LRGEV(-MMP): Left-Right Generalized Eigenvector (Multi-Mode Pencil) Method for Canonical Polyadic Decomposition

Charlotte Vermeylen[1], Nico Vervliet[1], and Lieven De Lathauwer[1,2]
[1]*KU Leuven, Department of Electrical Engineering ESAT–STADIUS, B-3001, Leuven, Belgium*
[2]*KU Leuven–Kulak, Group Science, Engineering and Technology, B-8500, Kortrijk, Belgium*
Email: charlotte.vermeylen@kuleuven.be, nico.vervliet@kuleuven.be, lieven.delathauwer@kuleuven.be

***Abstract*—Algebraic algorithms for canonical polyadic decomposition (CPD) play a key role in many signal processing applications because they are computationally efficient and provide high-quality initializations for optimization-based algorithms. However, a well-known limitation of algebraic algorithms is their sensitivity to noise. Two improvements to the standard generalized eigenvalue decomposition method for CPD are proposed to enhance robustness to noise and improve the accuracy. The first algorithm exploits both the left and right generalized eigenvectors, yielding more accurate factor estimates at a modest increase in computational cost, without changing the asymptotic complexity. The second algorithm further improves accuracy by jointly using information from two matrix pencils across two modes, under the assumption that all three factor matrices have full column rank. Numerical experiments across a range of signal-to-noise ratios show that the proposed methods can achieve up to an order-of-magnitude accuracy improvement on the factors and individual rank-1 term estimates.**

***Index Terms*—Tensors, canonical polyadic decomposition, generalized eigenvalue decomposition, eigenvectors, accuracy, noise**

I. Introduction

Higher-order tensors, or multiway arrays of numerical values, occur in countless applications in signal processing, machine learning, computational mathematics, etc. [1]–[3]. Employing different tensor decomposition models makes it possible to reveal latent signals, uncover meaningful structural patterns, and lower the computational complexity. One of the most interpretable and compact models is the *canonical polyadic decomposition (CPD)* [4], which decomposes a tensor $\mathcal{T} \in \mathbb{C}^{I\times J\times K}$ into a minimal number of rank-1 tensors:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r =: [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!], \tag{1}$$

where '$\otimes$' denotes the tensor or outer product and R is the rank of $\mathcal{T}$. The terms in a CPD are unique under mild conditions on the factors [5], [6]. The uniqueness is the key quality allowing each of the terms to be interpreted. In noisy settings, the data tensor is typically not exactly low rank and the equality in (1) becomes an approximation. Nevertheless, guarantees on the existence of a best low-rank approximation exist [7]. In this work, only tensors whose rank R does not exceed at least two of their dimensions are considered. Without loss of generality, it can be assumed that $I \geq J \geq K$, and consequently, $R \leq J$. The second proposed method requires $R \leq K$.

Algorithms for computing a CPD can be broadly divided into algebraic and optimization-based methods. Algebraic algorithms rely solely on standard linear algebra routines, such as the *singular value decomposition (SVD)*, eigenvalue decomposition, and linear system solving, to obtain the decomposition. While these methods are typically less accurate in noisy settings, they can provide effective initializations for optimization-based algorithms [2], [8]–[10], thereby eliminating the need for multiple random initializations and generally increasing the overall success rate.

The most basic algebraic method makes use of the *generalized eigenvalue decomposition (GEVD)* [11] to recover one factor matrix and computes the remaining ones using best rank-1 approximations of R tensors of order $N-1$. For higher-order tensors, the algorithms first reshape the tensor to a third-order one [12]. Similar methods have been developed in Chemometrics [13], [14]. The GEVD algorithm is applicable if at least two factor matrices have full column rank and the third factor has no proportional columns. In the case of proportional columns, a different tensor decomposition, i.e., an $(L_r, L_r, 1)$-decomposition, is obtained [15].

Due to the reduction of a high-dimensional space to a two-dimensional one in a single mode, pencil-based methods for CPD are unstable [16]. Consequently, small changes to a pencil can cause large errors in the factors. The conditioning of the problem, i.e., the sensitivity of the generalized eigenvalues to noise, can be improved through pencil balancing [17]. More advanced algebraic algorithms circumvent the ill-conditioned step [18], [19]: The *generalized eigenspace decomposition* (GESD) method uses multiple pencils and computes eigenspaces instead of eigenvectors of well-separated subsets of generalized eigenvalues. This is done recursively until individual rank-1 terms are obtained. The improved accuracy comes at a slightly higher computational cost: roughly three times higher compared to the GEVD-based method (estimated experimentally) [18]. Note that a slight increase in computational cost is outweighed by the overall gain of

The following funders supported this work: (1) the Flemish Government, under the AI Research Program. C.V, N.V., and L.D.L. are affiliated to Leuven.AI - KU Leuven Institute for AI, B-3000, Leuven, Belgium. (2) KU Leuven Internal Funds: iBOF/23/064, and C14/22/096.

a better approximation, which can considerably reduce the number of iterations in the optimization step, which has a complexity of $\mathcal{O}\left(RI^3\right)$ flop per iteration, assuming $R \ll I \approx J \approx K$, whereas pencil-based methods scale with $\mathcal{O}\left(R^4\right)$ flop. Lastly, the conditions for uniqueness and algebraic recovery can be relaxed by exploiting rank-1 structure, but at a high computational and memory cost [20].

