



A two-step algorithm for joint EigenValue decomposition - Application to canonical polyadic decomposition of fluorescence spectra

R. André, X. Luciani, Laurent Albera, E. Moreau

► To cite this version:

R. André, X. Luciani, Laurent Albera, E. Moreau. A two-step algorithm for joint EigenValue decomposition - Application to canonical polyadic decomposition of fluorescence spectra. *Chemometrics and Intelligent Laboratory Systems*, 2020, 206, pp.104065. 10.1016/j.chemolab.2020.104065 . hal-02960127

HAL Id: hal-02960127

<https://hal.science/hal-02960127>

Submitted on 8 Oct 2020

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

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

This work is collective, respective contributions could not and should not be distinguished

Rémi André:

Conceptualization; Data curation; Formal analysis; Funding acquisition; Investigation; Methodology; Project administration; Resources; Software; Supervision; Validation; Visualization; Roles/Writing – original draft; Writing – review & editing

Xavier Luciani:

Conceptualization; Data curation; Formal analysis; Funding acquisition; Investigation; Methodology; Project administration; Resources; Software; Supervision; Validation; Visualization; Roles/Writing – original draft; Writing – review & editing

Laurent Abera:

Conceptualization; Data curation; Formal analysis; Funding acquisition; Investigation; Methodology; Project administration; Resources; Software; Supervision; Validation; Visualization; Roles/Writing – original draft; Writing – review & editing

Eric Moreau:

Conceptualization; Data curation; Formal analysis; Funding acquisition; Investigation; Methodology; Project administration; Resources; Software; Supervision; Validation; Visualization; Roles/Writing – original draft; Writing – review & editing

A Two-step Algorithm for Joint EigenValue Decomposition - Application to Canonical Polyadic Decomposition of Fluorescence Spectra

Rémi André^a, Xavier Luciani^a, Laurent Albera^{b,c}, Eric Moreau^a

^aAix Marseille Université, Université de Toulon, CNRS, LIS, F-13000 Marseille, France

^bUniversité de Rennes 1, LTSI, F-35000 Rennes, France

^cINSERM, U1099, F-35000 Rennes, France

Abstract

In this paper, we propose a new Joint EigenValue Decomposition (JEVD) algorithm. JEVD problem belongs to the family of joint diagonalization problems. Hence, JEVD algorithms aim at estimating the common basis of eigenvectors of a matrix set. This problem occurs in many signal processing applications. It has notably allowed to develop efficient algorithms for the Canonical Polyadic Decomposition (CPD) of multiway arrays. The proposed JEVD algorithm is based on an original two-step approach. The first step consists in transforming the considered matrix set into a set of positive definite matrices. In this purpose, we introduce an *ad hoc* joint symmetrization algorithm. This first step allows us to transform the JEVD problem into a simpler orthogonal joint diagonalization problem. The second step is then performed using an efficient orthogonal joint diagonalization algorithm of the literature. Eventually, the performance of the proposed approach is deeply investigated in the CPD context of multidimensional fluorescence data. More particularly, we consider difficult scenarios such as the cases of an overestimated rank and highly correlated factors.

Keywords: Fluorescence spectroscopy, CPD, PARAFAC, Multiway analysis, Joint diagonalization, JEVD, Positive definite matrix

1. Introduction

In many data sciences or signal processing applications, the collected data has a multidimensional structure and can thus be stored in multiway arrays (tensors). In this context, multilinear algebra provides efficient tools to analyze such data sets. Indeed, tensor decompositions offer a large variety of multidimensional models with some interesting properties. Among all tensor decomposition models, Canonical Polyadic Decomposition (CPD) also known as PARALLEL FACTOR decomposition (PARAFAC) [1, 2] decomposes a multiway array as a linear combination of a finite number of rank-one tensors. Moreover, this decomposition is essentially unique under some mild conditions [3, 4]. For these reasons, CPD is largely used to estimate multilinear mixtures of source signals and plays a growing role in signal and image processing such as in brain source localization [5, 6], face recognition [7] and blind source separation [8, 9]. In chemometrics, CPD directly models the data measured from fluorescence spectroscopy [10, 11].

