

Canonical Polyadic Decomposition based on joint eigenvalue decomposition

Xavier Luciani^{a,b}, Laurent Albera^{c,d,e,*}

^aAix Marseille Université, CNRS, ENSAM, LSIS, UMR 7296, Marseille, F-13397, France.

^bUniversité de Toulon, CNRS, LSIS, UMR 7296, La Garde, F-83957, France.

^cInserm, UMR 642, Rennes, F-35000, France

^dLTSI, University of Rennes 1, Rennes, F-35000, France

^eInria, Centre Inria Rennes - Bretagne Atlantique, Rennes, F-35000, France

Abstract

A direct algorithm based on Joint EigenValue Decomposition (JEVD) has been proposed to compute the Canonical Polyadic Decomposition (CPD) of multi-way arrays (tensors). The iterative part of our method is thus limited to the JEVD computation. At this occasion we also propose an original JEVD technique. Most of the iterative CPD algorithms such as ALS have been shown by means of practical studies to suffer from convergence problems (local minima, slow convergence or high computational cost per iteration). On the other hand, direct methods seem in practice to confine these disadvantages but impose some restrictive necessary conditions. In this context, our proposed algorithm involves less restrictive necessary conditions than other recent direct approaches and a limited computational complexity. It has been compared to reference (direct and non-direct) algorithms on synthetic arrays and real spectroscopic data. These numerical examples highlight the main advantages of the proposed methods to solve both the JEVD and CPD problems.

Keywords: multi-way arrays, direct canonical polyadic decomposition, PARAFAC, joint eigenvalue decomposition, fluorescence, over-factoring

1. Introduction

In this paper, we mainly propose a direct algorithm for the canonical polyadic decomposition of real or complex-valued tensors (assimilated to multi-way arrays) using the Joint EigenValue Decomposition (JEVD) of a set of non-defective matrices. The present contribution is actually twofold since we jointly propose an algorithm to solve the JEVD problem. Tensor decomposition plays a wider and wider role in numerous application areas such as Psychometric [1], Signal Processing for Biomedical Engineering [2, 3, 4], Sensor array [5, 6, 7], Arithmetic Complexity [8] and Chemometrics [9, 10]. Thanks to its uniqueness properties [11, 12, 13, 14, 15, 16],

*Corresponding author. Address: University of Rennes 1, LTSI, Rennes F-35000, France. Phone: +33 2 23 23 50 58, fax: +33 2 23 23 69 17

Email addresses: luciani@univ-tln.fr (Xavier Luciani), laurent.albera@univ-rennes1.fr, http://perso.univ-rennes1.fr/laurent.albera/ (Laurent Albera)

9 the polyadic decomposition introduced in 1927 by Hitchcock [17] is probably the most popular
10 nowadays. In fact, it is now best known as CANonical DECOMPosition (CANDECOMP) [1],
11 PARAllel FACTor analysis (PARAFAC) [18] or CANDECOMP/PARAFAC (CP). In order to
12 be consistent and honor the original work we will keep the acronym CPD, which stands for
13 Canonical Polyadic Decomposition.

14 More precisely, a polyadic decomposition of an array is a sum of rank-one terms that yields
15 an exact fit [17]. The CPD is then defined as the minimal polyadic decomposition. The rank
16 of an array may be thus defined as the minimal number of rank-1 tensors needed to achieve the
17 CPD.

18 Many algorithms have been proposed in order to compute the CPD of multi-way arrays. One
19 of the most famous algorithms, due to its speed and ease of implementation, resorts to an iter-
20 ative Alternating Least Squares (ALS) procedure [18]. Other iterative algorithms based on first
21 and second order optimization methods such as gradient, Gauss-Newton, Levenberg-Marquardt
22 or conjugate gradient have also been proposed (see [19] [20, 21, 22] for a full comparison).
23 Recently, a set of iterative algorithms based on a reduced functional has been introduced in
24 [23]. These last algorithms bring qualitative information on the solution but the counter part is a
25 longer computational time. Furthermore, an Enhanced Line Search (ELS) procedure has been
26 proposed in [24] in order to speed up the ALS algorithm. ELS extension to other iterative CPD
27 algorithm and efficiency of the ALS-ELS algorithm has been highlighted in [21]. However, in
28 spite of this refinement, the ALS algorithm suffers from a classical drawback. Indeed, nothing
29 ensures its global convergence and it can be stuck in local minima. More generally, iterative
30 approaches show convergence problems when several factors of the CPD are correlated.

31 In the meantime, a few direct approaches have been proposed. One can mention the DTLTD
32 approach [25]. However it is restricted to three-way arrays and provide poor results [26, 20].
33 Thereby this kind of solution is generally used as a way of initializing iterative methods. Other
34 direct approaches have been proposed in the literature as well but not yet compared numerically
35 in studies such as the ones mentioned above. These methods rephrase the CPD as the simul-
36 taneous diagonalization, by equivalence [27, 28, 29] or congruence [15], of a set of matrices.
37 The CPD problem can also be translated into a simultaneous generalized Schur decomposition,
38 with orthogonal unknowns, as shown in [29]. Direct methods compute the CPD by solving an
39 alternative algebra problem of lower dimensions but they do not provide a solution in terms of
40 least squares contrarily to the ALS and derivative-based techniques. The reformulated problem
41 is usually solved by means of a Jacobi-like procedure.

42 We thus propose here a new formulation of the CPD as a JEVD problem leading to a novel
43 direct solution, named DIAG (Direct ALGORITHM for canonical polyadic decomposition), involv-
44 ing less restrictive necessary conditions than the "Closed Form Solution" (CFS) presented in
45 [27, 28]. Recall that the CFS algorithm requires that the rank of the considered CPD array does
46 not exceed two of the dimensions of the array. At this occasion we also propose an original
47 Jacobi-like JEVD algorithm, called JDTM (Joint Diagonalization algorithm based on Target-
48 ing hyperbolic Matrices). Numerical examples highlight the main advantages of the proposed
49 methods to solve the JEVD and CPD problems. Note that the DIAG method can be seen as
50 a generalization of the BIOME approach [30] to the case of unsymmetric arrays. JDTM and
51 DIAG have been presented briefly in two separate conference papers [31, 32], respectively. In
52 [32] DIAG was associated to another JEVD algorithm and was called SALT (SemiALgebraic
53 Tensor decomposition). The present paper details theoretical aspects of both algorithms in sec-
54 tions 2 and 3, respectively including their extension to the complex case which is not trivial and

55 their computational complexity. In addition subsection 3.5 is dedicated to the comparison of nec-
 56 essary conditions of different CPD algorithms, namely ALS, CFS and DIAG . Numerical results
 57 are also emphasized in section 4 which illustrate the main features of the DIAG approach, no-
 58 tably the problem of over-factoring is addressed. Finally a concrete application to fluorescence
 59 spectroscopy is proposed in section 5.

60 2. Joint eigenvalue decomposition of non-defective matrices

61 We use the following consistent notations in the whole paper: vectors, matrices and tensors
 62 are denoted by lower case boldface (\mathbf{a}), upper case boldface (\mathbf{A}) and upper case boldface calli-
 63 graphic (\mathcal{A}) letters respectively. The i -th entry of vector \mathbf{a} is denoted by a_i while A_{ij} is the (i, j) -
 64 th component of matrix \mathbf{A} . Entry (i_1, \dots, i_Q) of any Q -order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \dots \times I_Q}$ or $\mathbb{C}^{I_1 \times \dots \times I_Q}$
 65 ($Q > 2$) is denoted by $\mathcal{T}_{i_1, \dots, i_Q}$. Outer product, Kronecker product and Khatri-Rao product are
 66 denoted by \circ , \otimes and \odot , respectively. Moore-Penrose matrix inverse, euclidean and frobenius
 67 norm are denoted by \dagger , $\|\mathbf{E}^{(k)}\|_F$ and $\|\cdot\|_F$, respectively. We define $[x; y]_{\mathbb{N}} = [x; y] \cap \mathbb{N}$. $\lfloor \cdot \rfloor$ denotes
 68 the floor function. Complex modulus and conjugate of any complex z are denoted by $|z|$ and \bar{z}
 69 respectively. The imaginary unit is denoted by i .

70 Givens and hyperbolic rotation matrices are denoted by \mathbf{G} and \mathbf{H} , respectively. For instance in
 71 the real case, $\mathbf{G}(\theta_{ij})$ and $\mathbf{H}(\phi_{ij})$ are equal to the identity matrix, at the exception of the elements:

$$\begin{aligned} G(\theta_{ij})_{ii} &= G(\theta_{ij})_{jj} = \cos(\theta_{ij}) & G(\theta_{ij})_{ij} &= -G(\theta_{ij})_{ji} = \sin(\theta_{ij}) \\ H(\phi_{ij})_{ii} &= H(\phi_{ij})_{jj} = \cosh(\phi_{ij}) & H(\phi_{ij})_{ij} &= H(\phi_{ij})_{ji} = \sinh(\phi_{ij}) \end{aligned}$$

72 The JEVD problem consists in finding an eigenvector matrix \mathbf{A} from a set of non-defective
 73 matrices $\mathbf{M}^{(k)}$ satisfying:

$$\forall k \in [1; K]_{\mathbb{N}}, \mathbf{M}^{(k)} = \mathbf{A} \mathbf{D}^{(k)} \mathbf{A}^{-1}, \quad (1)$$

74 where the K diagonal matrices $\mathbf{D}^{(k)}$ are unknown. One could solve these EVDs separately, and
 75 retain the solution that leads to the best estimate regarding the considered application. However,
 76 as explained in [29], it is safer from a numerical point of view to decompose the K matrices
 77 $\mathbf{M}^{(k)}$ simultaneously, in some optimal sense, especially when the perturbation of these matrices
 78 may have caused eigenvalues to cross each other. Indeed, in practice only noisy observations
 79 of the K matrices $\mathbf{M}^{(k)}$ are clustered and it is well known that, when eigenvalues are close, the
 80 eigenvectors in a single EVD may be strongly affected by small perturbations [33]. The reason is
 81 that for coinciding eigenvalues only the corresponding eigenspace is defined; different directions
 82 in this subspace will emerge as eigenvectors for different infinitesimal perturbations. When this
 83 happens for one or more of the matrices in the JEVD problem, the other matrices may still allow
 84 to identify the actual eigenvectors. This follows theorem proved in [29]:

85 **Theorem 1.** *The JEVD is unique up to a permutation and a scaling of the columns of \mathbf{A} if and*
 86 *only if all the columns of the $K \times N$ matrix \mathbf{E} , whose (k, n) -th component $E_{k,n}$ is equal to $D_{n,n}^{(k)}$*
 87 *are not proportional.*

88 Note that in order to ensure uniqueness of the JEVD up to permutation and scale indeterminacies,
 89 we will assume in the sequel that the K involved diagonal matrices $\mathbf{D}^{(k)}$ fulfil the condition given
 90 in Theorem 1.

91 Few papers have proposed numerical solutions to the JEVD problem. All of them adapted
 92 Jacobi's principle to the search for a non-singular and non-necessarily orthogonal eigenmatrix \mathbf{A}

93 by using a suitable factorization, which is not reduced to the product of Givens matrices. This
 94 domination of Jacobi-like methods is due to their good convergence properties [34].

95 Two main kinds of Jacobi-like algorithms have been developed in this context, based on dif-
 96 ferent matrix factorizations. Originally, several authors had recourse to the QR factorization of \mathbf{A}
 97 in order to compute the different sets of eigenvalues [35, 36]. Arguing that these QR-algorithms
 98 suffer from convergence problems, Fu and Gao proposed an effective sh-rt algorithm [37] based
 99 on the polar decomposition. Indeed the polar decomposition has been used favourably for eigen-
 100 value decomposition purpose since a long time [38, 39, 34] and also for joint diagonalization
 101 by congruence [40]. Then the JUST algorithm was introduced in [41] as a variation of the sh-rt
 102 approach for which the iterative computation of the hyperbolic matrix is made by minimizing an
 103 alternative criterion. We propose here a third criterion and an appropriate optimization method,
 104 giving birth to the JD TM algorithm. Another JEVD approach based on LU factorization and
 105 called JET was introduced in [32] for real-valued matrices.

106 The real case is addressed in the three following subsections. The extension to the complex case
 107 is described in subsection 2.4. JD TM algorithm has been compared to JUST and sh-rt algorithms
 108 in various situations involving real matrices. Significant numerical results are given in section
 109 4.1.

111 2.1. A Jacobi-like process

112 In this subsection, all matrices are square matrices of order N . Polar matrix decomposition
 113 states that any non-singular real matrix can be factorized into the product of an orthogonal matrix
 114 \mathbf{Q} and a symmetric positive semidefinite matrix \mathbf{S} . It is well known that \mathbf{Q} can be decomposed
 115 into a product of Givens rotation matrices $\mathbf{G}(\theta_{ij})$ and a unitary diagonal matrix. In the same way,
 116 it has been shown that \mathbf{S} can be decomposed into a product of hyperbolic rotation matrices $\mathbf{H}(\phi_{ij})$
 117 and diagonal matrices [40]. Thereby, due to the indeterminacies of the JEVD problem mentioned
 118 in theorem 1 and taking into account that diagonal, hyperbolic and Givens matrices commute,
 119 the matrix \mathbf{A} solving the JEVD problem given by (1) can be chosen as a product of Givens and
 120 hyperbolic rotation matrices:

$$\mathbf{A} = \prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{G}(\theta_{ij}) \mathbf{H}(\phi_{ij}). \quad (2)$$

121 Inserting (2) into (1) and using the fact that $\mathbf{H}(\phi_{ij})^{-1} = \mathbf{H}(-\phi_{ij})$ we get:

$$\forall k \in [1; K]_{\mathbb{N}}, \mathbf{D}^{(k)} = \left(\prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{G}(\theta_{ij})^{\top} \mathbf{H}(-\phi_{ij}) \right) \mathbf{M}^{(k)} \left(\prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{G}(\theta_{ij}) \mathbf{H}(\phi_{ij}) \right), \quad (3)$$

122 but we prefer the simpler formulation:

$$\forall k \in [1; K]_{\mathbb{N}}, \mathbf{D}^{(k)} = \left(\prod_{m=1}^M \mathbf{H}(-\phi_m) \mathbf{G}(\theta_m)^{\top} \right) \mathbf{M}^{(k)} \left(\prod_{m=1}^M \mathbf{G}(\theta_m) \mathbf{H}(\phi_m) \right), \quad (4)$$

123 where each integer m of $[1; M]_{\mathbb{N}}$ stands for a couple (i, j) with $1 \leq i < j \leq N$. It is worth men-
 124 tioning that any Givens or hyperbolic matrix is defined by only one parameter (angle). Therefore,
 125 ideally we have to find a set of $M = N(N-1)/2$ couples of parameters $\{(\theta_{ij}, \phi_{ij})\}_{1 \leq i < j \leq N}$ in order
 126 to get (1). Instead of simultaneously identifying these M couples of parameters, a Jacobi-like

127 procedure will repeat sequences of $2M$ successive optimizations until convergence. Each opti-
 128 mization is performed with respect to only one parameter. A sequence of $2M$ optimizations is
 129 generally called a sweep. As a result, $N_s M$ couples of Givens and hyperbolic matrices are used
 130 in practice to identify \mathbf{A} , where N_s is the number of sweeps. We thus look for a matrix \mathbf{A} of the
 131 form $\mathbf{A} = \prod_{n_s=1}^{N_s} \prod_{m=1}^M \mathbf{G}(\theta_m^{n_s}) \mathbf{H}(\phi_m^{n_s})$. The idea is to iteratively diagonalize the $\mathbf{M}^{(k)}$ matrices by
 132 sequentially optimizing with respect to $\theta_m^{n_s}$ and $\phi_m^{n_s}$ for each value of m and n_s . Hence the first
 133 sweep ($n_s = 1$) consists on the following transformations:

$$\forall k \in [1; K]_{\mathbb{N}}, \mathbf{N}^{(k,1,1)} = \mathbf{G}(\theta_1^1)^\top \mathbf{M}^{(k)} \mathbf{G}(\theta_1^1), \quad (5)$$

$$\forall (k, m) \in [1; K]_{\mathbb{N}} \times [1; M]_{\mathbb{N}}, \mathbf{M}^{(k,m,1)} = \mathbf{H}(-\phi_m^1) \mathbf{N}^{(k,m,1)} \mathbf{H}(\phi_m^1). \quad (6)$$

$$\forall (k, m) \in [1; K]_{\mathbb{N}} \times [2; M]_{\mathbb{N}}, \mathbf{N}^{(k,m,1)} = \mathbf{G}(\theta_m^1)^\top \mathbf{M}^{(k,m-1,1)} \mathbf{G}(\theta_m^1) \quad (7)$$

134 Then the following sweeps ($1 < n_s \leq N_s$) follow the same scheme:

$$\forall (k, n_s) \in [1; K]_{\mathbb{N}} \times [2; N_s]_{\mathbb{N}}, \mathbf{N}^{(k,1,n_s)} = \mathbf{G}(\theta_1^{n_s})^\top \mathbf{M}^{(k,M,n_s-1)} \mathbf{G}(\theta_1^{n_s}), \quad (8)$$

$$\forall (k, m, n_s) \in [1; K]_{\mathbb{N}} \times [1; M]_{\mathbb{N}} \times [2; N_s]_{\mathbb{N}}, \mathbf{M}^{(k,m,n_s)} = \mathbf{H}(-\phi_m^{n_s}) \mathbf{N}^{(k,m,n_s)} \mathbf{H}(\phi_m^{n_s}). \quad (9)$$

$$\forall (k, m, n_s) \in [1; K]_{\mathbb{N}} \times [2; M]_{\mathbb{N}} \times [2; N_s]_{\mathbb{N}}, \mathbf{N}^{(k,m,n_s)} = \mathbf{G}(\theta_m^{n_s})^\top \mathbf{M}^{(k,m-1,n_s)} \mathbf{G}(\theta_m^{n_s}), \quad (10)$$

135 Thereby, the optimal corresponding Givens and hyperbolic matrices are sequentially com-
 136 puted in order to get K diagonal matrices $\mathbf{M}^{(k,M,N_s)}$ at the end of the process.