A challenge of GEVD and GESD is choosing good pencils with well-separated eigenvalues. For GESD, this results in fewer recursive steps. For GEVD, this results in more accurately computed eigenvectors and consequently more accurate factor matrices. The first two slices of the core tensors of the *multilinear singular value decomposition (MLSVD)* [21] give good results in practice (although important exceptions are known, such as CPD with orthogonal factors) [18].

Contribution: Two improved versions of GEVD for CPD are proposed to make the algorithm more robust to noise and competitive with advanced algebraic algorithms, such as GESD. The proposed algorithms leverage information from both the left and right generalized eigenvectors, without requiring matrix inverses or solving linear systems. Numerical experiments confirm that the accuracy of the factors and rank-1 terms is substantially improved. Both 'easy-to-decompose' and more challenging, ill-conditioned tensors are tested, reflecting data characteristics encountered in practical applications.

II. Preliminaries

The following notation is used. Tensors are denoted by calligraphic letters, e.g., $\mathcal{T}$, matrices by bold uppercase letters, e.g., $\mathbf{A}$, and vectors by bold lowercase letters, e.g., $\mathbf{b}$. $\mathbb{F}$ denotes either $\mathbb{C}$ or $\mathbb{R}$. The Khatri–Rao (KR) product or column-wise Kronecker product is denoted by $\odot$. The mode-1 unfolding $\mathbf{T}_{[1;2,3]}$ is a matricization of $\mathcal{T} \in \mathbb{F}^{I\times J\times K}$ to a matrix of size $I \times JK$ such that $\mathcal{T}_{[1;2,3]}(i, j + (k-1)J) = \mathcal{T}(i, j, k)$. Similarly, the mode-2 and mode-3 unfolding are denoted by $\mathcal{T}_{[2;1,3]}$ and $\mathcal{T}_{[3;1,2]}$, respectively. The matricization of $\mathbf{x} \in \mathbb{F}^{IJ}$ is $\mathrm{mat}\,(\mathbf{x}) \in \mathbb{F}^{I\times J}$. The shape is omitted when no confusion is possible. The conjugate is denoted by $\cdot^*$, and the conjugate transpose by $\cdot^{\mathrm{H}}$. The mode-1 multiplication is denoted by $\cdot_1$ and is defined such that $\mathcal{X} = \mathcal{T} \cdot_1 \mathbf{A} \Leftrightarrow \mathbf{X}_{[1;2,3]} = \mathbf{A}\mathbf{T}_{[1;2,3]}$, and similarly for $\cdot_2$ and $\cdot_3$. The mode-n rank is the matrix rank of the mode-n unfolding. The tuple of mode-n ranks forms the multilinear rank (ML-rank). The operator $\mathrm{diag}(\mathbf{x})$ constructs a diagonal matrix with the elements of $\mathbf{x}$ on the diagonal. The vectors $\mathbf{y}$ and $\mathbf{z}$ yielding a best rank-1 approximation of $\mathbf{X}$ in the least-squares sense, i.e., $\mathrm{argmin}_{\mathbf{x},\mathbf{y}}\|\mathbf{X} - \mathbf{x}\mathbf{y}^\top\|_{\mathrm{F}}$, can be obtained from the truncated SVD, denoted by $[\mathbf{y}, \mathbf{z}] = \mathrm{SVD}_1(\mathbf{X})$, where σ_1 is absorbed in $\mathbf{y}$. A mode-n tensor slice is a matrix obtained from a third-order tensor while fixing the index of mode n, e.g., $\mathbf{T}_{::k}$ for $n = 3$ and $\mathbf{T}_{i::}$ for $n = 1$, also called frontal and horizontal slices, respectively.

A. Multilinear singular value decomposition

A common preprocessing step for CPD is to compute a (truncated) MLSVD [21], [22]:

$$\mathcal{T} \approx \mathcal{S} \cdot_1 \mathbf{Q}^{(1)} \cdot_2 \mathbf{Q}^{(2)} \cdot_3 \mathbf{Q}^{(3)},$$

where $\mathbf{Q}^{(i)}$, $i = 1, 2, 3$, are orthogonal matrices, and obtain a CPD of $\mathcal{S}$ instead of the full tensor $\mathcal{T}$ [23], [24] (if $\mathcal{S} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, then $\mathcal{T} = [\![\mathbf{Q}^{(1)}\mathbf{A}, \mathbf{Q}^{(2)}\mathbf{B}, \mathbf{Q}^{(3)}\mathbf{C}]\!]$). The dimension of the core tensor equals the ML-rank of the tensor. Note that the mode-n ranks cannot exceed the rank R or the dimension of mode n. The computational complexity is $\mathcal{O}\left(I^4\right)$ flop, assuming $I \approx J \approx K$.