CPD decomposes multiway arrays as a combination of factors matrices. For instance, the rank- N CPD of a three way tensor $\mathcal{T} \in \mathbb{R}^{P \times Q \times R}$ is given by:

$$\mathcal{T}_{p,q,r} \simeq \sum_{n=1}^N A_{p,n} B_{q,n} C_{r,n}, \quad \forall (p, q, r) \quad (1)$$

where matrices \mathbf{A} , \mathbf{B} and \mathbf{C} are the factor matrices. It is noteworthy that the smallest value of N for which equation (1) is exact is called the tensor rank.

Many algorithms have been proposed in order to compute the CPD of a given tensor. All of them aim at minimizing iteratively a specific cost function built (in)directly from the data. They can be classified in three main families. First, there are alternating or block coordinate approaches such as Alternating Least Squares (ALS) [2]. These algorithms update different subsets of factors alternatively at each iteration. Secondly, we find the well-known first/second order optimization methods based on (conjugate) descent [12], Quasi-Newton and Levenberg-Marquardt [13] optimization strategies. Finally, semi-algebraic approaches are also available, which generally rephrase the CPD as a Joint Diagonalization (JD) problem [14, 15, 16, 17].

Joint diagonalization consists in finding a matrix, which transforms (by congruence or by similarity) a given set of square matrices into a set of diagonal matrices. In this paper, we focus on the JD problem by similarity well-known as the Joint EigenValue Decomposition (JEVD) problem. In other words, JEVD consists in jointly diagonalizing a set of non-defective matrices of size $N \times N$ sharing the same eigenvectors basis \mathbf{W} :

$$\mathbf{M}^{(k)} = \mathbf{W} \mathbf{D}^{(k)} \mathbf{W}^{-1}, \quad \forall k = 1, \dots, K. \quad (2)$$

where $\mathbf{D}^{(k)}$ is a diagonal matrix containing the eigenvalues of $\mathbf{M}^{(k)}$. It has been shown in [16], through numerical simulations, that CPD algorithms based on JEVD have several advantages. First, they need few iterations to converge. Second, they are relatively insensitive to correlated factors and to an over-estimation of the CPD rank (overfactoring problem). On the other hand, JEVD-based CPD methods may require more re-

strictive conditions (see [16, section 3.5] for a comparative analysis).

A naive approach to solve the JEVD problem consists in computing the eigenvalue decomposition of only one matrix, for instance $\mathbf{M}^{(k)}$ or the product of several $\mathbf{M}^{(k)}$ matrices. However, this approach is very sensitive to the noise and has no unique solution in the presence of a degenerate eigenvalue (i.e. multiplicity strictly greater than one). Indeed, JEVD has less restrictive uniqueness conditions [18]. Note that by uniqueness we mean essential uniqueness, that is to say the matrix \mathbf{W} of eigenvectors can be estimated only up to a diagonal and a permutation matrix. Consequently, solving the JEVD problem by means of a single EVD is not a good thing. Generally, the JEVD problem is solved by minimizing a diagonality criterion applied simultaneously to multiplicative updates of the $\mathbf{M}^{(k)}$ matrices. The main differences between the existing JEVD algorithms reside in the parametrization of the updating matrix and the implemented strategy to compute these parameters.

When the eigenvector matrix \mathbf{W} is orthogonal, the JEVD problem becomes an Orthogonal Joint Diagonalization (OJD) problem. The latter is a well-know JD problem for which many algorithms have been proposed. One of the most famous algorithm is JAD (Jacobi Angles for simultaneous Diagonalization) [19]. In [20] and [21], authors focused on the approximate OJD of a set of positive definite matrices by minimizing an original cost function based on the log-likelihood approach. These algorithms are particularly efficient and have good convergence properties. Indeed, the quadratic local convergence of Pham's method is theoretically proven while simulations show that a few iterations are needed to ensure convergence [21]. In order to take benefit of these attractive properties, we introduce a novel two-step JEVD approach transforming the original matrix set into a set of symmetric positive definite matrices. For this purpose, we introduce the problem of joint symmetrization and design an *ad-hoc* algorithm. We show the good behavior of the proposed JEVD method when it is combined with DIAG in order to canonically decompose data measured from fluorescence spectroscopy. In addition, we show the ability of our method to face difficult cases encountered in fluorescence measurements.