137 2.2. Optimization of matrix angles

138 A natural criterion to compute the optimal (m, n_s) -th Givens angle $\theta_m^{n_s}$ is thus to minimize the
 139 sum of the euclidean norms of the off-diagonal terms of the K matrices $\mathbf{N}^{(k,m,n_s)}$:

$$\zeta_G(\theta_m^{n_s}) = \sum_{k=1}^K \sum_{\substack{p=1, q=1 \\ p \neq q}}^{N_s} \left(N_{pq}^{(k,m,n_s)} \right)^2. \quad (11)$$

140 This criterion is the generalization of the original Jacobi criterion to the joint diagonalization
 141 context. Since Givens matrices are orthogonal, the same definition of $\mathbf{N}^{(k,m,n_s)}$ holds in both the
 142 joint diagonalization by congruence and JEVD cases and thus the same optimization algorithms
 143 can be used. For instance, our proposed algorithm resorts to the same approach as the JAD
 144 algorithm described in [42] whereas the sh-rt and JUST algorithms use their own minimization
 145 scheme.

146 Once the optimal Givens matrix $\mathbf{G}(\theta_m^{n_s})$ is computed, different criteria can be used for the
 147 optimal computation of $\mathbf{H}(\phi_m^{n_s})$. This is the main difference between the three JEVD algorithms.
 148 The sh-rt method aims at minimizing the Frobenius norm of $\mathbf{M}^{(h,m,n_s)}$ where h is found such
 149 that $\left| M_{ii}^{(h,m,n_s)} - M_{jj}^{(h,m,n_s)} \right| = \max_{1 \leq k \leq K} \left| M_{ii}^{(k,m,n_s)} - M_{jj}^{(k,m,n_s)} \right|$, whereas the JUST algorithm resorts to
 150 criterion (11) by replacing $\mathbf{N}^{(k,m,n_s)}$ by $\mathbf{M}^{(k,m,n_s)}$. Instead of minimizing all the (off-diagonal)
 151 entries, we propose to target two particular off-diagonal entries of $\mathbf{M}^{(k,m,n_s)}$: if m corresponds to
 152 the $(i, j)_{i < j}$ couple, we simply aim at computing the optimal $M_{ij}^{(k,m,n_s)}$ and $M_{ji}^{(k,m,n_s)}$ components by
 153 using a "targeting" hyperbolic matrix. It is noteworthy that the transformation (9) affects the i -th
 154 and j -th rows and the i -th and j -th columns of $\mathbf{M}^{(k,m,n_s)}$ but only the (i, j) and the (j, i) components
 155 are twice affected by the hyperbolic matrix and its inverse. Hence our choice to focus on the latter.

156 Therefore, our Joint Diagonalization algorithm based on Targeting hyperbolic Matrices (JDTM)
 157 resorts to the following alternative criterion ζ_H^{JDTM} for the computation of the hyperbolic matrix:

$$\zeta_H^{JDTM}(\phi_m^{n_s}) = \sum_{k=1}^K \left(M_{ij}^{(k,m,n_s)} \right)^2 + \left(M_{ji}^{(k,m,n_s)} \right)^2, \quad (12)$$

158 Targeting some components was originally proposed by Souloumiac in a different context [40].
 159 In the case of Givens matrices we showed that the optimizations of criteria (11) and (12) were
 160 mathematically equivalent.

161 Now, let us look at the components of $\mathbf{M}^{(k,m,n_s)}$. As previously mentioned, we only consider the
 162 (i, j) -th and (j, i) -th components which are given by:

$$M_{ij}^{(k,m,n_s)} = \left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) \frac{\sinh(2\phi_m^{n_s})}{2} + N_{ij}^{(k,m,n_s)} \cosh(\phi_m^{n_s})^2 - N_{ji}^{(k,m,n_s)} \sinh(\phi_m^{n_s})^2, \quad (13)$$

$$M_{ji}^{(k,m,n_s)} = \left(N_{jj}^{(k,m,n_s)} - N_{ii}^{(k,m,n_s)} \right) \frac{\sinh(2\phi_m^{n_s})}{2} - N_{ij}^{(k,m,n_s)} \sinh(\phi_m^{n_s})^2 + N_{ji}^{(k,m,n_s)} \cosh(\phi_m^{n_s})^2. \quad (14)$$

164 Furthermore we can write that:

$$\left(M_{ij}^{(k,m,n_s)} \right)^2 + \left(M_{ji}^{(k,m,n_s)} \right)^2 = \frac{\left(M_{ij}^{(k,m,n_s)} + M_{ji}^{(k,m,n_s)} \right)^2}{2} + \frac{\left(M_{ij}^{(k,m,n_s)} - M_{ji}^{(k,m,n_s)} \right)^2}{2}. \quad (15)$$

165 The first term of the right-hand side does not depend on $\phi_m^{n_s}$. Indeed, we derive from (13) and
 166 (14) the following equality:

$$\frac{\left(M_{ij}^{(k,m,n_s)} + M_{ji}^{(k,m,n_s)} \right)^2}{2} = \frac{\left(N_{ij}^{(k,m,n_s)} + N_{ji}^{(k,m,n_s)} \right)^2}{2}. \quad (16)$$

167 Thereby minimizing ζ_H^{JDTM} is equivalent to minimize the λ function defined by:

$$\lambda(\phi_m^{n_s}) = \sum_{k=1}^K \left(M_{ij}^{(k,m,n_s)} - M_{ji}^{(k,m,n_s)} \right)^2. \quad (17)$$

168 We denote by $\mathbf{y}^{(m,n_s)}$ the column vector of \mathbb{R}^K defined by $y_k^{(m,n_s)} = M_{ij}^{(k,m,n_s)} - M_{ji}^{(k,m,n_s)}$, so that
 169 $\lambda(\phi_m^{n_s}) = \mathbf{y}^{(m,n_s)\top} \mathbf{y}^{(m,n_s)}$. It is easily shown that the system of linear equations (13) and (14) can be
 170 rewritten such that:

$$\mathbf{y}^{(m,n_s)} = \mathbf{W}^{(m,n_s)} \mathbf{x}(\phi_m^{n_s}), \quad (18)$$

171 with:

$$\mathbf{W}^{(m,n_s)} = \begin{bmatrix} N_{ii}^{(1,m,n_s)} - N_{jj}^{(1,m,n_s)} & N_{ij}^{(1,m,n_s)} - N_{ji}^{(1,m,n_s)} \\ \vdots & \vdots \\ N_{ii}^{(K,m,n_s)} - N_{jj}^{(K,m,n_s)} & N_{ij}^{(K,m,n_s)} - N_{ji}^{(K,m,n_s)} \end{bmatrix}; \quad \mathbf{x}(\phi_m^{n_s}) = \begin{bmatrix} \sinh(2\phi_m^{n_s}) \\ \cosh(2\phi_m^{n_s}) \end{bmatrix}.$$

172 Now defining the diagonal 2×2 matrix \mathbf{J} such that $J_{11} = -J_{22} = -1$ and observing that
 173 $\mathbf{x}(\phi_m^{n_s})^\top \mathbf{J} \mathbf{x}(\phi_m^{n_s}) = 1$, we have thus to minimize the quantity $\mathbf{x}(\phi_m^{n_s})^\top \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)} \mathbf{x}(\phi_m^{n_s})$ under

174 the constraint that $\mathbf{x}(\phi_m^{n_s})^\top \mathbf{J} \mathbf{x}(\phi_m^{n_s}) = 1$. This can be done using the Lagrange multipliers strategy.
 175 Thereby, we have to minimize the L function given by:

$$L(\mathbf{x}(\phi_m^{n_s}), \mu(\phi_m^{n_s})) = \mathbf{x}(\phi_m^{n_s})^\top \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)} \mathbf{x}(\phi_m^{n_s}) - \mu(\phi_m^{n_s}) \mathbf{x}(\phi_m^{n_s})^\top \mathbf{J} \mathbf{x}(\phi_m^{n_s}). \quad (19)$$

176 Differentiation with respect to $\mathbf{x}(\phi_m^{n_s})$ leads to:

$$\mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)} \mathbf{x}(\phi_m^{n_s}) = \mu(\phi_m^{n_s}) \mathbf{J} \mathbf{x}(\phi_m^{n_s}). \quad (20)$$

177 Since $\mathbf{J}^{-1} = \mathbf{J}$ we have:

$$\mathbf{J} \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)} \mathbf{x}(\phi_m^{n_s}) = \mu(\phi_m^{n_s}) \mathbf{x}(\phi_m^{n_s}). \quad (21)$$

178 Thus, $\mu(\phi_m^{n_s})$ and $\mathbf{x}(\phi_m^{n_s})$ are associated eigenvalue and eigenvector of matrix $\mathbf{J} \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)}$.
 179 More particularly, we have the following lemma:

180 **Lemma 1.** *If the columns of $\mathbf{W}^{(m,n_s)}$ are different then $\mathbf{J} \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)}$ has two nonzero eigen-*
 181 *values of opposite sign and $\mathbf{x}(\phi_m^{n_s})$ is the eigenvector associated to the positive eigenvalue.*

182 **Proof 1.** *Let \mathbf{w}_1 and \mathbf{w}_2 be the column vectors of matrix $\mathbf{W}^{(m,n_s)}$. Both belong to \mathbb{R}^K , equipped*
 183 *with the Euclidean norm and we define $a = \mathbf{w}_1^\top \mathbf{w}_1$, $b = \mathbf{w}_1^\top \mathbf{w}_2$ and $c = \mathbf{w}_2^\top \mathbf{w}_2$. Hence a , b and c*
 184 *denote the squared euclidean norm of \mathbf{w}_1 , the scalar product between \mathbf{w}_1 and \mathbf{w}_2 and the squared*
 185 *Euclidean norm of \mathbf{w}_2 respectively. Hence,*

$$\mathbf{J} \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)} = \begin{bmatrix} -a & -b \\ b & c \end{bmatrix}$$

186 *The characteristic polynomial is then:*

$$P(\alpha) = \alpha^2 + (a - c)\alpha + (b^2 - ca) \quad (22)$$

187 *and the discriminant is:*

$$\begin{aligned} \Delta &= (a - c)^2 - 4b^2 + 4ca \\ &= (a + c - 2b)(a + c + 2b) \\ &= \|\mathbf{w}_1 - \mathbf{w}_2\|^2 \|\mathbf{w}_1 + \mathbf{w}_2\|^2 \end{aligned}$$

188 *Thereby, since $\mathbf{w}_1 \neq \mathbf{w}_2$, $\Delta > 0$ and $\mathbf{J} \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)}$ is diagonalizable and admits two distinct*
 189 *eigenvalues α_1 and α_2 . Then we have:*

$$\begin{aligned} \alpha_1 \alpha_2 &= \frac{(a - c)^2 - \Delta}{4a^2} \\ &= \frac{b^2 - ac}{a^2} \end{aligned}$$

190 *The Cauchy-Schwartz inequality gives $b^2 < ac$ hence $\alpha_1 \alpha_2 < 0$.*

191 *We now demonstrate the second part of the lemma. Multiplying (21) by $\mathbf{x}(\phi_m^{n_s})^\top \mathbf{J}$ yields:*

$$\begin{aligned} \mathbf{x}(\phi_m^{n_s})^\top \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)} \mathbf{x}(\phi_m^{n_s}) &= \mu(\phi_m^{n_s}) \mathbf{x}(\phi_m^{n_s})^\top \mathbf{J} \mathbf{x}(\phi_m^{n_s}), \\ &= \mu(\phi_m^{n_s}). \end{aligned} \quad (23)$$

192 *The quadratic form $\mathbf{x}(\phi_m^{n_s})^\top \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)} \mathbf{x}(\phi_m^{n_s})$ is positive thus $\mu(\phi_m^{n_s})$ is positive too.*

193 Hence the previous lemma allows us to easily compute $\mathbf{x}(\phi_m^{n_s})$ from $\mathbf{W}^{(m,n_s)}$ and $\phi_m^{n_s}$ is deduced
 194 from the definition of $\mathbf{x}(\phi_m^{n_s})$:

$$\phi_m^{n_s} = \frac{1}{2} \operatorname{atanh} \left(\frac{x(\phi_m^{n_s})_1}{x(\phi_m^{n_s})_2} \right). \quad (24)$$

Algorithm 1 summarizes the proposed method.

Algorithm 1: Summary of the JD TM algorithm

- 1: Define a threshold ε and a maximal number of sweep N_s^{max}
- 2: Initialize \mathbf{A} with the identity matrix;
- 3: $n_s = 1$;
- 4: **while** $\sum_k \sum_{p \neq q} (M_{p,q}^{(k)})^2 > \varepsilon$ and $n_s \leq N_s^{max}$ **do**
- 5: $m = 1$;
- 6: **for** $i = 1$ to $N - 1$ **do**
- 7: **for** $j = i + 1$ to N **do**
- 8: Compute the optimal angle $\theta_m^{n_s}$ corresponding to the couple (i, j) and build $\mathbf{G}(\theta_m^{n_s})$;
- 9: Replace the K matrices $\mathbf{M}^{(k)}$ by $\mathbf{G}(\theta_m^{n_s})^\top \mathbf{M}^{(k)} \mathbf{G}(\theta_m^{n_s})$;
- 10: Compute the optimal angle $\phi_m^{n_s}$ corresponding to the couple (i, j) and build $\mathbf{H}(\phi_m^{n_s})$;
- 11: Replace the K matrices $\mathbf{M}^{(k)}$ by $\mathbf{H}(-\phi_m^{n_s}) \mathbf{M}^{(k)} \mathbf{H}(\phi_m^{n_s})$;
- 12: Replace \mathbf{A} by $\mathbf{A} \mathbf{G}(\theta_m^{n_s}) \mathbf{H}(\phi_m^{n_s})$;
- 13: $m = m + 1$;
- 14: **end for**
- 15: **end for**
- 16: $n_s = n_s + 1$;
- 17: **end while**
- 18: $N_s = n_s$;

195

196 2.3. Computational complexity

197 The computational complexity of an algorithm is given by the number Γ of floating point
 198 operations (flop), given in practice by the number of required multiplications. At each sweep,
 199 there are $N(N - 1)/2$ Givens and hyperbolic matrices to compute and as many updates of ma-
 200 trices $\mathbf{A}, \mathbf{M}^{(1)}, \dots, \mathbf{M}^{(K)}$. Computation of each hyperbolic matrix is dominated by the product
 201 $\mathbf{J} \mathbf{W}^{(m,n_s)\top} \mathbf{W}^{(m,n_s)}$ which requires $3K$ multiplications. Givens matrices are computed in a similar
 202 way [42] and thus also need $3K$ multiplications. For each update (line 12 of algorithm 1), matrix
 203 \mathbf{A} is multiplied by a Givens and a hyperbolic matrix. Both products can be done using a total of
 204 $8N$ multiplications. Finally the update of each matrix $\mathbf{M}^{(k)}$ (lines 9 and 11 of algorithm 1) is twice
 205 more costly and involves $16N$ multiplications. Therefore the total computational complexity is:

$$\Gamma_{JD TM} = N_s N(N - 1)(3K + 4N + 8KN) \quad (25)$$

206 2.4. Extension to the complex case

207 Let's now consider that matrices \mathbf{A} and $\mathbf{M}^{(1)}, \dots, \mathbf{M}^{(K)}$ belong to the complex field. In this
 208 case, the JD TM algorithm has to be significantly modified. Indeed, each of the Givens and
 209 hyperbolic rotation matrices involved in the polar decomposition of a complex matrix is now
 210 defined by two parameters. Similarly to the real case, we only focus on the determination of

211 hyperbolic matrices \mathbf{H} which makes the specificity of the proposed algorithm. Indeed, \mathbf{G} can
 212 still be estimated by the classic procedure [42].