B. Generalized eigenvalue decomposition

We assume here that the working conditions of GEVD are satisfied when the frontal slices of $\mathcal{T}$ are used for the pencil selection, i.e., that $\mathbf{A}$ and $\mathbf{B}$ are square full rank matrices and that $\mathbf{C}$ has no proportional columns [11], [12]. Otherwise, MLSVD compression can be performed when $R \leq J$, as described above. Each frontal slice can be written as:

$$\mathbf{T}_k := \mathcal{T}_{::k} = \mathbf{A}\mathrm{diag}\left(\mathbf{c}_{k:}\right)\mathbf{B}^\top,$$

and any two slices form a pencil $\left(\mathbf{T}_{k_1}, \mathbf{T}_{k_2}\right)$ of which the left generalized eigenvectors $\mathbf{V}$ are[1]:

$$\mathbf{V}^{\mathrm{H}}\mathbf{T}_{k_1} = \mathbf{D}\mathbf{V}^{\mathrm{H}}\mathbf{T}_{k_2},$$

where $\mathbf{D}$ is a diagonal matrix containing the generalized eigenvalues. It can easily be verified that the following relations between the generalized eigenvectors and eigenvalues, and factor matrices $\mathbf{A}$ and $\mathbf{C}$, respectively, hold:

$$\mathbf{A} = \mathbf{V}^{-\,\mathrm{H}}, \qquad \mathbf{D} = \mathrm{diag}\left(\mathbf{c}_{k_1:}/\mathbf{c}_{k_2:}\right), \tag{2}$$

where the division of the row vectors is elementwise. When the eigenvalues are distinct, the eigenvectors are unique up to scaling with a nonzero constant, and consequently, the factor matrix $\mathbf{A}$ is unique. When $\mathbf{C}$ has no proportional columns, there always exists a pencil with distinct eigenvalues. The scale invariance of the eigenvectors is not an issue, since the factor vectors are scale-invariant as well, since $\alpha_r\mathbf{a} \otimes \beta_r\mathbf{b}_r \otimes \gamma_r\mathbf{c}_r = \mathbf{a} \otimes \mathbf{b}_r \otimes \mathbf{c}_r$ for all $\alpha_r, \beta_r, \gamma_r \neq 0$ that satisfy $\alpha_r\beta_r\gamma_r = 1$.

Because of (2), $\mathbf{V}^*$ can be used as *unmixing matrix* of $\mathcal{T}_{[1;2,3]}$ such that each column corresponds to one term:

$$\mathbf{M}^{(1)} := \mathcal{T}^\top_{[1;2,3]}\mathbf{V}^* = (\mathbf{C} \odot \mathbf{B})\,\mathbf{A}^\top\mathbf{V}^* = (\mathbf{C} \odot \mathbf{B}) \in \mathbb{F}^{R^2\times R}.$$

The rank-1 structure of the reshaped columns of $\mathbf{M}^{(1)}$ can be exploited to determine the columns of $\mathbf{B}$ and $\mathbf{C}$ using R SVDs of $R \times R$ matrices, requiring $\mathcal{O}\left(R^4\right)$ flop. An overview of the standard GEVD algorithm is given in Alg. 1. When $R \ll K$, the compression step dominates the computational complexity.

In numerical experiments, it is typically observed that $\mathbf{B}$ and $\mathbf{C}$ are more accurately computed than $\mathbf{A}$. This is because $\mathbf{A}$ is computed based on the information of two slices, whereas $\mathbf{B}$ and $\mathbf{C}$ use information of the entire tensor. Therefore, practical implementations of GEVD typically include an additional alternating least-squares step on $\mathbf{A}$ [16], [25]. When some eigenvectors are inaccurately computed, which can happen, e.g., due to noise or clustered eigenvalues, taking the inverse of $\mathbf{V}$ or solving the linear system on line 4 can negatively affect

[1]Note that eigenvalue solvers, e.g., `eig` in Matlab, typically return the right eigenvectors first. This can be solved using the pencil $(\mathbf{T}^\top_{k_1}, \mathbf{T}^\top_{k_2})$, instead.

