. DEHGHAN, *Global least squares method based on tensor form to solve linear systems in Kronecker format*, Trans. Inst. Measur. Contr., 40 (2018), pp. 2378–2386.
[14] H. A. KIERS, *Towards a standardized notation and terminology in multiway analysis*, J. Chemometrics, 14 (2000), pp. 105–122.
[15] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
[16] J. B. KRUSKAL, *Rank, decomposition, and uniqueness for 3-way and N-way arrays*, in Multiway Data Analysis (Rome, 1988), R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 7–18.
[17] N. LEE AND A. CICHOCKI, *Fundamental tensor operations for large-scale data analysis using tensor network formats*, Multidimens. Syst. Signal Process., 29 (2018), pp. 921–960.
[18] C. C. PAIGE AND M. A. SAUNDERS, *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[19] F. TOUTOUNIAN AND S. KARIMI, *Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides*, Appl. Math. Comput., 178 (2006), pp. 452–460.
[20] L. N. TREFETHEN AND D. BAU III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[21] L. R. TUCKER, *Implications of factor analysis of three-way matrices for measurement of change*, in Problems in Measuring Change, C. W. Harris, ed., University of Wisconsin Press, Madison, 1963, pp. 122–137.
[22] ———, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[23] C. F. VAN LOAN AND N. PITSIANIS, *Approximation with Kronecker products*, in Linear Algebra for Large Scale and Real-Time Applications (Leuven, 1992), M. S. Moonen, G. H. Golub, and B. L. R. De Moor, eds., vol. 232 of NATO Adv. Sci. Inst. Ser. E: Appl. Sci., Kluwer, Dordrecht, 1993, pp. 293–314.