213 We resort to the following classical parametrization of complex hyperbolic matrices, for each
 214 couple $m = (i, j)_{i < j}$ we have:

$$H(\phi_m, \alpha_m)_{ii} = H(\phi_m, \alpha_m)_{jj} = \cosh(\phi_m); \quad H(\phi_m, \alpha_m)_{ij} = \overline{H(\phi_m, \alpha_m)_{ji}} = \sinh(\phi_m) e^{i\alpha_m}$$

215 Thereby we have to estimate for each matrix the couple (ϕ_{ij}, α_{ij}) that minimizes the new
 216 JD $\overline{\text{TM}}$ cost function:

$$\zeta_{HC}^{JD\overline{\text{TM}}}(\phi_m^{n_s}, \alpha_m^{n_s}) = \sum_{k=1}^K |M_{ij}^{(k,m,n_s)}|^2 + |M_{ji}^{(k,m,n_s)}|^2. \quad (26)$$

217 Using the previous parametrization, we obtain:

$$M_{ij}^{(k,m,n_s)} = \left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) \frac{\sinh(2\phi_m^{n_s})}{2} e^{-i\alpha_m^{n_s}} + N_{ij}^{(k,m,n_s)} \cosh(\phi_m^{n_s})^2 - N_{ji}^{(k,m,n_s)} \sinh(\phi_m^{n_s})^2 e^{-2i\alpha_m^{n_s}}, \quad (27)$$

218

$$M_{ji}^{(k,m,n_s)} = \left(N_{jj}^{(k,m,n_s)} - N_{ii}^{(k,m,n_s)} \right) \frac{\sinh(2\phi_m^{n_s})}{2} e^{i\alpha_m^{n_s}} - N_{ij}^{(k,m,n_s)} \sinh(\phi_m^{n_s})^2 e^{2i\alpha_m^{n_s}} + N_{ji}^{(k,m,n_s)} \cosh(\phi_m^{n_s})^2. \quad (28)$$

219 It can be easily shown that minimizing $\zeta_{HC}^{JD\overline{\text{TM}}}$ is equivalent to minimizing $\tilde{\zeta}_{HC}^{JD\overline{\text{TM}}}$:

$$\tilde{\zeta}_{HC}^{JD\overline{\text{TM}}}(\phi_m^{n_s}, \alpha_m^{n_s}) = \sum_{k=1}^K |\tilde{M}_{ij}^{(k,m,n_s)} + M_{ji}^{(k,m,n_s)}|^2 + |\tilde{M}_{ij}^{(k,m,n_s)} - M_{ji}^{(k,m,n_s)}|^2, \quad (29)$$

220 where:

$$\tilde{M}_{ij}^{(k,m,n_s)} = \left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) \frac{\sinh(2\phi_m^{n_s})}{2} e^{i\alpha_m^{n_s}} + N_{ij}^{(k,m,n_s)} \cosh(\phi_m^{n_s})^2 e^{-2i\alpha_m^{n_s}} - N_{ji}^{(k,m,n_s)} \sinh(\phi_m^{n_s})^2. \quad (30)$$

221 After some straightforward computations, (28), (29) and (30) yield:

$$\begin{aligned} \tilde{\zeta}_{HC}^{JD\overline{\text{TM}}}(\phi_m^{n_s}, \alpha_m^{n_s}) = & \sum_{k=1}^K \left(\left| N_{ij}^{(k,m,n_s)} \right|^2 + \left| N_{ji}^{(k,m,n_s)} \right|^2 \right) \cosh(2\phi_m^{n_s})^2 \\ & + \left(\left| N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right|^2 - \left(N_{ij}^{(k,m,n_s)} \overline{N_{ji}^{(k,m,n_s)}} e^{2i\alpha_m^{n_s}} + N_{ij}^{(k,m,n_s)} \overline{N_{ji}^{(k,m,n_s)}} e^{-2i\alpha_m^{n_s}} \right) \right) \sinh(2\phi_m^{n_s})^2 \\ & + \frac{1}{2} \left(\overline{\left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) N_{ij}^{(k,m,n_s)}} - \left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) \overline{N_{ji}^{(k,m,n_s)}} \right) e^{i\alpha_m^{n_s}} \sinh(4\phi_m^{n_s}) \\ & + \frac{1}{2} \left(\left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) \overline{N_{ij}^{(k,m,n_s)}} - N_{ji}^{(k,m,n_s)} \overline{\left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right)} \right) e^{-i\alpha_m^{n_s}} \sinh(4\phi_m^{n_s}) \end{aligned} \quad (31)$$

222

223

which can be rewritten as a function of $4\phi_m^{n_s}$ and $\alpha_m^{n_s}$:

$$\begin{aligned} \tilde{\zeta}_{HC}^{JD\overline{\text{TM}}}(4\phi_m^{n_s}, \alpha_m^{n_s}) = & \frac{1}{2} \sum_{k=1}^K \left(\left| N_{ij}^{(k,m,n_s)} \right|^2 + \left| N_{ji}^{(k,m,n_s)} \right|^2 \right) (\cosh(4\phi_m^{n_s}) + 1) \\ & + \left(\left| N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right|^2 - \left(N_{ij}^{(k,m,n_s)} \overline{N_{ji}^{(k,m,n_s)}} e^{2i\alpha_m^{n_s}} + N_{ij}^{(k,m,n_s)} \overline{N_{ji}^{(k,m,n_s)}} e^{-2i\alpha_m^{n_s}} \right) \right) (\cosh(4\phi_m^{n_s}) - 1) \\ & + \overline{\left(\left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) N_{ij}^{(k,m,n_s)} \right)} - \left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) \overline{N_{ji}^{(k,m,n_s)}} e^{i\alpha_m^{n_s}} \sinh(4\phi_m^{n_s}) \\ & + \left(\left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right) \overline{N_{ij}^{(k,m,n_s)}} - N_{ji}^{(k,m,n_s)} \overline{\left(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)} \right)} \right) e^{-i\alpha_m^{n_s}} \sinh(4\phi_m^{n_s}). \end{aligned} \quad (32)$$

224

Differentiating (32) with respect to $4\phi_m^{n_s}$ and $\alpha_m^{n_s}$ alternatively, then defining $t_m^{n_s} = \tanh(2\phi_m^{n_s})$ and $z_m^{n_s} = e^{i\alpha_m^{n_s}}$, it can be shown after few more trivial computations that the solution couple which minimizes $\tilde{\zeta}_{HC}^{JDTM}$ is also a solution of the following polynomial system:

$$\begin{cases} P_0(z_m^{n_s}) + (2P_1(z_m^{n_s})t_m^{n_s} + P_0(z_m^{n_s})t_m^{n_s})t_m^{n_s} = 0 & (33) \\ (Q_1(z_m^{n_s})t_m^{n_s} - Q_0(z_m^{n_s}))t_m^{n_s} = 0 & (34) \end{cases}$$

225 with:

$$\begin{aligned} P_0(z) &= \sum_{k=1}^K \left(\overline{(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)})} N_{ij}^{(k,m,n_s)} - (N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)}) \overline{N_{ji}^{(k,m,n_s)}} \right) z^3 \\ &\quad + \left((N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)}) \overline{N_{ij}^{(k,m,n_s)}} - N_{ji}^{(k,m,n_s)} \overline{(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)})} \right) z \\ P_1(z) &= \sum_{k=1}^K -\overline{N_{ji}^{(k,m,n_s)}} N_{ij}^{(k,m,n_s)} z^4 + \left(|N_{ij}^{(k,m,n_s)}|^2 + |N_{ji}^{(k,m,n_s)}|^2 + |N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)}|^2 \right) z^2 - \overline{N_{ij}^{(k,m,n_s)}} N_{ji}^{(k,m,n_s)} \\ Q_0(z) &= \sum_{k=1}^K \left(\overline{(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)})} N_{ij}^{(k,m,n_s)} - (N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)}) \overline{N_{ji}^{(k,m,n_s)}} \right) z^3 \\ &\quad - \left((N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)}) \overline{N_{ij}^{(k,m,n_s)}} - N_{ji}^{(k,m,n_s)} \overline{(N_{ii}^{(k,m,n_s)} - N_{jj}^{(k,m,n_s)})} \right) z \\ Q_1(z) &= \sum_{k=1}^K 2 \left(\overline{N_{ji}^{(k,m,n_s)}} N_{ij}^{(k,m,n_s)} z^4 - \overline{N_{ij}^{(k,m,n_s)}} N_{ji}^{(k,m,n_s)} \right) \end{aligned} \quad (35)$$

226 Solution sets are then easily given by:

$$P_0(z_m^{n_s}) = 0 \text{ and } t_m^{n_s} = 0; \quad (36)$$

227 or:

$$P_0(z_m^{n_s})(Q_1(z_m^{n_s}))^2 + 2P_1(z_m^{n_s})Q_0(z_m^{n_s})Q_1(z_m^{n_s}) + P_0(z_m^{n_s})(Q_0(z_m^{n_s}))^2 = 0 \text{ and } t_m^{n_s} = \frac{Q_0(z_m^{n_s})}{Q_1(z_m^{n_s})}. \quad (37)$$

228 **3. Toward a new direct CPD algorithm: the DIAG method**229 *3.1. The Canonical Polyadic Decomposition*

230 CPD states that any Q -order tensor (or Q -way array) \mathcal{T} of size $I_1 \times \dots \times I_Q$ can be exactly
231 decomposed into a sum of Q -order rank-1 tensors. A Q -order rank-1 tensor can be defined as the
232 outer product between Q vectors $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(Q)}$. The rank R of \mathcal{T} is then the minimal number of
233 rank-1 tensors needed to achieve the following decomposition:

$$\mathcal{T} = \sum_{r=1}^R \mathbf{x}_r^{(1)} \circ \dots \circ \mathbf{x}_r^{(Q)}. \quad (38)$$

234 Usually one also defines Q "loading" (or factor) matrices $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(Q)}$ of size $I_1 \times R, \dots,$
 235 $I_Q \times R$, respectively, so that $\mathbf{x}_r^{(q)}$ is the r^{th} column of $\mathbf{X}^{(q)}$ and the CPD is commonly rewritten as:

$$\forall q \in [1; Q]_{\mathbb{N}}, \forall i_q \in [1; I_q]_{\mathbb{N}}, \mathcal{T}_{i_1 \dots i_Q} = \sum_{r=1}^R \mathbf{X}_{i_1 r}^{(1)} \mathbf{X}_{i_2 r}^{(2)} \dots \mathbf{X}_{i_Q r}^{(Q)}. \quad (39)$$

236 Our main problem is thus to find for a given tensor \mathcal{T} of given rank R and order Q , the Q factor
 237 matrices that solves (39).

238 3.2. Unfolding matrix

239 It is well known that the CPD can be rewritten in a matrix form. Indeed, the tensor dimensions
 240 can be merged in order to store all tensor entries in a single "unfolding" matrix. Obviously, there
 241 are many way to merge the tensor dimensions and thus many possible unfolding matrices. As it
 242 will be seen, the choice of the unfolding matrix has an impact on the algorithm limitations and
 243 performance. Therefore, in order to cover all the possibilities, we introduce a P parameter in
 244 order that the P first dimensions are merged into the matrix rows whereas the remaining $Q - P$
 245 dimensions are merged into the matrix columns. The corresponding unfolding matrix is denoted
 246 by $\mathbf{T}(P)$. Note that all the other unfolding matrices can be merely obtained by permuting the
 247 tensor dimensions and changing the P value. $\mathbf{T}(P)$ entries are linked to \mathcal{T} entries by the following
 248 transfer formulas:

$$\forall (m, n) \in [1; \pi_1^P]_{\mathbb{N}} \times [1; \pi_{P+1}^Q]_{\mathbb{N}}, \mathbf{T}(P)_{m,n} = \mathcal{T}_{i_1, \dots, i_Q} \quad (40)$$

249 where, $\pi_a^a = I_a, \pi_a^b = I_a I_{a+1} \dots I_b$ and:

$$\forall m \in [1; \pi_1^P]_{\mathbb{N}}, \quad m = i_1 + \sum_{q=2}^P (i_q - 1) \pi_1^{q-1}, \quad (41)$$

$$\forall n \in [1; \pi_{P+1}^Q]_{\mathbb{N}}, \quad n = i_{P+1} + \sum_{q=P+2}^Q (i_q - 1) \pi_{P+1}^{q-1}. \quad (42)$$

250 Then after some computations the CPD equation (39) can be rewritten as:

$$\mathbf{T}(P) = (\mathbf{X}^{(P)} \circ \dots \circ \mathbf{X}^{(1)}) (\mathbf{X}^{(Q)} \circ \dots \circ \mathbf{X}^{(P+1)})^T. \quad (43)$$

251 It is worth mentioning that a majority of CPD algorithms such as ALS or CFS resorts to the
 252 $P = 1$ case.

253 3.3. The DIAG algorithm

254 The algorithm presented here is available both in the real and complex field. We start from
 255 equation (43) and we define for a given couple of integers a and $b, a < b$, the matrix $\mathbf{Y}_X^{(b,a)}$ by:

$$\mathbf{Y}_X^{(b,a)} = \mathbf{X}^{(b)} \circ \dots \circ \mathbf{X}^{(a)}. \quad (44)$$

256 Now, let USV^H be the singular value decomposition of $\mathbf{T}(P)$ truncated at the order R , assuming
 257 that $R \leq \min(\pi_1^P, \pi_{P+1}^Q)$ (hypothesis \mathcal{H}_1). Thus there exists an invertible square matrix \mathbf{M} of size
 258 $R \times R$ such that:

$$\mathbf{Y}_X^{(P,1)} = \mathbf{U} \mathbf{M}, \quad (45)$$

$$\mathbf{Y}_X^{(Q, P+1)T} = \mathbf{M}^{-1} \mathbf{S} \mathbf{V}^H. \quad (46)$$

259 Recalling that $\mathbf{Y}_X^{(Q,P+1)} = \mathbf{X}^{(Q)} \odot \mathbf{Y}_X^{(Q-1,P+1)}$ and using the definition of the Kathri-Rao product,
 260 $\mathbf{Y}_X^{(Q,P+1)\top}$ can be seen as a row block matrix:

$$\mathbf{Y}_X^{(Q,P+1)\top} = [\boldsymbol{\phi}^{(1)} \mathbf{Y}_X^{(Q-1,P+1)\top}, \dots, \boldsymbol{\phi}^{(I_Q)} \mathbf{Y}_X^{(Q-1,P+1)\top}], \quad (47)$$