Algorithm 1 Standard GEVD for CPD

Input: $\mathcal{T}$ having ML-rank (R, R, K), with $K \geq 2$

1: $\mathcal{T} \approx \mathcal{S} \cdot_1 \mathbf{Q}^{(1)} \cdot_2 \mathbf{Q}^{(2)} \cdot_3 \mathbf{Q}^{(3)}$ *MLSVD compression*

2: Choose a matrix pencil $(\mathbf{S}_1, \mathbf{S}_2)$ from the tensor slices with distinct generalized eigenvalues

3: $\mathbf{V} \leftarrow \text{eig}(\mathbf{S}_1, \mathbf{S}_2)$ *left generalized eigenvectors*

4: Solve $\mathbf{V}^{\text{H}}\mathbf{A} = \mathbf{Q}^{(1)}$ for $\mathbf{A}$ *linear system (can be problematic)*

5: $\mathbf{M}^{(1)} \leftarrow \mathcal{S}_{[1;2,3]}^{\top}\mathbf{V}^*$ *unmixing of the terms*

6: **for** $r = 1, \ldots, R$ **do**

7: $\quad [\mathbf{b}_r, \mathbf{c}_r] \leftarrow \text{SVD}_1\big(\text{mat}\big(\mathbf{m}_r^{(1)}\big)\big)$

8: $\mathbf{B} \leftarrow \mathbf{Q}^{(2)}\mathbf{B}$ and $\mathbf{C} \leftarrow \mathbf{Q}^{(3)}\mathbf{C}$ *decompression*

Output: $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$

the accuracy of all other terms. This can be understood from theoretical error bounds on perturbed eigenvectors, which are typically inversely related to the eigenvalue gap [26], which is small if the columns are almost proportional; a detailed perturbation analysis is out of the scope of this paper.

III. Improved GEVD-based methods for CPD

The GEVD algorithm has two main issues: it computes $\mathbf{A}$ from two slices instead of using the entire tensor, and as the solution of a linear system. We show that using the whole GEVD and the whole tensor, the accuracy can be improved significantly. Additionally, combining information from two pencils–one from mode-3 slices and one from mode-1 slices–improves the accuracy further, but requires $\mathbf{C}$ to have full column rank as well.

A. Based on one pencil: LRGEV

Firstly, Alg. 1 can be improved by also considering the right generalized eigenvectors of the same pencil $(\mathbf{T}_{k_1}, \mathbf{T}_{k_2})$:

$$\mathbf{T}_{k_1}\mathbf{U} = \mathbf{T}_{k_2}\mathbf{U}\mathbf{D} \qquad \Leftrightarrow \qquad \mathbf{B}^{-1} = \mathbf{U}^{\top}. \tag{3}$$

Both sets of generalized eigenvectors can be computed simultaneously at a slightly higher computational cost [27]. Using (3), the right generalized eigenvectors are then used as unmixing matrix of $\mathcal{T}_{[2;1,3]}$:

$$\mathcal{T}_{[2;1,3]} = \mathbf{B}\left(\mathbf{C} \odot \mathbf{A}\right)^{\top} \Leftrightarrow \mathbf{M}^{(2)} := \mathcal{T}_{[2;1,3]}^{\top}\mathbf{U} = \left(\mathbf{C} \odot \mathbf{A}\right).$$

Furthermore, the third factor matrix can be obtained from

$$\mathbf{c}_{k_1:} = \text{diag}\left(\mathbf{V}^{\text{H}}\mathbf{T}_{k_1}\mathbf{U}\right) \Leftrightarrow \mathbf{C} = \mathcal{T}_{[3;1,2]}\left(\mathbf{U} \odot \mathbf{V}^*\right). \tag{4}$$

In this way, the computation of $\mathbf{C}$ uses information from both the left and right generalized eigenvectors simultaneously. It is experimentally observed that this increases the accuracy, which is expected as the rows of $\mathbf{C}$ play a similar role to the eigenvalues, which are typically more accurate than eigenvectors [26]. Note that $(\mathbf{U} \odot \mathbf{V}^*)$ can be seen as an unmixing matrix of $\mathcal{T}_{[3;1,2]}$ with KR structure on the columns.

The improved algebraic algorithm for CPD, the *left-right generalized eigenvector method (LRGEV)*, is shown in Alg. 2. The above derivation does not consider scaling of the factor matrices. Because $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are estimated from separate equations on lines 7-8 involving different rank-1 approximations, an appropriate rescaling step is required. It is known that $\mathbf{V}^* = \mathbf{A}^{-\top}\mathbf{D}_1$ and $\mathbf{U} = \mathbf{B}^{-\top}\mathbf{D}_2$, where $\mathbf{D}_1$ and $\mathbf{D}_2$ are unknown diagonal full rank scaling matrices. The computed $\hat{\mathbf{C}}$ on line 5 then satisfies

$$\hat{\mathbf{C}} = \mathbf{C}\left(\mathbf{B} \odot \mathbf{A}\right)^{\top}\left(\mathbf{U} \odot \mathbf{V}^*\right) = \mathbf{C}\mathbf{D}_1\mathbf{D}_2.$$

Thus, the scaling can be corrected by dividing the columns of $\hat{\mathbf{C}}$ by the inner products of the eigenvectors and the factor vectors of $\mathbf{A}$ and $\mathbf{B}$; see line 9 in Alg. 2.

Algorithm 2 Left-right generalized eigenvector method (LRGEV)

Input: $\mathcal{T}$ having ML-rank (R, R, K), with $K \geq 2$

1: $\mathcal{T} \approx \mathcal{S} \cdot_1 \mathbf{Q}^{(1)} \cdot_2 \mathbf{Q}^{(2)} \cdot_3 \mathbf{Q}^{(3)}$ *MLSVD compression*

2: Choose a matrix pencil $(\mathbf{S}_1, \mathbf{S}_2)$ from the tensor slices with distinct generalized eigenvalues

3: $[\mathbf{U}, \mathbf{V}] \leftarrow \text{eig}(\mathbf{S}_1, \mathbf{S}_2)$ *left and right generalized eigenvectors*

4: $\mathbf{M}^{(1)} \leftarrow \mathcal{S}_{[2,3;1]}\mathbf{V}^*$ and $\mathbf{M}^{(2)} \leftarrow \mathcal{S}_{[1,3;2]}\mathbf{U}$ *unmixing*

5: $\mathbf{C} \leftarrow \mathcal{S}_{[3;1,2]}\left(\mathbf{U} \odot \mathbf{V}^*\right)$ *unmixing with KR structure*

6: **for** $r = 1, \ldots, R$ **do**

7: $\quad [\mathbf{a}_r, \sim] \leftarrow \text{SVD}_1\big(\text{mat}\big(\mathbf{m}_r^{(2)}\big)\big)$

8: $\quad [\mathbf{b}_r, \sim] \leftarrow \text{SVD}_1\big(\text{mat}\big(\mathbf{m}_r^{(1)}\big)\big)$

9: $\quad \mathbf{c}_r \leftarrow \mathbf{c}_r / \left(\left\langle \mathbf{v}_r^*, \mathbf{a}_r \right\rangle \left\langle \mathbf{u}_r, \mathbf{b}_r \right\rangle\right)$ *correct scale*

10: $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\} \leftarrow \{\mathbf{Q}^{(1)}\mathbf{A}, \mathbf{Q}^{(2)}\mathbf{B}, \mathbf{Q}^{(3)}\mathbf{C}\}$ *decompression*

Output: $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$

B. Based on two pencils across two modes: LRGEV-MMP

A natural extension of LRGEV is to compute $\mathbf{A}$ and $\mathbf{B}$ in the same way as $\mathbf{C}$ in (4). This can be done by additionally considering a pencil $\left(\mathbf{T}'_{i_1}, \mathbf{T}'_{i_2}\right)$ containing mode-1 slices:

$$\mathbf{T}'_i := \mathcal{T}_{i::} := \mathbf{B}\ \text{diag}(\mathbf{a}_{i:})\mathbf{C}^{\top},$$

and computing its right generalized eigenvectors $\mathbf{U}'$. Together with (2)-(3), the following relations are obtained:

$$\mathbf{B}^{\top}\mathbf{U} = \mathbf{I}, \qquad \mathbf{A}^{\top}\mathbf{V}^* = \mathbf{I}, \qquad \mathbf{C}^{\top}\mathbf{U}' = \mathbf{D}_1\mathbf{P}, \tag{5}$$

where $\mathbf{D}_1$ and $\mathbf{P}$ are (unknown) diagonal and permutation matrices, respectively, because the eigenvectors from two pencils can be permuted and scaled. Thus, using (5), $\mathbf{A}$ and $\mathbf{B}$ can be computed using unmixing matrices with KR structure—constructed from the generalized eigenvectors—on the mode-1 and mode-2 unfolding:

$$\mathbf{A} = \mathcal{T}_{[1;2,3]}\left(\mathbf{U}'\mathbf{P}^{\top}\mathbf{D}_1^{-1} \odot \mathbf{U}\right),$$
$$\mathbf{B} = \mathcal{T}_{[2;1,3]}\left(\mathbf{U}'\mathbf{P}^{\top}\mathbf{D}_1^{-1} \odot \mathbf{V}^*\right).$$

Note that, in contrast to Alg. 1 and Alg. 2, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ need to have full column rank to ensure that both the mode-1 and mode-3 slices are square after MLSVD-compression. An overview of the *left-right generalized eigenvector multi-mode pencil method (LRGEV-MMP)* is shown in Alg. 3. $\mathbf{D}_1$ and $\mathbf{P}$ can be computed using a greedy approach, e.g., by iteratively searching for the maximum element of $\mathbf{C}^{\top}\mathbf{U}'$ and setting the

corresponding row and column to zero. In the noiseless case, $\mathbf{P}$ and $\mathbf{D}_1$ are obtained directly from the nonzero elements of $\mathbf{C}^\top\mathbf{U}'$. Note that any other combination of modes can be chosen. However, the eigenvalues of the pencils must still be distinct to ensure recovery of the factors.