261 where $\boldsymbol{\phi}^{(1)}, \dots, \boldsymbol{\phi}^{(I_Q)}$ are the I_Q diagonal matrices built from the I_Q rows of the matrix \mathbf{X}^Q . As a
 262 consequence, equations (46) and (47) yield:

$$\mathbf{S}\mathbf{V}^H = [\boldsymbol{\Gamma}^{(1)\top}, \dots, \boldsymbol{\Gamma}^{(I_Q)\top}], \quad (48)$$

263 where :

$$\forall i \in [1; I_Q]_{\mathbb{N}}, \boldsymbol{\Gamma}^{(i)} = \mathbf{Y}_X^{(Q-1,P+1)} \boldsymbol{\phi}^{(i)} \mathbf{M}^\top. \quad (49)$$

264 All matrices $\boldsymbol{\Gamma}^{(i)}$ and $\mathbf{Y}_X^{(Q-1,P+1)}$ are of size $\pi_{P+1}^{Q-1} \times R$. We assume that P is chosen so that $P < Q-1$
 265 and $R \leq \pi_{P+1}^{Q-1}$ (hypothesis \mathcal{H}_2) and that they all admit a Moore-Penrose matrix inverse. Then we
 266 define:

$$\forall i_1, i_2 \in [1; I_Q]_{\mathbb{N}}, i_2 > i_1 \boldsymbol{\Theta}^{(i_1, i_2)} = \boldsymbol{\Gamma}^{(i_1)} \# \boldsymbol{\Gamma}^{(i_2)}. \quad (50)$$

267 Now replacing $\boldsymbol{\Gamma}^{(i)}$ by its definition yields:

$$\boldsymbol{\Theta}^{(i_1, i_2)} = \mathbf{M}^{-\top} \boldsymbol{\phi}^{(i_1)-1} \mathbf{Y}_X^{(Q-1,P+1)} \# \mathbf{Y}_X^{(Q-1,P+1)} \boldsymbol{\phi}^{(i_2)} \mathbf{M}^\top, \quad (51)$$

$$= \mathbf{M}^{-\top} \boldsymbol{\Lambda}^{(i_1, i_2)} \mathbf{M}^\top, \quad (52)$$

268 where $\boldsymbol{\Lambda}^{(i_1, i_2)} = \boldsymbol{\phi}^{(i_1)-1} \boldsymbol{\phi}^{(i_2)}$. Thus, $\mathbf{M}^{-\top}$ performs the JEVD of the known set of matrices $\boldsymbol{\Theta}^{(i_1, i_2)}$.
 269 Therefore $\mathbf{M}^{-\top}$ can be estimated by the JD TM algorithm. Then one can immediately deduce
 270 $\mathbf{Y}_X^{(P,1)}$ and $\mathbf{Y}_X^{(Q,P+1)}$ from (45) and (46). At this stage there are several ways to estimate the factor
 271 matrices from $\mathbf{Y}_X^{(P,1)}$ and $\mathbf{Y}_X^{(Q,P+1)}$. One simple approach is to estimate each column of the first P
 272 factor matrices from the corresponding column of $\mathbf{Y}_X^{(P,1)}$ and each column of the $Q-P$ remaining
 273 factor matrices from the corresponding column of $\mathbf{Y}_X^{(Q,P+1)}$. Indeed, column r of $\mathbf{Y}_X^{(P,1)}$ can be
 274 reshaped into an order- P , rank-1 tensor $\boldsymbol{\mathcal{Y}}_{X_r}^{(P,1)}$ whose factor vectors are the r -th columns of matri-
 275 ces $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(P)}$. Thereby a simple rank-1 High-Order SVD (HOSVD, [43]) of $\boldsymbol{\mathcal{Y}}_{X_r}^{(P,1)}$ provides
 276 a direct estimation of $\mathbf{x}_r^{(1)}, \dots, \mathbf{x}_r^{(P)}$. In the same way, the column r of $\mathbf{Y}_X^{(Q,P+1)}$ can be reshaped
 277 in a $(Q-P)$ -order, rank-1 tensor $\boldsymbol{\mathcal{Y}}_{X_r}^{(Q,P+1)}$ whose factor vectors are the r -th columns of matrices
 278 $\mathbf{X}^{(P+1)} \dots \mathbf{X}^{(Q)}$. Hence, $\mathbf{x}_r^{(P+1)} \dots \mathbf{x}_r^{(Q)}$ can be estimated from the rank-1 HOSVD of $\boldsymbol{\mathcal{Y}}_{X_r}^{(Q,P+1)}$. Fi-
 279 nally both operations are repeated for all the r values. The DIAG algorithm is summarized by
 280 Algorithm 2.

281 3.4. Computational complexity

282 Γ_{DIAG} is clearly dominated by the three following computations. First, the truncated SVD of
 283 the unfolding matrix of size (π_1^P, π_{P+1}^Q) requires $2\pi_{P+1}^Q (\pi_1^P)^2 + 5R^2(\pi_1^P + \pi_{P+1}^Q) - 2(R^3 + (\pi_1^P)^3)/3$
 284 multiplications, assuming that $\pi_{P+1}^Q > \pi_1^P$. Then, the computation of the $\boldsymbol{\Theta}$ matrices needs ap-
 285 proximately $(RI_Q)^2 \pi_{P+1}^{Q-1}$ additional multiplications. Finally the cost of the JEVD procedure is
 286 approximated by $8N_s(I_Q)^2 R^3$. Additional computations can be neglected and thus we have:

$$\Gamma_{DIAG} \approx 2\pi_{P+1}^Q (\pi_1^P)^2 + 5R^2(\pi_1^P + \pi_{P+1}^Q) - 2(R^3 + (\pi_1^P)^3)/3 + (RI_Q)^2 \pi_{P+1}^{Q-1} + 8N_s(I_Q)^2 R^3. \quad (53)$$

Algorithm 2: Summary of the DIAG algorithm

- 1: Choose a value of P and a permutation of the dimensions of \mathcal{T} as described in section 3.6;
- 2: Matricize the (possibly permuted) tensor \mathcal{T} into matrix $T(P)$ according to (40), (41) and (42);
- 3: Compute the SVD USV^H of $T(P)$, truncated at rank R ;
- 4: Split SV^H into I_Q blocks of size $R \times \pi_{P+1}^{Q-1}$ in order to form the I_Q matrices $\Gamma^{(i)}$ given by (49);
- 5: **for** $i_1 = 1$ to $I_Q - 1$ **do**
- 6: **for** $i_2 = i_1 + 1$ to I_Q **do**
- 7: Compute $\Theta^{(i_1, i_2)} = \Gamma^{(i_1)} \# \Gamma^{(i_2)}$;
- 8: **end for**
- 9: **end for**
- 10: Compute matrix M^{-T} by JEVD of the set of $\Theta^{(i_1, i_2)}$ matrices;
- 11: Deduce matrices $Y_X^{(P,1)} = UM$ and $Y_X^{(Q,P+1)} = M^{-1}SV^H$;
- 12: **for** $r = 1$ to R **do**
- 13: Build $\mathcal{Y}_{X_r}^{(P,1)}$ and $\mathcal{Y}_{X_r}^{(Q,P+1)}$ by reshaping the r -th columns of $Y_X^{(P,1)}$ and $Y_X^{(Q,P+1)}$;
- 14: Deduce $\mathbf{x}_r^{(1)}, \dots, \mathbf{x}_r^{(P)}$ from the rank 1 HOSVD of $\mathcal{Y}_{X_r}^{(P,1)}$;
- 15: Deduce $\mathbf{x}_r^{(P+1)}, \dots, \mathbf{x}_r^{(Q)}$ from the rank 1 HOSVD of $\mathcal{Y}_{X_r}^{(Q,P+1)}$;
- 16: **end for**

287 Γ_{DIAG} should be compared to the numerical complexity of the ALS algorithm which is approxi-
 288 mately given by:

$$\Gamma_{ALS} \approx N_{ALS} \left(3R\pi_1^Q + 7R^2 \sum_{q=1}^Q \prod_{\substack{k=1 \\ k \neq q}}^Q I_k \right), \quad (54)$$

289 However the numerical complexity of the DIAG algorithm is strongly related to the choice of the
 290 unfolding matrix and both complexities depend on a large number of parameters. Furthermore
 291 N_{ALS} can fluctuate wildly. Therefore at this point it would be very hazardous to draw general con-
 292 clusions from the previous formulas even in simple cases. Nevertheless we made some extensive
 293 flop comparisons between both algorithms by varying Q, R, P and the tensor dimensions. Results
 294 are reported in section 4.2.4. It will be shown that in all the considered situations $\Gamma_{DIAG} \leq \Gamma_{ALS}$
 295 and $N_s \ll N_{ALS}$.

296
 297 The numerical complexity of the CFS algorithm is very complicated to establish since this
 298 algorithm computes several estimations of each factor matrix. However we can easily explain
 299 what makes DIAG a cheaper approach. CFS is a three step algorithm. The first step is algebraic
 300 and performs the HOSVD of the tensor. In terms of numerical complexity this operation is
 301 usually close to the SVD of the unfolding matrix performed in the DIAG algorithm. The second
 302 step is the resolution of $Q(Q-1)^2$ JEVDs whereas DIAG requires only one JEVD. Finally, we
 303 have to choose the best estimates of the factor matrices among a large number of combinations
 304 which is also very time consuming.

305 3.5. Necessary conditions to the identifiability of DIAG, ALS and CFS

306 The CPD algorithms are not always applicable due to their intrinsic restricted conditions.
 307 We propose to compare here necessary conditions that ensure identifiability of the ALS, CFS and

308 DIAG methods. Let Q , R and $I(i)$ be the tensor order, the CPD rank and the i -th dimension of
 309 the tensor, respectively. A tensor of order Q and rank R can be canonically decomposed by ALS
 310 only if:

$$(C_{ALS}) : \forall q \in [1; Q]_{\mathbb{N}}, \prod_{\substack{i=1 \\ i \neq q}}^Q I(i) \geq R. \quad (55)$$

311 DIAG conditions are given by hypotheses \mathcal{H}_1 and \mathcal{H}_2 . \mathcal{H}_1 and \mathcal{H}_2 were expounded for a given
 312 order of the tensor dimensions (default order). Actually, By taking into account that the dimen-
 313 sions can be permuted we obtain the following more general condition:

$$(C_{DIAG}) : \exists P \in [2; Q - 1]_{\mathbb{N}}, \exists f_I \text{ a permutation of the } Q \text{ first natural numbers and } \exists q_s > P \text{ such that:} \\ \prod_{i=1}^P I(f_I(i)) \geq R \text{ and } \prod_{\substack{i=P+1 \\ i \neq q_s}}^Q I(f_I(i)) \geq R. \quad (56)$$

314 Finally, the condition C_{CFS} for the closed-form solution is given in [28]:

$$(C_{CFS}) : \exists (q_1, q_2) \in [1; Q]_{\mathbb{N}}^2, q_1 \neq q_2 \text{ such that } I(q_1) \geq R \text{ and } I(q_2) \geq R. \quad (57)$$

315 **Proposition 1.** C_{DIAG} is more restrictive than C_{ALS} but less restrictive than C_{CFS} :
 316 $C_{CFS} \Rightarrow C_{DIAG} \Rightarrow C_{ALS}$

317 A proof is given in appendix. In practice the DIAG condition implies $P \leq Q - 2$ and can be
 318 reformulated quite easily for low order tensors ($3 \leq Q \leq 5$):

319 **Third order tensors, $Q = 3$.** Here we have necessarily $P = 1$ hence C_{DIAG} becomes simply: at
 320 least two of the tensor dimensions are greater or equal to the CPD rank R . Thereby at order
 321 3 (and only at order 3) C_{DIAG} and C_{CFS} are equivalent.

322 **Fourth order tensors, $Q = 4$.** Here we can choose either $P = 1$ or $P = 2$ but the condition
 323 remains the same in both cases and is simply: at least one tensor dimension is greater than
 324 R and at least one product of two of the remaining dimensions is also greater than R .

325 **Fifth order tensors, $Q = 5$.** Here $1 \leq P \leq 3$:

- 326 • if we choose $P = 1$ or $P = 3$ then C_{DIAG} becomes: at least one tensor dimension is
 327 greater than R and at least one product of three of the remaining dimensions is also
 328 greater than R .
- 329 • if we choose $P = 2$ then C_{DIAG} becomes: at least one product between two tensor
 330 dimensions and another product between two of the remaining dimensions are greater
 331 than R .

332 3.6. Choice of the unfolding matrix

333 An obvious criterion is the residual error between \mathcal{T} and the reconstructed tensor built from
 334 the estimated factor matrices. However it would be very time consuming to test several possibil-
 335 ities. As a consequence the choice of the more appropriate unfolding matrix should be related to
 336 hypothesis \mathcal{H}_1 and \mathcal{H}_2 . Indeed, one has to choose a permutation of the tensor dimensions and
 337 a P value that ensure both hypotheses. Otherwise, the DIAG algorithm is not suitable as it is
 338 explained in the previous section. Recall notably that the DIAG algorithm implies $P \leq Q - 2$.

339 Indeed, at order 3 we have necessarily $P = 1$. At order 4 we have two possible values (1 and
340 2) and so on. Therefore if one wants to maximize the value of the highest possible rank then
341 one should maximize $\min(\pi_1^{q-1}, \pi_{p+1}^{q-1})$, hence choose $\mathbf{T}(p)$ as squared as possible. In practice
342 we observed that this recommendation is always a good option even if all tensor dimensions are
343 greater than the rank. Apart from that one should note that the number of matrices to be jointly
344 diagonalized is directly related to the squared dimension of the last mode and thus the numerical
345 complexity of the JEVD step. Therefore in the case of a tensor with one very large dimension
346 we do not recommend to put it at the end (if possible). More generally, we recommend to take
347 into consideration the overall complexity of the DIAG algorithm given by equation (53) and to
348 consider that with the JD TM algorithm the number of sweeps (N_s) exceeds very rarely 10. In
349 section 4.2.4 we give several significant numerical examples of DIAG complexity for various
350 tensor dimensions and unfolding matrices.

351 4. Numerical simulations

352 The proposed algorithms are first validated on synthesized data sets. We first focus the JEVD
353 sub-problem for which we compare JD TM performances to these of other JEVD algorithms.
354 Then we compare the DIAG approach with CFS, an other direct algorithm and ALS-ELS which
355 is a reference iterative method, with respect to several scenarios. The last subsection is dedicated
356 to a particular tensor family for which iterative algorithms consistently fail to find the CPD.

357 4.1. Performance comparison of the JD TM algorithm

358 The performance of the JD TM algorithm is studied and compared to that of the JET, sh-rt
359 and JUST methods by varying the number K of matrices to be jointly diagonalized, the Signal-
360 to-Noise Ratio (SNR) and the matrix dimensions N . The matrix set to be jointly diagonalized is
361 built according to the following model:

$$\forall k \in [1; K]_{\mathbb{N}}, \mathbf{M}^{(k)} = \frac{\tilde{\mathbf{M}}^{(k)}}{\|\tilde{\mathbf{M}}^{(k)}\|_F} + \sigma \frac{\mathbf{E}^{(k)}}{\|\mathbf{E}^{(k)}\|_F} \text{ with } \tilde{\mathbf{M}}^{(k)} = \mathbf{A} \mathbf{D}^{(k)} \mathbf{A}^{-1}. \quad (58)$$

362 Entries of \mathbf{A} , $\mathbf{D}^{(k)}$ and $\mathbf{E}^{(k)}$ are drawn randomly according to a standard normal distribution. The
363 scalar parameter σ allows us to regulate the power of the Gaussian additive noise $\mathbf{E}^{(k)}$. The SNR
364 is then equal to $-20 \log_{10}(\sigma)$. Hence, σ is chosen in order to obtained the desired value of SNR.

365 At the end of each sweep, the squared off-diagonal components of the K matrices $\mathbf{M}^{(k, M, n_s)}$
366 are summed and the obtained value is compared to the value computed at the previous sweep.
367 Algorithms are stopped when the relative deviation between two successive values is smaller
368 than 10^{-3} .