Algorithm 3 Left-right generalized eigenvector multi-mode pencil method (LRGEV-MMP)

Input: $\mathcal{T}$ having ML-rank (R, R, R)

1: $\mathcal{T} \approx \mathcal{S} \cdot_1 \mathbf{Q}^{(1)} \cdot_2 \mathbf{Q}^{(2)} \cdot_3 \mathbf{Q}^{(3)}$ *MLSVD compression*
2: Choose $(\mathbf{S}_1, \mathbf{S}_2)$ and $(\mathbf{S}_1', \mathbf{S}_2')$ from mode-3 and mode-1 slices, resp., with distinct generalized eigenvalues
3: $[\mathbf{U}, \mathbf{V}] \leftarrow \text{eig}(\mathbf{S}_1, \mathbf{S}_2)$ *left and right generalized eigenvectors*
4: $[\mathbf{U}', \mathbf{V}'] \leftarrow \text{eig}(\mathbf{S}_1', \mathbf{S}_2')$ *left and right generalized eigenvectors*
5: $\mathbf{C} \leftarrow \mathcal{S}_{[3;1,2]}\,(\mathbf{U} \odot \mathbf{V}^*)$ *unmixing with KR structure*
6: $\mathbf{D}_1\mathbf{P} \approx \mathbf{C}^\top\mathbf{U}'$ *determine scale and permutation*
7: $\mathbf{U}' \leftarrow \mathbf{U}'\mathbf{P}^\top\mathbf{D}_1^{-1}$ *correct scale and permutation*
8: $\mathbf{A} \leftarrow \mathcal{S}_{[1;2,3]}\,(\mathbf{U}' \odot \mathbf{U})$ *unmixing with KR structure*
9: $\mathbf{B} \leftarrow \mathcal{S}_{[2;1,3]}\,(\mathbf{U}' \odot \mathbf{V}^*)$ *unmixing with KR structure*
10: $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\} \leftarrow \{\mathbf{Q}^{(1)}\mathbf{A}, \mathbf{Q}^{(2)}\mathbf{B}, \mathbf{Q}^{(3)}\mathbf{C}\}$ *decompression*

Output: $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$

IV. Numerical experiments

To test the proposed algorithms, 50 complex third-order tensors $\mathcal{T}$ of size $100 \times 100 \times 100$ and rank 30 are generated with random, normally distributed factor matrix entries. Gaussian noise $\mathcal{N}$ is added with a certain signal-to-noise ratio (SNR) $\eta \in \{-10, 0, \cdots, 30\}$ dB: $\eta := 20\log_{10}(\|\mathcal{T}\|_\text{F}/\|\mathcal{N}\|_\text{F})$. All experiments are performed using Tensorlab [25] in Matlab 2023b on a laptop with an AMD Ryzen 7 PRO 7840U processor and 64 GB of RAM. The proposed algorithms LRGEV(-MMP) are compared with GEVD (Alg. 1) and GESD [18] using two accuracy measures. The first is the factor matrix error, computed with the `cpderr` function [25]:

$$\mathbf{e}_\text{fm} := \min_{\mathbf{P},\mathbf{D}_i} \left\{ \frac{\|\mathbf{A} - \hat{\mathbf{A}}\mathbf{P}\mathbf{D}_1\|_\text{F}}{\|\mathbf{A}\|_\text{F}}, \cdots, \frac{\|\mathbf{C} - \hat{\mathbf{C}}\mathbf{P}\mathbf{D}_3\|_\text{F}}{\|\mathbf{C}\|_\text{F}} \right\},$$

where $\{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}\}$ is the computed CPD, and $\mathbf{P}$ and $\mathbf{D}_i$, $i = 1, 2, 3$, are scaling and permutation matrices to match the columns of the ground truth CPD factors $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, as well as possible. The product of the diagonal scaling matrices is not necessarily equal to the identity matrix, i.e., errors on the norms of the rank-1 terms are not taken into account. The second accuracy measure is the relative rank-1 term error:

$$\mathbf{e}_\text{term} := \left[\frac{\left\| [\![\tilde{\mathbf{a}}_r, \tilde{\mathbf{b}}_r, \tilde{\mathbf{c}}_r]\!] - [\![\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r]\!] \right\|_\text{F}}{\|[\![\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r]\!]\|_\text{F}} \right]_{r=1}^{R} \in \mathbb{R}^R,$$

where $\{\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}\} = \{\hat{\mathbf{A}}\mathbf{P}, \hat{\mathbf{B}}\mathbf{P}, \hat{\mathbf{C}}\mathbf{P}\}$ with $\mathbf{P}$ obtained from `cpderr` (without scaling).