369 After having removed the scaling and permutation indeterminacies we define r_A as the rela-
370 tive root squared error between the true eigenvector matrix and its estimate $\widehat{\mathbf{A}}$:

$$r_A = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^N (A_{i,j} - \widehat{A}_{i,j})^2}{\sum_{i=1}^N \sum_{j=1}^N (A_{i,j})^2}}. \quad (59)$$

371 Note that in most practical applications and notably in blind source separation, one is only inter-
372 ested by the estimation of the eigenvector matrix. Hence r_A appears as a relevant JEVD criterion.

373 Finally the number of sweeps, N_s , required by each algorithm is stored in order to com-
 374 pute the values of the total numerical complexities Γ . Therefore, algorithm results are judged
 375 according to three criteria, namely N_s , Γ and r_A .

376 Each simulation is repeated 100 times with a new draw of the matrices \mathbf{A} , $\mathbf{D}^{(k)}$ and $\mathbf{E}^{(k)}$ at
 377 each time. We present here median values of r_A and mean values of Γ and N_s obtained from each
 378 algorithm.

379 Figures 1, 2 and 3 show simulation results for 3 SNR values (60 dB, 40 dB and 20 dB
 380 respectively). The number of matrices to be jointly diagonalized was fixed to $K = 64$ whereas we
 381 varied the matrix size N from 2 to 32. We first note that the estimation precision of the algorithms
 382 logically increases with the ratio K/N and the SNR. Second, according to r_A criterion JUST
 383 algorithm is consistently outperformed by other algorithms whatever the considered situation.
 384 At 60 dB, figure 1(a) points out that the JD TM and JET algorithm outclass the sh-rt approach
 385 concerning the estimation of eigenvectors matrix. According to this r_A criterion JET performs
 386 slightly better than JD TM for matrix size lower or equal to 16 whereas for the largest size JD TM
 387 clearly provides the best performances. The comparison of the average computational costs
 388 displayed in figure 1(b) shows very closed results between all the algorithms. However JD TM
 389 appears more clearly as the less costly algorithm for largest matrix sizes. This is explained by a
 390 lower and remarkably stable number of sweeps (figure 1(c)). Previous conclusions hold at 40 dB.
 391 However it is interesting to note that concerning the estimation of the eigenvectors matrix JD TM
 392 is now significantly more accurate than JET for $N = 16$ and $N = 32$. Finally, the 20 dB case
 393 highlights the efficiency of the JD TM algorithm which clearly improves JET and sh-rt results, for
 394 matrix sizes larger than 8. However JET is now the faster algorithm. In conclusion JD TM appears
 395 as a very versatile algorithm which provide very accurate results in all the considered situation
 396 (in comparison to its competitors) for a lower number of sweeps. This number is remarkably
 397 stable, being comprised between 3 and 10 in all the considered scenarios. Moreover JD TM
 398 consistently provides the best estimate of the eigenvector matrix for the largest matrix size and
 399 this gap increases with the SNR. To sum up, JD TM offers quite similar performances than its
 400 best competitors (sh-rt or JET) in the easiest cases (regarding SNR and K/N ratio) whereas it
 401 clearly becomes the better choice as the difficulty increases.

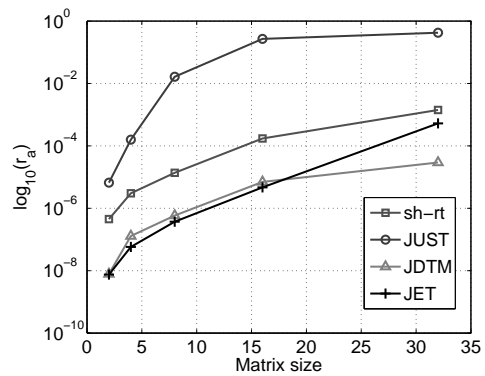
402 As part of this study, we also evaluate JD TM ability to deal with an ill-conditioned eigen-
 403 vector matrix. For this purpose, we now compute the eigenvector matrix \mathbf{A} with pairwise corre-
 404 lated columns as follows: odd columns, \mathbf{a}_{2r-1} , are still randomly drawn as previously but even
 405 columns, \mathbf{a}_{2r} , are built in the following way :

$$\forall r \in [1; N/2]_{\mathbb{N}}, \quad \mathbf{a}_{2r} = \nu \mathbf{a}_{2r-1} + (1 - \nu) \mathbf{n}_r, \quad (60)$$

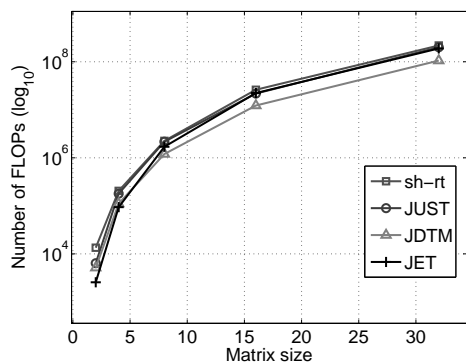
406 where \mathbf{n}_r is a vector of \mathbb{R}^N whose components are randomly drawn according to a standard
 407 normal distribution. Thereby ν defines a collinearity factor which will vary from 0.1 to 0.9 so
 408 that matrices \mathbf{A} can be very ill-conditioned. Figure 4 shows simulation results for a set of 10
 409 matrices of size 10 ($K = N = 10$) at 80 dB. It can be seen that sh-rt, JD TM and JET perform well
 410 for $\nu < 0.9$. JD TM and JET provide the best results in terms of estimation precision but JD TM
 411 requires a minimal number of sweeps and computational cost.

412 4.2. Performance comparison of the DIAG algorithm

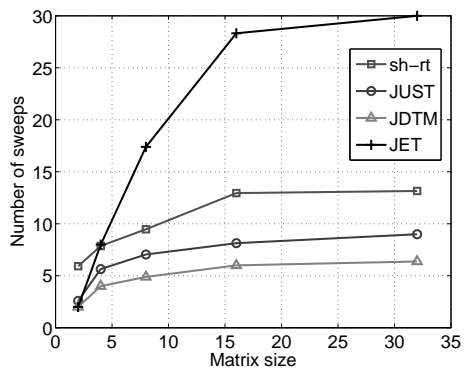
413 We now study performances of the DIAG algorithm for the decomposition of noisy tensors.
 414 Indeed, in most practical applications involving tensor analysis, a noisy tensor of rank R is mod-



(a) r_A criterion (median)

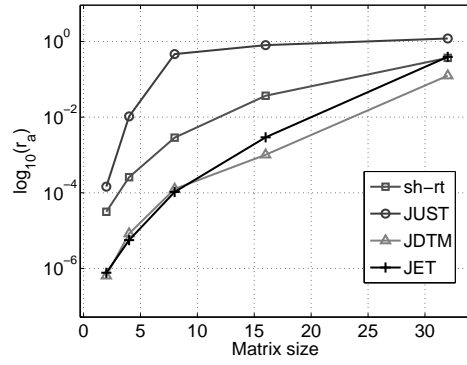


(b) Γ criterion (mean)

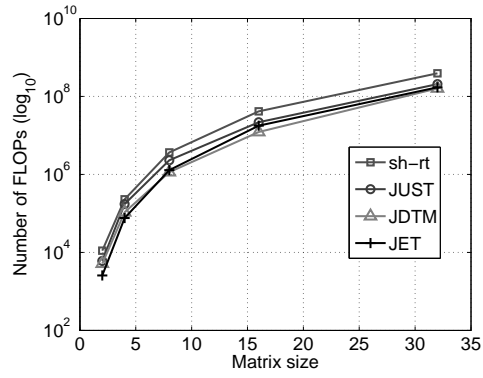


(c) Number of Sweeps (mean)

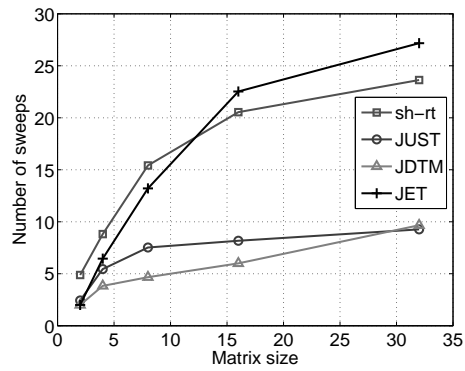
Figure 1: Evolution of the three comparison criteria as a function of the matrix size for a set of 64 matrices with an SNR value of 60 dB.



(a) r_A criterion (median)

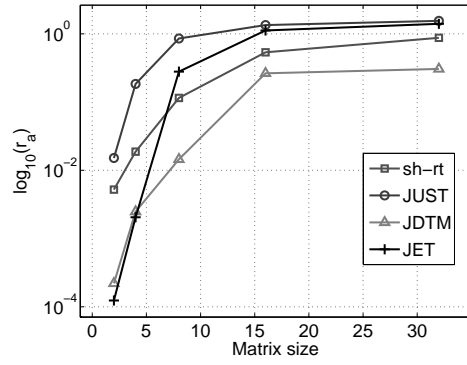


(b) Γ criterion (mean)

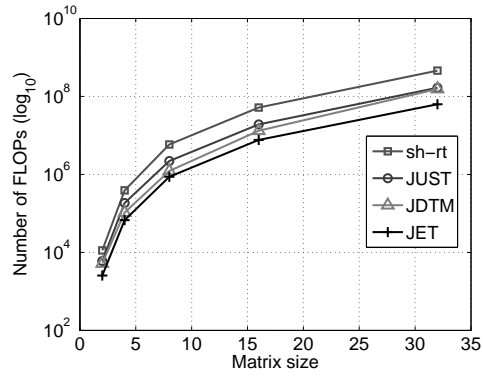


(c) Number of Sweeps (mean)

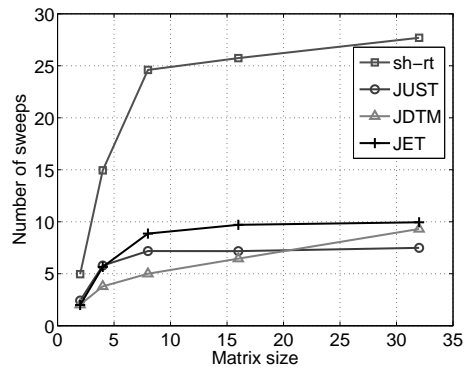
Figure 2: Evolution of the three comparison criteria as a function of the matrix size for a set of 64 matrices with an SNR value of 40 dB.



(a) r_A criterion (median)

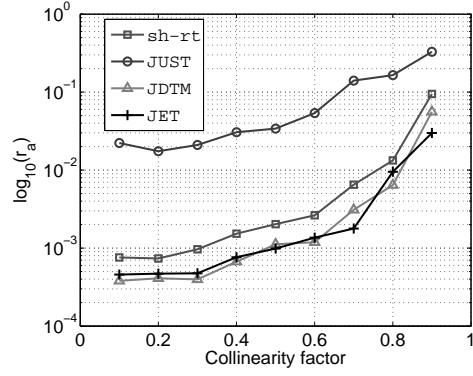


(b) Γ criterion (mean)

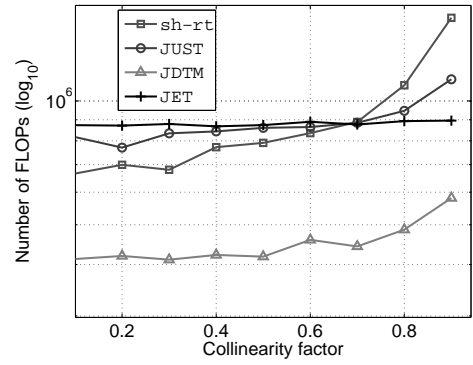


(c) Number of Sweeps (mean)

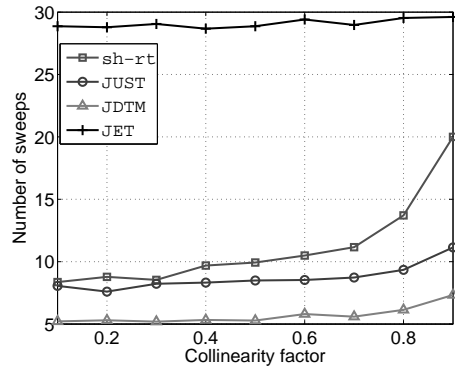
Figure 3: Evolution of the three comparison criteria as a function of the matrix size for a set of 64 matrices with an SNR value of 20 dB.



(a) r_A criterion (median)



(b) Γ criterion (mean)



(c) Number of Sweeps (mean)

Figure 4: Evolution of the three comparison criteria as a function of the correlated factor between columns of matrix A for a set of 10 matrices of size 10 and an SNR value of 80 dB.

415 elized by "truncated" CPD of rank $R_m < R$ which is usually more relevant than the exact CPD:

$$\forall q \in [1; Q]_{\mathbb{N}}, \forall i_q \in [1; I_q]_{\mathbb{N}}, \mathcal{T}_{i_1, \dots, i_Q} = \sum_{r=1}^{R_m} \mathbf{X}_{i_1, r}^{(1)} \mathbf{X}_{i_2, r}^{(2)} \dots \mathbf{X}_{i_Q, r}^{(Q)} + \mathcal{E}_{i_1, \dots, i_Q}, \quad (61)$$

416 where \mathcal{E} is an error term. R_m is the model rank. The DIAG algorithm is compared with an
 417 ALS-ELS algorithm and with the CFS algorithm in various situations by means of Monte-Carlo
 418 experiments. For each new experiment, a noise free tensor is built from factor matrices of R_m
 419 columns whose entries are randomly drawn according to a standard normal distribution. We then
 420 add a Gaussian white noise whose the power is regulated according to the desired SNR value.
 421 The comparison criterion, r_X , is the Normalized Mean Squares Error (NMSE) computed between
 422 actual and estimated factor matrices. Hence for a tensor of order Q we have:

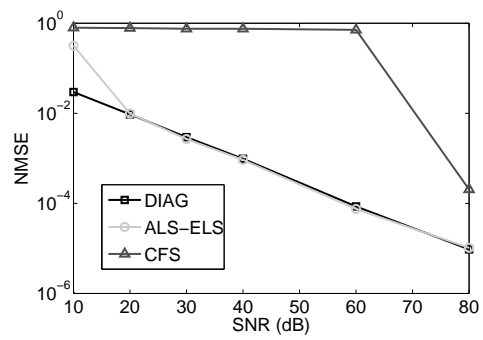
$$r_X = \frac{1}{Q} \sum_{q=1}^Q \text{med} \left(\sqrt{\frac{\text{vec}(\mathbf{X}^{(q)} - \widehat{\mathbf{X}}^{(q)})^T \text{vec}(\mathbf{X}^{(q)} - \widehat{\mathbf{X}}^{(q)})}{\text{vec}(\mathbf{X}^{(q)})^T \text{vec}(\mathbf{X}^{(q)})}} \right), \quad (62)$$

423 where $\widehat{\mathbf{X}}^{(q)}$ denotes the estimation of the factor matrix $\mathbf{X}^{(q)}$, the $\text{vec}(\cdot)$ operator maps a matrix
 424 to a column vector by stacking its columns one below the other and $\text{med}(\cdot)$ denotes the median
 425 value computed from 100 MC experiments. Permutation and scaling ambiguities in the estimated
 426 factor matrices are fixed in the same manner as in [21]. All algorithms were written in-house. The
 427 ALS-ELS algorithm can be found in the tensor package web-page¹. It is stopped as soon as the
 428 relative deviation between two consecutive values of the CPD cost function becomes lower than
 429 10^{-6} or the number of ALS iterations reaches 1000. ELS procedure is run every 5 iterations. For
 430 the decomposition of order-3 tensors, we use the CFS algorithm described in [27] with the best
 431 matching scheme proposed in section 4.2 of [27] whereas higher order tensors were decomposed
 432 using the N-order version described in [28], using the sub-optimal matching rules proposed by
 433 the authors. Implemented versions of DIAG and CFS resort to the JD TM algorithm to solve the
 434 JEVD problem and are stopped as soon as the relative deviation between two consecutive values
 435 of the JEVD cost function becomes lower than 10^{-6} or the number of JEVD iterations reaches
 436 30. Unfolding matrix in the DIAG algorithm is generally chosen to be as squared as possible.
 437 Since the number of test parameters is large, it would be impossible to perform here an exhaustive
 438 comparison. As a consequence we have limited ourselves to some key situations which illustrate
 439 the main features of the proposed approach : *i.* its ability to decompose high order tensors of
 440 high rank, *ii.* tensors with almost collinear factors, *iii.* its insensitivity to over-factoring and *iv.*
 441 its low computational complexity.

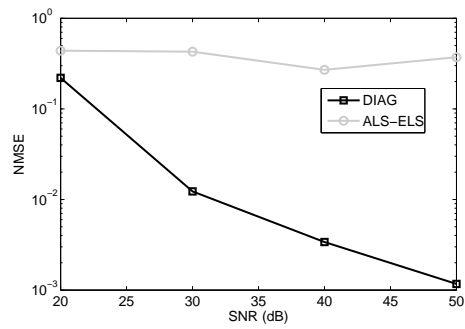
442 4.2.1. High order tensors

443 We first consider a set of 6-order tensors of rank 5 whose all the dimensions are equal to 5.
 444 DIAG parameter P is set to 3 and we vary the SNR from 10 dB to 80 dB. Results are plotted
 445 on figures 5(a). CFS only works for the highest SNR value, probably because this is a difficult
 446 situation for which we are very close to its intrinsic limitation. DIAG provides as accurate
 447 estimations as ALS-ELS for SNR values greater than 10 dB. ALS-ELS fails at 10 dB while
 448 DIAG still works. Notably it clearly outperforms ALS at 10 dB.

¹<http://www.gipsa-lab.grenoble-inp.fr/~pierre.comon/TensorPackage/tensorPackage.html>



(a) $5 \times 5 \times 5 \times 5 \times 5$ tensors of rank 5.



(b) $3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3$ tensors of rank 6.

Figure 5: Median NMSE as a function of the SNR at the output of the ALS and DIAG algorithms applied to high order tensors.

449 We then consider 8-order tensors of rank 6 whose all the dimensions are equal to 3. For this
 450 more difficult case, we vary SNR values from 20 dB to 50 dB. CFS is inapplicable because
 451 of its restrictive necessary condition. Indeed tensor rank is larger than the two largest tensor
 452 dimensions. P is set to 4. Figures 5(b) shows that in spite of ELS, ALS is usefulness here.
 453 Conversely DIAG performs well for the three SNR values above 20 dB.

454 4.2.2. Influence of the collinearity factor

455 In the next two following examples we consider the CPD of rank 4 tensors whose columns
 456 of the random factor matrices are pairwise correlated in all the modes (swamp). For instance,
 457 correlated columns in mode q are built following the scheme of equation (60):

$$\forall r \in [1; R/2]_{\mathbb{N}} \mathbf{x}_{2r}^{(q)} = \nu \mathbf{x}_{2r-1}^{(q)} + (1 - \nu) \mathbf{n}_r^{(q)}. \quad (63)$$

458 Note that it has been shown previously in [21] that in this kind of scenarios ALS performances
 459 are significantly improved by using ELS. First simulation involves third order tensors of size
 460 $4 \times 4 \times 4$. For the second simulation we consider fourth order tensors of size $4 \times 4 \times 4 \times 4$. Results
 461 are plotted on figures 6(a) and 6(b) respectively. DIAG is the only algorithm which works well in
 462 all the considered situations including the most difficult ones (high values of ν) except for $\nu = 0.9$
 463 at order 3. ALS-ELS algorithm fails or is outperformed for largest values of ν ($\nu > 0.5$ at order
 464 3 and $\nu > 0.7$ at order 4). At order 3 CFS results are slightly better than DIAG ones while at
 465 order 4 we find an opposite situation when $\nu < 0.7$. When dealing with higher values only DIAG
 466 works.

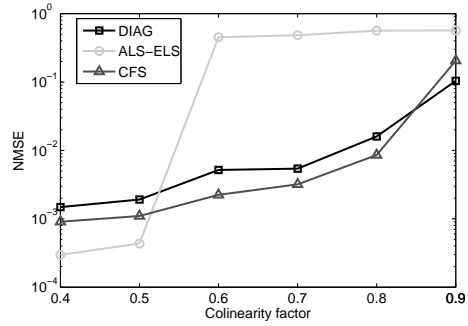
467 We then perform a third simulation with third order tensors of size $4 \times 4 \times 10 \times 4$. This time all
 468 the factors in each mode are mutually correlated:

$$\forall q \in [1; Q]_{\mathbb{N}}, \forall r \in [2; R]_{\mathbb{N}} \mathbf{x}_r^{(q)} = \nu \mathbf{x}_1^{(q)} + (1 - \nu) \mathbf{n}_r^{(q)}. \quad (64)$$

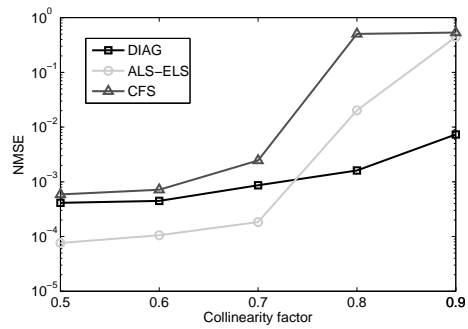
469 Then we vary tensors rank from 3 to 7 while ν is set to 0.8. This simulation again highlights the
 470 main restriction of the CFS algorithm which cannot perform CPD of rank higher than 4. ALS-
 471 ELS results are slightly better than DIAG ones for ranks 3 and 4. On the opposite DIAG appears
 472 as the best option for higher rank values. Notably it still provide satisfactory results for $R = 7$
 473 contrary to ALS-ELS. Finally we compare the complex version of our algorithm DIAG using
 474 the complex JDTM method with the complex version of the ALS algorithm. Complex-valued
 475 tensors are built as for the two first examples of this section but using complex-valued factor
 476 matrices. We consider here third order tensors of size $5 \times 5 \times 5$ and rank 3. Results are displayed
 477 in figure 6(d). Results obtained in the complex field are very similar to those obtained in the real
 478 field for example 1. Indeed ALS starts to fail for $\nu > 0.4$ whereas DIAG still works at $\nu = 0.9$.

479 4.2.3. Over-factoring

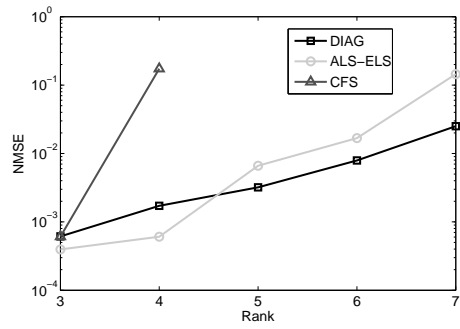
480 In many practical situations the actual model rank R_m of the data tensor to be decomposed
 481 is unknown and it is usually not equal to the tensor rank. Few methods exist for estimating this
 482 number. In addition, these sometimes provide ambiguous or contradictory results. This can lead
 483 to overestimate the model rank. In other words the corresponding decomposition implies more
 484 factors than it is necessary (over-factoring). Suppose that \widehat{R}_m is an overestimation of R_m and Q
 485 is the tensor order. A classical problem with ALS is that the $Q(\widehat{R}_m - R_m)$ extra factors not only model
 486 the additive noise but also the signal. Hence their estimation affects the estimation of the QR_m
 487 actual factors. We study here the impact of over-factoring on DIAG results. For this purpose we
 488 successively compute 5 CPD of 3-order noisy tensors of model rank 3 ($R_m = 3, I_1 = I_2 = I_3 = 7$,



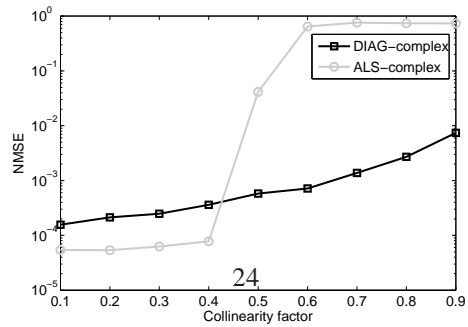
(a) Median NMSE versus collinearity factor for $4 \times 4 \times 4$ tensors of rank 4.



(b) Median NMSE versus collinearity factor for $4 \times 4 \times 4 \times 4$ tensors of rank 4.

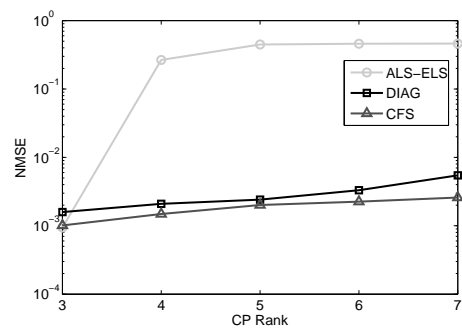


(c) Median NMSE versus tensor rank for $4 \times 4 \times 10 \times 4$ tensors and $\nu = 0.8$.

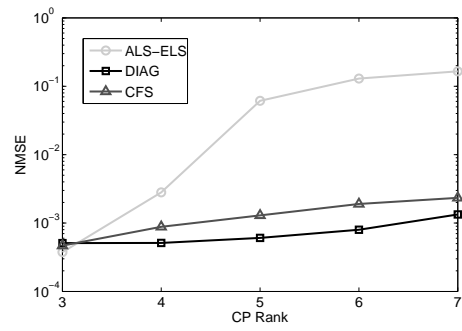


(d) Median NMSE versus tensor rank for $5 \times 5 \times 5$ tensors of rank 3 in the complex field.

Figure 6: Influence of the collinearity factor on the CP decomposition



(a) $7 \times 7 \times 7$ tensors of real rank 3.



(b) $7 \times 7 \times 7 \times 7$ tensors of real rank 3.

Figure 7: Median NMSE as a function of SNR at the output of the ALS-ELS, CFS and DIAG algorithms in the case of over-factoring.

Table 1: Median NMSE, averaged number of iterations and averaged number of flops for small tensors of order 3

Algorithm	$R = 4, I = 10 \times 10 \times 10$			$R = 4, I = 5 \times 100 \times 5$			$R = 4, I = 5 \times 5 \times 100$		
	r_X	N_{it}	Γ	r_X	N_{it}	Γ	r_X	N_{it}	Γ
ALS	2.3×10^{-3}	130	6×10^6	2.5×10^{-3}	140	2×10^7	2.7×10^{-3}	192	2.8×10^7
ALS-ELS	2.3×10^{-3}	47	2×10^6	2.5×10^{-3}	46	7.5×10^6	2.7×10^{-3}	67	1.1×10^7
DIAG	5×10^{-3}	5	3×10^5	6.6×10^{-3}	5	1.7×10^5	2×10^{-2}	5	2.9×10^7
DIAG + ALS-ELS	2.2×10^{-3}	7	6×10^5	2.5×10^{-4}	8	1.4×10^6	2.7×10^{-3}	9	3×10^7

Table 2: Median NMSE, averaged number of iterations and averaged number of flops for large tensors of order 3

Algorithm	$R = 7, I = 50 \times 50 \times 50$			$R = 5, I = 100 \times 100 \times 100$			$R = 4, I = 50 \times 100 \times 50$		
	r_X	N_{it}	Γ	r_X	N_{it}	Γ	r_X	N_{it}	Γ
ALS	5.6×10^{-4}	58	3×10^8	2.3×10^{-4}	23	4.6×10^8	3.4×10^{-4}	47	2.1×10^8
ALS-ELS	5.6×10^{-4}	27	1.8×10^8				3.3×10^{-4}	14	8.2×10^7
DIAG	2.1×10^{-3}	5	5.5×10^7	5.8×10^{-4}	5	2.8×10^8	5.7×10^{-4}	5	3.5×10^7
DIAG + ALS-ELS	5.4×10^{-4}	4	7.6×10^7	2.3×10^{-4}	3	3.3×10^8	3.3×10^{-4}	3	4.9×10^7

489 SNR=50 dB) truncated at rank 3 to 7 respectively. After each CPD and for each estimated factor
490 matrix we keep the three columns that best correspond to the actual 3 factors. Thereby at the end
491 of the process we can compute r_X for each CPD. DIAG results are compared with those of ALS-
492 ELS and CFS on figure 7(a). It is worth mentioning that over-factoring has little impact on DIAG
493 and CFS results while ALS-ELS provides incorrect estimations of the actual factors as soon as
494 the model rank is overestimated. This is an important feature of direct approaches. A second
495 simulation is performed in the same way but with 4-order tensors of dimensions $7 \times 7 \times 7 \times 4$.
496 Model rank is still set to 3. Results are plotted on figure 7(b). Again over-factoring strongly
497 affects ALS-ELS estimates. Conversely DIAG and CFS results are consistent even in the case a
498 large number of extra factors is used. We can also note that at order 4 DIAG is less sensitive to
499 over-factoring than CFS.

500 4.2.4. A trade-off between speed and precision

501
502
503
504

505 We have shown some particular situations for which the DIAG algorithm provides the best
506 estimation results. However one of the main advantages of the proposed approach with respect

Table 3: Median NMSE, averaged number of iterations and averaged number of flops for tensors of order 4

Algorithm	$R = 5, I = 5 \times 10 \times 5 \times 10$			$R = 5, I = 5 \times 5 \times 10 \times 10$			$R = 8, I = 5 \times 5 \times 10 \times 5$		
	r_X	N_{it}	Γ	r_X	N_{it}	Γ	r_X	N_{it}	Γ
ALS	1.5×10^{-3}	50	1.5×10^7	1.4×10^{-3}	31	9.3×10^6	3.5×10^{-3}	11	4.7×10^7
ALS-ELS	1.5×10^{-3}	29	9.4×10^6	1.4×10^{-3}	23	7.3×10^6	3.5×10^{-3}	54	2.4×10^7
DIAG	7.1×10^{-3}	5	7.1×10^5	3.2×10^{-3}	5	6.5×10^5	1.7×10^{-2}	6	6.9×10^5
DIAG + ALS-ELS	1.5×10^{-3}	6	2.7×10^6	1.4×10^{-3}	6	2.5×10^6	3.5×10^{-3}	9	4.6×10^6

Table 4: Median NMSE, averaged number of iterations and averaged number of flops for higher order tensor and tensors with correlated factors (all tensors are rank 4)

Algorithm	$I = 7 \times 7 \times 7 \times 7 \times 7 \times 7$			$I = 10 \times 10 \times 10 \times 10 \times 10$			$I = 10 \times 10 \times 10$		
	r_X	N_{it}	Γ	r_X	N_{it}	Γ	r_X	N_{it}	Γ
ALS	2.3×10^{-4}	21	2.6×10^8	2.5×10^{-4}	28	1.9×10^8	4.3×10^{-1}	831	4×10^7
ALS-ELS	2.6×10^{-4}	19	2.5×10^8				1.2×10^{-3}	444	2×10^7
DIAG	2.3×10^{-4}	4	5.4×10^7	2.8×10^{-4}	4	2×10^7	2.5×10^{-3}	5	3×10^5
DIAG + ALS-(ELS)	2.2×10^{-4}	2	8.3×10^7	2.5×10^{-4}	2	3.5×10^7	6.8×10^{-4}	10	7×10^5

507 to iterative algorithms is its high convergence speed and its lower numerical complexity. Further-
508 more we still have to evaluate DIAG performances in the case of big tensors. For this purpose,
509 we study here 12 representative examples by varying the tensor dimensions and the CPD rank.
510 Examples are classified into 4 groups of three examples: small tensors of order 3, large tensors
511 of order 3, tensors of order 4 and finally, higher order tensors and tensors with correlated CPD
512 factors. Median NMSE values, averaged numbers of iterations N_{it} and averaged numbers of flops
513 Γ are reported in tables 1, 2, 3 and 4 for each example of the four groups and for an SNR value
514 of 40 dB. DIAG is here compared to ALS and ALS-ELS. DIAG estimates can also be used as
515 initial guests of the ALS-ELS procedure. Hence in these tables, "DIAG + ALS-ELS" refers to
516 the ALS-ELS algorithm initialized with DIAG estimates.