The results of the four methods are shown in Fig. 1. As a baseline, the result of the optimization routine `cpd_nls` [8], initialized with the output from LRGEV-MMP, is used. The accuracy of LRGEV(-MMP) is close to that of `cpd_nls`, significantly more accurate than GEVD on the rank-1 terms and on $\mathbf{A}$, and almost as accurate as GESD. As the SNR decreases, all algebraic methods show reduced performance because of the noise sensitivity. GEVD is least accurate on $\mathbf{A}$ because of the computation based on the inverse of the eigenvectors (using the linear system on line 4 in Alg. 1), which also results in a higher mean and max rank-1 error. The computation time of LRGEV(-MMP) and GEVD was the same (0.18 s on average), while the computation time of GESD was slightly higher (0.21 s on average). In contrast, the computation time of `cpd_nls` was 0.85 s on average.

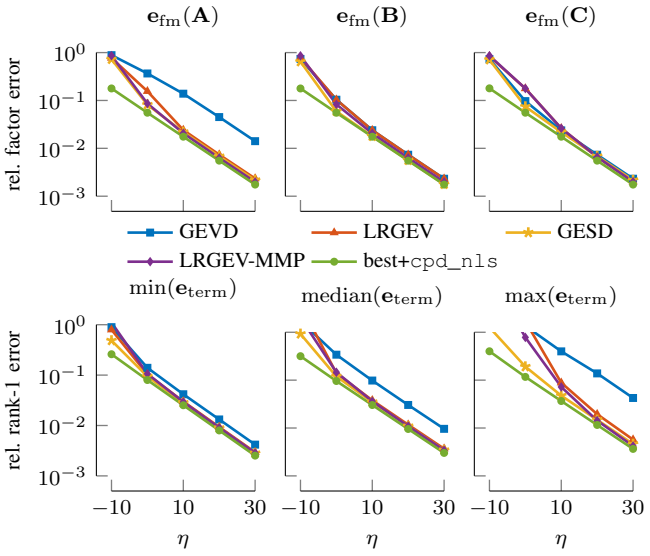


Fig. 1. LRGEV(-MMP) have a significantly lower relative error on $\mathbf{A}$ and the rank-1 terms than GEVD for square third-order tensors of dimension 100 and rank 30. The accuracy is close to that of GESD and numerical optimization. Median values over 50 experiments per SNR value are reported.

The second experiment simulates situations where the columns of $\mathbf{A}$ and $\mathbf{B}$ are similar, i.e., having a small angle between them, increasing the condition number of $\mathbf{A}$ and $\mathbf{B}$. This occurs frequently in practice, e.g., when unmixing or estimating signals originating from sources close together. The angle is set to approximately $10°$ using the Simplex algorithm and a random change of basis. Again, the SNR value is varied, but, because of the higher condition number of the factors, between $\eta \in \{20, 40, \ldots, 100\}$. For lower SNR values, no accurate digits remain, even for `cpd_nls`, indicating the difficulty. Compared to the previous experiment, it can be seen in Fig. 2 that LRGEV-MMP is more accurate than LRGEV, certainly in terms of the maximum rank-1 error, thanks to the multi-mode pencil strategy, and that they are both more accurate than GESD. GESD computes $\mathbf{B}$ using an alternating least-squares step, destroying the accuracy of all terms in the case of a few inaccurate eigenvectors, occurring due to ill-conditioning. Note that since $\mathbf{A}$ and $\mathbf{B}$ have a higher condition number than $\mathbf{C}$, $\mathbf{U}$ and $\mathbf{V}$ will also have a higher condition number than $\mathbf{U}'$. Consequently, $\mathbf{C}$ on line 5 in Alg. 3 is computed using the KR product of two ill-conditioned

matrices, increasing the numerical errors, whereas $\mathbf{A}$ and $\mathbf{B}$ on lines 8-9 are computed using only one ill-conditioned matrix in the KR product, resulting in a smaller error than on $\mathbf{C}$. Furthermore, it is observed that the other methods, including `cpd_nls`, also have a higher error on $\mathbf{C}$.

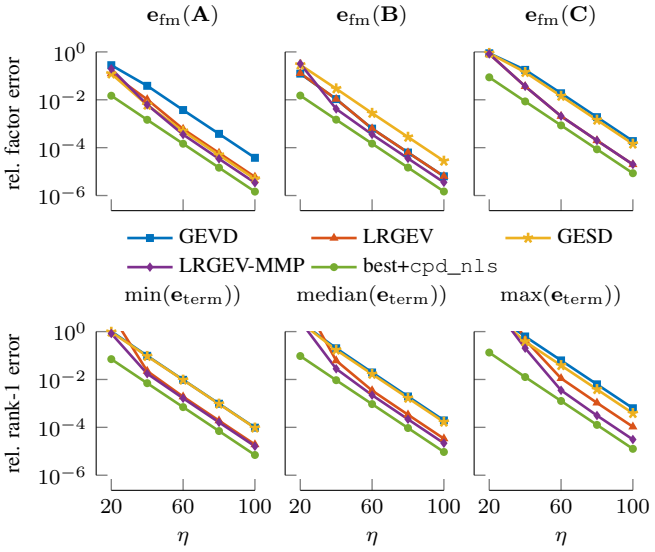


Fig. 2. LRGEV(-MMP) are significantly more accurate than GEVD and GESD across all six measures on the rank-1 and factor matrix error. They ensure a close-to-optimal factor matrix and rank-1 error with a median and maximum value close to the minimum one. A similar setup as in Fig. 1 is used but with almost collinear columns in $\mathbf{A}$ and $\mathbf{B}$ ($10°$).

V. Conclusion

Two novel algebraic algorithms LRGEV and LRGEV-MMP for CPD are proposed using the generalized left and right eigenvectors of one pencil or multiple pencils across different modes. The factor matrices are computed using all tensor slices to improve the accuracy and without solving linear systems or computing inverses, which would otherwise destroy the accuracy when some eigenvectors are inaccurate due to noise or clustered eigenvalues. LRGEV-MMP is most accurate but requires all factor matrices to have full rank, a requirement that less is relaxed but not unrealistic in practice. LRGEV(-MMP) are compared with GEVD and GESD under different SNR values on well- and ill-conditioned tensors. Both are shown to outperform GEVD on the accuracy of individual terms and the different factors. The obtained accuracy is close to that of optimization-based algorithms.

References

[1] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Trans. Intell. Syst. Technol.*, vol. 8, Oct. 2016.

[2] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] A. Cichocki, D. Mandic, A.-H. Phan, C. Caiafa, G. Zhou, Q. Zhao, and L. De Lathauwer, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[4] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-model factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[5] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, no. 2, pp. 95–138, 1977.

[6] I. Domanov and L. De Lathauwer, "Generic uniqueness of a structured matrix factorization and applications in blind source separation," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 701–711, 2016.

[7] E. Evert and L. De Lathauwer, "Guarantees for existence of a best canonical polyadic approximation of a noisy low-rank tensor," *SIAM J. Matrix Anal. Appl.*, vol. 43, no. 1, pp. 328–369, 2022.

[8] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 695–720, 2013.

[9] C. Vermeylen, N. Vervliet, and L. De Lathauwer, "Reducing swamp behavior for the canonical polyadic decomposition problem by rank-1 freezing," *Numer. Algorithms*, vol. 100, pp. 831–859, 2025.

[10] E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *J. Chemometrics*, vol. 25, no. 2, pp. 67–86, 2011.

[11] S. E. Leurgans, R. T. Ross, and R. B. Abel, "A decomposition for three-way arrays," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 4, pp. 1064–1083, 1993.

[12] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 2, pp. 636–660, 2014.

[13] E. Sanchez and B. R. Kowalski, "Generalized rank annihilation factor analysis," *Analytical Chemistry*, vol. 58, no. 2, pp. 496–499, 1986.

[14] E. Sanchez and B. R. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *J. Chemometrics*, vol. 4, no. 1, pp. 29–45, 1990.

[15] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1033–1066, 2008.

[16] C. Beltrán, P. Breiding, and N. Vannieuwenhoven, "Pencil-based algorithms for tensor rank decomposition are not stable," *SIAM J. Matrix Anal. Appl.*, vol. 40, no. 2, pp. 739–773, 2019.

[17] D. Lemonnier and P. Van Dooren, "Balancing regular matrix pencils," *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 1, pp. 253–263, 2006.

[18] E. Evert, M. Vandecappelle, and L. De Lathauwer, "A recursive eigenspace computation for the canonical polyadic decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 43, no. 1, pp. 274–300, 2022.

[19] E. Evert, M. Vandecappelle, and L. De Lathauwer, "Canonical polyadic decomposition via the generalized Schur decomposition," *IEEE Signal Process. Lett.*, vol. 29, pp. 937–941, 2022.

[20] I. Domanov and L. De Lathauwer, "Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm," *Linear Algebra Appl.*, vol. 513, pp. 342–375, 2017.

[21] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[22] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. A1027–A1052, 2012.

[23] J. D. Carroll, S. Pruzansky, and J. B. Kruskal, "Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters," *Psychometrika*, vol. 45, no. 1, pp. 3–24, 1980.

[24] R. Bro and C. A. Andersson, "Improving the speed of multiway algorithms: Part II: Compression," *Chemometrics and Intelligent Laboratory Systems*, vol. 42, no. 1, pp. 105–113, 1998.

[25] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab 3.0," March 2016. Available online at www.tensorlab.net.

[26] A. Greenbaum, R.-C. Li, and M. L. Overton, "First-order perturbation theory for eigenvalues and eigenvectors," *SIAM Rev.*, vol. 62, no. 2, pp. 463–482, 2020.

[27] G. H. Golub and C. F. Van Loan, *Matrix Computations - 4th Edition*. Philadelphia, PA: Johns Hopkins University Press, 2013.