517 *Group of small third order tensors.* In the two first examples we show that ALS and ALS-ELS
518 perform slightly better than DIAG in terms of estimation precision. However on average DIAG
519 only requires 5 JDJM iterations to converge against 46 and 140 for ALS-ELS and ALS, respec-
520 tively. Hence Γ_{DIAG} is 10 to 100 times lower than Γ_{ALS} and $\Gamma_{ALS-ELS}$. Another interesting point
521 is that the DIAG + ALS-ELS procedure limits the number of ALS iterations to 7-8 (the averaged
522 number of iterations reported in the table for DIAG + ALS-ELS is the averaged number of ALS-
523 ELS iterations used after an initialization with DIAG) and we can see from these results that this
524 is enough to obtain a precision similar or better than that of ALS-ELS. Consequently, the numer-
525 ical complexity of this approach is 3 to 10 times lower than those of Γ_{ALS} and $\Gamma_{ALS-ELS}$.
526 The last example is similar to the second one but tensor dimensions have been permuted so that
527 only the DIAG unfolding matrix is different. Here DIAG results are degraded both in terms of
528 precision and numerical complexity. We can conclude that if it is possible, it is better to not place
529 the larger dimension of the tensor at the end.

530 *Group of large third order tensors.* We consider now third order tensors whose all the dimen-
531 sions are equal to 50 or 100. As a consequence, the CPD rank is far lower than the tensor
532 dimensions and all algorithms perform better and need fewer iterations. This explain that the gap
533 between the different algorithms is narrowing. However we can still draw the same general con-
534 clusion: DIAG remains the cheapest solution and DIAG + ALS-ELS provides the same precision
535 than ALS and ALS-ELS for a lower numerical complexity.

536 *Group of fourth order tensors.* We obtain the same kind of results that with the first group so
537 that DIAG + ALS-ELS still appears to give the best compromise between precision and cost.
538 One should note however that in the last case DIAG is by far the cheapest whereas its results
539 regarding the NMSE are not as good. This is explained by the fact than the rank is greater than
540 three of the tensor dimensions and slightly lower than the remaining one.

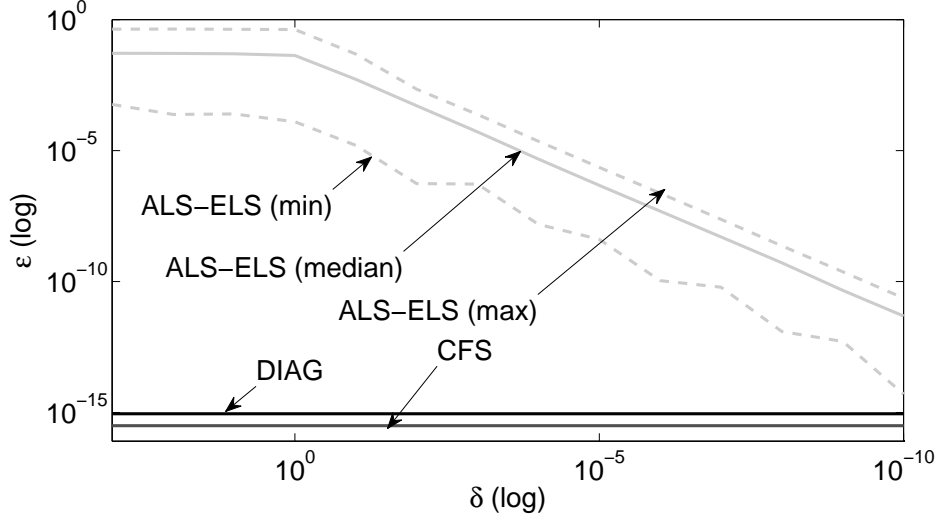


Figure 8: Decomposition of the Paatero tensor, evolution of the minimal median and maximal values of the ALS-ELS estimation error, according to a distance between the solution and the starting values and comparison with CFS and DIAG results.

541 *Higher order tensors and tensors with correlated CPD factors.* The first example of this group
542 deals with fifth order tensors (for which our version of ELS is not working). For the second
543 example we consider sixth order tensors. It is worth mentioning here that in both cases DIAG
544 provides as accurate estimates as ALS and ALS-ELS do while its numerical complexity remains
545 largely lower. Now looking at the last example with correlated factors, one can first note that
546 ALS doesn't work whereas ALS-ELS is more accurate than DIAG. The price to paid is a very
547 high number of iterations (444) and an increased computational cost (about 2×10^7 flops) against
548 5 iterations and about 3×10^5 flops for DIAG. In this case one should not that DIAG + ALS-ELS
549 is significantly better than ALS-ELS in terms of estimation precision for a limited numerical
550 complexity.

551
552 As a first conclusion DIAG appears as a good trade-off between estimation precision, speed
553 and numerical complexity. Besides, the DIAG + ALS-ELS procedure provides a similar or
554 better precision than that of ALS-ELS whereas its numerical complexity remains quite close to
555 that of DIAG. Hence by combining both algorithms one can achieve the best precision, a good
556 convergence speed and a reduced numerical complexity.

557 4.3. Results on the Paatero tensor

558 In [44], Paatero introduced a very simple 3-order tensor of size $2 \times 2 \times 2$ which has the
559 following form:

$$\mathcal{T} = \begin{bmatrix} 0 & 1 & e & 0 \\ 1 & d & 0 & h \end{bmatrix}. \quad (65)$$

560 Let's define its determinant Δ by:

$$\Delta = 4h + d^2e. \quad (66)$$

561 Then, it can be shown that the equation $\Delta = 0$ partitions the space into two subspaces, which
 562 hence have a non zero volume: The inequality $\Delta > 0$ defines the subspace of rank-2 tensors,
 563 whereas $\Delta < 0$ defines the subspace of rank-3 tensors. Finally, the closed set of tensors of rank 1
 564 lies on the hypersurface $\Delta = 0$. [21]

565 Some of these tensors have the particularity of misleading any iterative algorithm, although
 566 the chosen starting value is close to the solution. As an example, Paatero notably consider in [44]
 567 to decompose the tensor defined by $(e, d, h) = (30, 0.26, 0.34)$ from the initial value $(e, d, h) =$
 568 $(30, 0.3, 0.12)$. This tensor belongs to the rank 2 subspace but it is close to the variety $\Delta = 0$. Its
 569 decomposition is given by the three following factor matrices:

$$569 \mathbf{A} = \begin{bmatrix} 1/x & -1/x \\ y_1 & y_2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1/x & -1/x \\ y_1 & y_2 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1/x & -1/x \\ y_3 & y_4 \end{bmatrix}, \quad (67)$$

570 with: $x = (4h/e + d^2)^{\frac{1}{6}}$, $y_2 = (x^3 - d)/(2x)$, $y_1 = x^2 - a_2$, $y_4 = h/(y_2(y_1 + y_2))$ and $y_3 = y_2 y_4 / y_1$.

571 Later in [21], authors confirmed that in this case, even the most efficient iterative algorithms
 572 such as ALS-ELS and Levenberg-Marquardt get stuck in a local minimum of the cost function,
 573 leading to a very bad estimation of the factor matrices. Actually, since the iterative algorithms
 574 works by successive optimization of rank-2 tensors, they cannot take the shorter paths to the
 575 solution which could cross the space of rank-3 tensors. Thereby, this is an other typical situation
 576 where direct algorithms can help. In order to see this we have reproduced the experiment here not
 577 only for the Paatero starting values but for different starting values around the solution. Hence
 578 we define a parameter δ such that the initial factor matrices of the ALS-ELS, $\mathbf{A}^{(0)}$, $\mathbf{B}^{(0)}$

$$579 \mathbf{A}^{(0)} = \mathbf{A} + \delta \mathbf{E}_A; \quad \mathbf{B}^{(0)} = \mathbf{B} + \delta \mathbf{E}_B; \quad \mathbf{C}^{(0)} = \mathbf{C} + \delta \mathbf{E}_C, \quad (68)$$

580 where \mathbf{E}_A , \mathbf{E}_B and \mathbf{E}_C are matrices of size 2×2 whose elements are randomly drawn according
 581 to a standard normal law. We now define ϵ as the mean estimation error upon the three estimated
 582 factor matrices, $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{C}}$:

$$583 \epsilon = \frac{1}{3} \left(\frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F} + \frac{\|\mathbf{B} - \widehat{\mathbf{B}}\|_F}{\|\mathbf{B}\|_F} + \frac{\|\mathbf{C} - \widehat{\mathbf{C}}\|_F}{\|\mathbf{C}\|_F} \right). \quad (69)$$

584 The ALS-ELS algorithm is run 500 times on the tensor \mathcal{T} , with a new draw of the \mathbf{E}_A , \mathbf{E}_B and
 585 \mathbf{E}_C matrices at each time, and for different values of δ comprised between 1000 and 10^{-10} . We
 586 present on figure 8 the plots of the median, minimal and maximal values of $\epsilon_{ALS-ELS}$ according
 587 to the δ value. For comparison, both ϵ_{DIAG} and ϵ_{CF5} values are also reported on the figure. It can
 588 be seen that the iterative algorithm needs a very good initialisation in order to get an estimation
 589 precision close to the machine precision. Recall that we are looking for an exact decomposition
 590 since the considered tensors are noise free. Conversely, direct algorithms such as the closed-
 form solution or DIAG provide a perfect decomposition of \mathcal{T} and a thus an exact estimation of
 the factor matrices.

591 5. Application to fluorescence spectroscopy

592 A good application example of the CPD is found in fluorescence spectroscopy since after
 593 some numerical corrections measured data can be modelled by a CPD with physical meaning.
 594 Standard spectrofluorimeters allow to measure the intensity of the fluorescence signal emitted

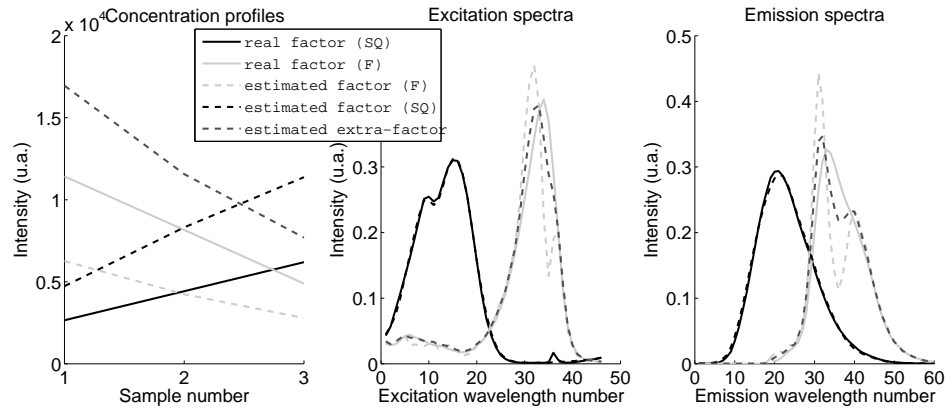


Figure 9: CPD factors of the fluorescence tensor using ALS.

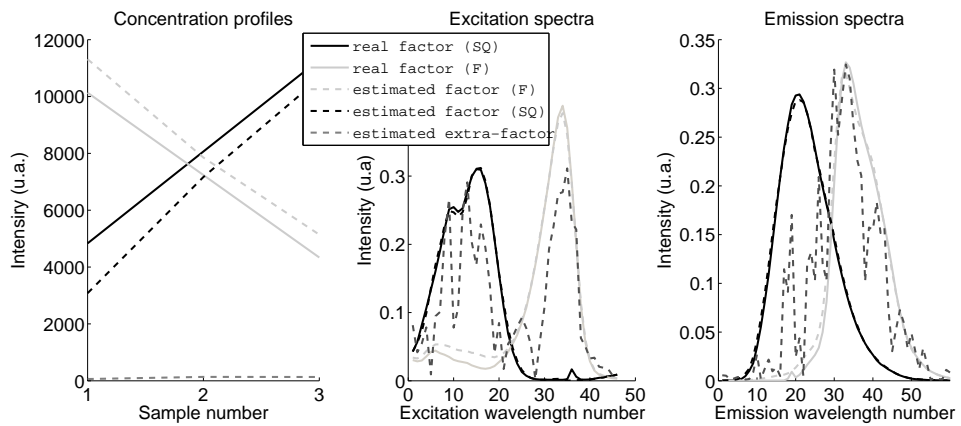


Figure 10: CPD factors of the fluorescence tensor using DIAG-JDTM.

595 by a diluted solution at wavelength λ_j by exciting the solution at wavelength λ_i . Hence, by
596 scanning the excitation-emission couples (λ_i, λ_j) one obtains an $I \times J$ matrix of fluorescence
597 which is called the Fluorescence Excitation-Emission Matrix (FEEM) of the solution. In many
598 applications one have to measure a FEEM set corresponding to a set of K solutions and thus
599 obtains a fluorescing data tensor \mathcal{X} of order 3 and size $I \times J \times K$ which contains the K FEEM.
600 Solutions are often mixtures of a small number, R_m , of diluted fluorescing chemical species
601 (fluorophores). Fluorophore concentrations vary from a solution to an other. Hence fluorophore
602 r is characterized by its excitation spectrum, $e_r(\lambda_i)$, its fluorescence emission spectrum, $f_r(\lambda_j)$
603 and the variation of its concentration through the solution set (concentration profile), $c_r(k)$. In
604 practice one wants to recover e_r , f_r and c_r ($r = 1 \cdots R_m$) from the measured FEEMs. It can be
605 shown that after removing scattering effects and correcting (or preventing) inner filter effects, the
606 contribution of each fluorophore to the solution signal is linear in excitation, in emission and in
607 concentration. In other words we have:

$$\mathcal{X}_{i,j,k} = \sum_{r=1}^{R_m} E_{i,r} F_{j,r} C_{k,r}, \quad (70)$$

608 where $E_{i,r} = e_r(\lambda_i)$, $F_{j,r} = f_r(\lambda_j)$ and $C_{k,r} = c_r(k)$, so that the CPD solves this inverse problem in
609 a deterministic way. This is the reason why CPD has been largely applied to analyze FEEM sets
610 since original works of Bro in this area [9, 10].

611 In most applications of fluorescence spectroscopy the number of fluorophores which defines the
612 model rank of the decomposition is unknown and has to be estimated. However few methods
613 exist and can give contradictory results and lead to over-factoring. A good example of this
614 situation can be found in [45]. This would be acceptable if in each estimated factor matrix
615 one obtain the R_m real factors aside with extra factors whose the contributions are almost null.
616 Actually this is not the case with ALS which is very commonly used for analyzing this kind of
617 data. Therefore this problem remains an important issue of FEEM analysis. In order to highlight
618 the reliability of DIAG in this context we consider here a fluorescence tensor which contains
619 the fluorescence intensity of 3 distinct mixtures of two fluorophores (fluorescein and quinine
620 sulphate) measured at 46×71 excitation-emission wavelength couples. Hence the tensor size is
621 $3 \times 46 \times 71$ and the model rank is 2. CPD of rank 3 were then used to decompose the tensor.
622 ALS and DIAG results are reported on figure 9 and 10 respectively and compared to the actual
623 factors after removing permutation and scaling indeterminacy. Excitation and Emission factors
624 are normalized so that factor contributions are condensed in the concentration mode. Actual
625 concentration profiles are perfectly known since these are laboratory mixtures and actual spectra
626 were measured aside from pure solutions of fluorescein and quinine sulphate. This is a simple
627 case for which both algorithms give perfect results when the good model rank ($R_m = 2$) is
628 selected (data not shown). However ALS sensitivity to over-factoring effect in a concrete case
629 clearly appears here. Indeed actual factors are not well estimated (notably the fluorescein spectra
630 and the concentration profiles). Moreover contribution of the extra factor to the decomposition
631 is significant. Recall that this factor has no physical meaning. On the opposite DIAG results are
632 satisfying notably regarding the estimated spectra. One can verify that the contribution of the
633 extra factor is almost null.

634 6. Conclusion

635 We have described in this paper a CPD algorithm that takes advantage of the link between
636 CPD and Joint EVD in an original way. A JEVD algorithm has been conjointly proposed. Com-

637 computational complexities and extension to the complex field have been given for both algorithms.
638 Numerical simulations point out the efficiency of the proposed JD TM algorithm to solve the
639 JEVD problem. This algorithm usually offers more accurate results than its competitors espe-
640 cially in the most difficult cases involving big matrices and low SNR values. In terms of numer-
641 ical complexity, the JD TM algorithm also provides good performances thanks to a remarkably
642 low and stable number of iterations.

643 Classical iterative CPD algorithms such as ALS are usually efficient but suffer from convergence
644 problem, notably in case of highly correlated factors, and are very sensitive to over-factoring.
645 In addition they require a large number of iterations to reach the convergence. ELS allows us
646 to reduce this number and deals with correlated factors in some situations but it remains useless
647 in case of over-factoring. In addition, we have seen that there are some simple cases for which
648 iterative approaches consistently fail for theoretical reasons.

649 In this context direct approaches such as the proposed DIAG algorithm have been designed
650 to prevent such issues. First the DIAG algorithm involves a limited iterative procedure which
651 requires very few iterations hence limiting global computational cost of the algorithm. Second it
652 is insensitive to over-factoring thanks to the initial SVD which is independent of the chosen rank.
653 These features have been verified in this paper by using many numerical simulations. Notably we
654 have shown that DIAG was able to deal with highly correlated factors in all the modes or a large
655 number of extra factor in case of over-factoring. Furthermore our results also demonstrates that
656 DIAG is very efficient to decompose high order tensors. Finally it is a very fast algorithm with a
657 lower computational complexity than ALS or ALS-ELS, notably in the case of small tensors or
658 correlated factors.

659 As a counterpart, DIAG implies more restricted necessary conditions on the CPD rank than
660 ALS. Therefore ALS-ELS is more accurate than DIAG when the rank is close to DIAG intrinsic
661 limit. This is usually not the case in fluorescence spectroscopy applications for which at least one
662 tensor dimension is largely greater than the model rank. In addition it has been shown that DIAG
663 results can be improved by adding very few ALS iterations with a limited impact on the overall
664 numerical complexity. Conversely, one should note that over-factoring is an important issue of
665 FEEM analysis. This makes DIAG an attractive alternative to the classical ALS procedure for
666 the CPD of fluorescence tensors, as it has been shown on a practical example.

667 Eventually, comparing to the CFS algorithm which is also a reference direct CPD approach,
668 DIAG is a cheaper algorithm since it only involves one JEVD procedure and does not require to
669 compare several estimates of the factor matrices. But its main advantage definitely comes from
670 the necessary condition of CFS which is more restricted than DIAG's one. Hence there are many
671 simple cases that CFS cannot handle. More generally CFS accuracy decreases as we get closer
672 to its intrinsic limit. Otherwise CFS results are close to DIAG results.

673 **Appendix A. Proof of proposition 1**

674 **Proof 2.** $C_{CFS} \Rightarrow C_{DIAG}$ is trivial. Indeed if $\exists (q_1, q_2) \in [1; Q]_{\mathbb{N}}^2$, $q_1 \neq q_2$ such that $I(q_1) \geq$
675 R and $I(q_2) \geq R$ then taking any permutation f_1 of the Q first natural number such that $f_1(1) = q_1$
676 and $f_1(2) = q_2$, $P = 1$ and $q_s = Q$ ensures C_{DIAG} .

677

Let now suppose that C_{DIAG} is true and that C_{ALS} is false, i.e.:

$$\exists P \in [2; Q-1]_{\mathbb{N}}, \exists f_I, \exists q_s > P \text{ and } q \leq Q \text{ such that: } 1) \prod_{i=1}^P I(f_I(i)) \geq R, \quad (\text{A.1})$$

$$2) \prod_{\substack{i=P+1 \\ i \neq q_s}}^Q I(f_I(i)) \geq R, \quad (\text{A.2})$$

$$3) \prod_{i=1}^Q I(i) < RI(q). \quad (\text{A.3})$$

678

Since we have $1 \leq q \leq Q$ thus $q \in \{f_I(1), \dots, f_I(P)\} \cup \{f_I(P+1), \dots, f_I(Q)\}$.

679

• We first assume that $q \in \{f_I(1), \dots, f_I(P)\}$. 2) and 3) give:

$$\frac{1}{I(f_I(q_s))} \prod_{i=P+1}^Q I(f_I(i)) > \frac{1}{I(q)} \prod_{i=1}^Q I(i), \quad (\text{A.4})$$

$$\frac{1}{I(f_I(q_s))} \prod_{i=P+1}^Q I(f_I(i)) > \frac{1}{I(q)} \prod_{i=1}^Q I(f_I(i)), \quad (\text{A.5})$$

$$I(q) > I(f_I(q_s)) \prod_{i=1}^P I(f_I(i)). \quad (\text{A.6})$$

680

Since $q \in \{f_I(1), \dots, f_I(P)\}$,

$$\prod_{i=1}^P I(f_I(i)) = I(q) \prod_{\substack{i=1 \\ i \neq f_I^{-1}(q)}}^P I(f_I(i)) \quad (\text{A.7})$$

681

thereby,

$$1 > I(f_I(q_s)) \prod_{\substack{i=1 \\ i \neq f_I^{-1}(q)}}^P I(f_I(i)) \quad (\text{A.8})$$

682

which is absurd.

683

• Now we assume that $q \in \{f_I(P+1), \dots, f_I(Q)\}$. Thereby,

$$I(q) < \prod_{i=P+1}^Q I(f_I(i)), \quad (\text{A.9})$$

$$I(q) \prod_{i=1}^P I(f_I(i)) < \prod_{i=1}^Q I(f_I(i)), \quad (\text{A.10})$$

684

while 1) and 3) give:

$$I(q) \prod_{i=1}^P I(f_I(i)) > \prod_{i=1}^Q I(i), \quad (\text{A.11})$$

$$I(q) \prod_{i=1}^P I(f_I(i)) > \prod_{i=1}^Q I(f_I(i)), \quad (\text{A.12})$$

685 *which is contradictory to (A.10).*

686 *Therefore if C_{DIAG} is verified then C_{ALS} is verified.*

687 **References**

- 688 [1] Carroll, J.D., Chang, J.J.: Analysis of Individual Differences in Multidimensional Scaling via N-Way Generalization
689 of Eckart-Young Decomposition, *Psychometrika*, 35 (3) pp. 283-319 (1970)
- 690 [2] Becker, A., Comon, P., Albera, L., Haardt, M., Merlet, I.: Multi-way Space-Time-Wave-Vector analysis for EEG
691 source separation. *Signal Processing*, to appear in
- 692 [3] Deburchgraeve W., Cherian P.J., De Vos M., Swarte R.M., Blok J.H., Visser G.H., Govaert P., Van Huffel S.: Neonatal
693 seizure localization using PARAFAC decomposition, *Clinical Neurophysiology*, 120, pp. 1787-1796 (2009)
- 694 [4] Vanderperren K., De Vos M., Mijovic B., Ramautar J.R., Novitskiy N., Vanrumste B., Stiers P., Van den Bergh
695 B.R.H., Wagemans J., Lagae L., Sunaert S., Van Huffel S.: PARAFAC on ERP data from a visual detection task during
696 simultaneous fMRI acquisition, in *Proc. of the International Biosignal Processing Conference.*, Berlin, Germany,
697 Jul. 2010, 103, pp. 1-4 (2010)
- 698 [5] Karfoul, A. , Albera, L., De Lathauwer, L.: Iterative methods for the canonical decomposition of multi-way arrays:
699 Application to blind underdetermined mixture identification. *Signal Processing*, 91 (8), pp. 1789-1802 (2011)
- 700 [6] Sidiropoulos, N.D., Giannakis, G.B., Bro, R.: Blind PARAFAC receivers for DS-CDMA systems, *IEEE Transactions*
701 *On Signal Processing*, 48 (8), pp. 810-823 (2000)
- 702 [7] Nion, D., Sidiropoulos, N.D., Tensor Algebra and Multi-dimensional Harmonic Retrieval in Signal Processing for
703 MIMO Radar, *IEEE Trans. on Signal Processing*, 58 (11), pp. 5693-5705 (2010)
- 704 [8] Bürgisser, P., Clausen, M., Shokrollahi, M.A.: *Algebraic Complexity Theory*, 315, Springer (1997)
- 705 [9] Bro, R.: PARAFAC, Tutorial and Applications, *Chemom. Intel. Lab. Syst.*, 38, pp. 149-171 (1997)
- 706 [10] Stedmon, C.A., Markager, S., Bro, R.: Tracing dissolved organic matter in aquatic environments using a new
707 approach to fluorescence spectroscopy, *Marine Chemistry*, 82 (3-4), pp. 239-254 (2003)
- 708 [11] Harshman, R.A.: Determination and proof of minimum uniqueness conditions for PARAFAC-1," *UCLA Working*
709 *Papers in Phonetics*, 22, pp. 111-117 (1972)
- 710 [12] Kruskal, J.B.: Three-Way Arrays: Rank and Uniqueness of Trilinear Decompositions, *Linear Algebra and Appli-*
711 *cations*, 18, pp. 95-138 (1977)
- 712 [13] Ten Berge, J.M.F., Sidiropoulos, N.D.: On uniqueness in CANDECOMP/PARAFAC, *Psychometrika*, 67, pp. 399-
713 409 (2002)
- 714 [14] Jiang, T., Sidiropoulos, N.D.: Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC
715 and bilinear models with constant modulus constraints, *Trans. on Sig. Proc.*, 52 (9), pp. 2625-2636 (2004)
- 716 [15] De Lathauwer, L.: A Link between Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix
717 Diagonalization, *Journal on Matrix Analysis*, 28 (3), pp. 642-666 (2006)
- 718 [16] Stegeman, A., Sidiropoulos, N.D.: On Kruskal's uniqueness condition for the Candecom/Parafac decomposition,
719 *Linear Algebra and its Applications*, 420, pp. 540-552 (2007)
- 720 [17] Hitchcock, F.L.: The expression of a tensor or a polyadic as a sum of products, *J. Math. Phys*, 6 (1), pp. 164-189
721 (1927)
- 722 [18] Harshman, R.A.: Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal
723 factor analysis, *UCLA Working Papers in Phonetics*, 16, pp. 1-84 (1970)
- 724 [19] Tendeiro, J., Bennani Dosse, M., Ten Berge, J.M.F., First and second-order derivatives for CP and INDSCAL,
725 *Chemometrics and Intelligent Laboratory Systems*, 102 (1), pp. 27-36 (2011)
- 726 [20] Tomasi, G., Bro, R.: A comparison of algorithms for fitting the PARAFAC model, *Comp. Stat. Data Anal.*, 50, pp.
727 1700-1734 (2006)
- 728 [21] Comon, P., Luciani, X., De Almeida, A.L.F.: Tensor Decompositions, Alternating Least Squares and other Tales,
729 *Journal of Chemometrics*, 23 (9), pp. 393-405 (2009)
- 730 [22] Acar E., Dunlavy D.M., Kolda, T.G.: Scalable Tensor Factorizations with Missing Data, In *SDM10: Proceedings*
731 *of the 2010 SIAM International Conference on Data Mining*, pp. 701-712 (2010)
- 732 [23] Kindermann, S., Navasca, C.: New algorithms for tensor decomposition based on a reduced functional, *Numer.*
733 *Linear Algebra Appl.* (2013)
- 734 [24] Rajih, M., Comon, P., Harshman, R.: Enhanced Line Search : A Novel Method to Accelerate PARAFAC, *SIAM*
735 *Journal on Matrix Analysis Appl.*, 30 (3), pp. 1148-1171 (2008)
- 736 [25] Faber, N.M., Buydens, L.M.C., Kateman, G.: Generalized rank annihilation method: II. Bias and variance in the
737 estimated eigenvalues, *J. Chemom.*, 8, pp.181-203 (1994)

- 738 [26] Faber N.M., Bro R., Hopke P.K.: Recent developments in CANDECOMP/PARAFAC algorithms: a critical review,
739 Chemom. Intel. Lab. Syst., 65 (1), pp. 119-137 (2003)
- 740 [27] Roemer, F., Haardt, M.: A closed-form solution for Parallel Factor (PARAFAC) Analysis, In IEEE ICASSP'2008,
741 pp. 2365-2368 (2008)
- 742 [28] Roemer, F., Haardt, M.: A closed-form solution for multilinear PARAFAC decompositions, In IEEE SAM 2008,
743 pp. 487-491 (2008)
- 744 [29] De Lathauwer, L., De Moor, B., Vandewalle, J.: Computation of the Canonical Decomposition by Means of a
745 Simultaneous Generalized Schur Decomposition, SIAM Journal on Matrix Analysis and Applications, 26, pp. 295-
746 327 (2001)
- 747 [30] Albera, L., Ferréol, A., Comon, P., Chevalier, P.: Blind Identification of Overcomplete MixturEs of sources
748 (BIOME), Linear Algebra and its Applications, 391, pp. 3-11 (2004)
- 749 [31] Luciani, X., Albera, A.: Joint Eigenvalue Decomposition using Polar Matrix Factorization, In LVA/ICA 2010, pp.
750 612-619 (2010)
- 751 [32] Luciani, X., Albera, A.: Semi-algebraic canonical decomposition of multi-way arrays and joint eigenvalue decom-
752 position, In IEEE ICASSP'2011, pp. 4104-4107 (2011)
- 753 [33] Karfoul, A., Albera, L., Birot, G.: Blind underdetermined mixture identification by joint canonical decomposition
754 of HO cumulants, IEEE Trans. Signal Proc., 58 (2), pp. 638-649 (2010)
- 755 [34] Ruhe, A.: On the quadratic convergence of a generalization of the Jacobi method to arbitrary matrices, BIT Nu-
756 merical Mathematics, 8, pp. 210-231 (1968)
- 757 [35] Haardt, M., Nosseck, J. A.: Simultaneous Schur decomposition of several nonsymmetric matrices to achieve auto-
758 matic pairing in multidimensional harmonic retrieval problems, IEEE Trans. Signal Proc., 46, pp. 161-169 (1998)
- 759 [36] Strobach, P.: Bi-iteration multiple invariance subspace tracking and adaptive ESPRIT, IEEE Trans. Signal Proc.,
760 48, pp. 442-456 (2000)
- 761 [37] Fu, T., Gao, X.: Simultaneous Diagonalization with Similarity Transformation for Non-defective Matrices. In IEEE
762 ICASSP'2006, pp. 1137-1140 (2006)
- 763 [38] Goldstine, H.H.: Horwitz, L.P.: A procedure for the diagonalization of normal matrices, J. ACM, 6 (2), pp. 176-195
764 (1959)
- 765 [39] Eberlein, P.J.: A Jacobi-like method for the automatic computation of eigenvalues and eigenvectors of an arbitrary
766 matrix, Journal of the Society for Industrial and Applied Mathematics, 10 (1), pp. 74-88 (1962)
- 767 [40] Souloumiac, A.: Nonorthogonal joint Diagonalization by Combining Givens and Hyperbolic Rotations, IEEE
768 Trans. Signal Proc., 57 (6), pp. 2222-2231 (2009)
- 769 [41] Iferroudjene, R., Abed Meraim, K., Belouchrani, A.: A New Jacobi-like Method for Joint Diagonalization of
770 Arbitrary non-defective Matrices, Applied Mathematics and Computation 211, pp. 363-373 (2009)
- 771 [42] Cardoso, J.-F., Souloumiac, A.: Jacobi Angles for Simultaneous Diagonalization, SIAM Journal on Matrix Analy-
772 sis and Applications, 17 (1), pp. 161-164 (1996)
- 773 [43] De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition, SIAM Journal on
774 Matrix Analysis and Applications, 21 (4), pp. 1253-1278, (2000)
- 775 [44] Paatero, P.: Construction and analysis of degenerate Parafac models, Journal of Chemometrics, 14, pp. 285-299
776 (2000)
- 777 [45] Luciani, X., Mounier, S., Paraquetti, H.H.M., Redon, R., Lucas, Y., Bois, A., Lacerda, L.D., Raynaud, M., Ripert,
778 M.: Tracing of dissolved organic matter from the SEPETIBA Bay (Brazil) by PARAFAC analysis of total lumines-
779 cence matrices, Marine Environmental Research, 65 (2), pp. 148-157 (2008)