The paper is organized as follows. We first recall how the CPD problem can be rewritten into a JEVD problem. Then we describe, in section 3, the proposed approach to solve the JEVD problem. Section 4 presents a comparative computational analysis of i) the proposed JEVD method and ii) the DIAG method using this JEVD solution. The last part of the paper is dedicated to a chemometric application. Thus, in section 5, we first recall the CPD model of three-way fluorescence measurements. Then, we investigate the behavior of the proposed approach to decompose the observed data tensors in comparison with some other existing algorithms.

Notations. In the following, scalars are denoted by a lower case (x), vectors by a boldface lower case (\mathbf{x}), matrices by a boldface upper case (\mathbf{X}) and tensors by an upper case boldface calligraphic (\mathcal{X}). x_i is the i -th element of vector \mathbf{x} , $X_{i,j}$ is the (i, j) -th component of matrix \mathbf{X} and $X_{i,j,k}$ is the (i, j, k) -element of third order tensor \mathcal{X} . Vectors $\mathbf{x}_{i,:}$ and $\mathbf{x}_{:,j}$ are the i^{th} row and j^{th} col-

umn of matrix \mathbf{X} respectively. Matrix $\mathbf{X}_{\dots,r}$ is a r^{th} frontal slice of tensor \mathcal{X} . Operator $[\mathbf{X}]_+$ projects the entries of \mathbf{X} on \mathbb{R}^+ by forcing all negative entries of \mathbf{X} to 0. $\text{Diag}\{\mathbf{X}\}$ is an operator that set to zero the off-diagonal entries of matrix \mathbf{X} whereas $\text{diag}\{\mathbf{x}\}$ creates a diagonal matrix from the entries of vector \mathbf{x} . The euclidean norm is denoted by $\|\cdot\|_2$. \odot is the Khatri-Rao product, defined as a column-wise Kronecker product. Eventually, $\{\mathbf{X}^{(k)}\}$ denotes the set of K matrices indexed by k : $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(K)}\}$.

2. From CPD to JEVD

There are several ways to compute CPD by means of JEVD. The first step consists in compressing the considered tensor. Different approaches can be used to do that. Hence, in the SECSI framework [15], JEVD arises from the Higher-Order Singular Value Decomposition (HOSVD) of the tensor while in the DIAG [16] and SSD-CP [17] algorithms, it comes from the Singular Value Decomposition (SVD) of one unfolding matrix. Hereafter, we describe the latter approach as example but note that the proposed JEVD algorithm can be applied to both approaches.

As mentioned in introduction, a fluorescence third order tensor is considered in this paper, for which the rank- N CPD is given by equation (1). By juxtaposing the R frontal slices, the tensor \mathcal{T} can be reshaped into the unfolding matrix \mathbf{T} of size $P \times QR$ where the second and third modes of \mathcal{T} are merged. The CPD is then rewritten in matrix form as:

$$\mathbf{T} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T. \quad (3)$$

Two other unfolding matrices are available. As explained in [16], it is crucial to use the squarest unfolding matrix or at least the unfolding matrix which is rank- N . Without loss of generality, we consider in this section that it is given by (3).

In order to transform the CPD problem into a JEVD problem, a rank- N truncated SVD of \mathbf{T} is computed. The three matrices of this truncated SVD are denoted by \mathbf{U} , $\mathbf{\Sigma}$ and \mathbf{V}^T , respectively. Then there exists a non-singular square matrix \mathbf{W} of size $N \times N$ such that

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{W} \text{ and } \mathbf{C} \odot \mathbf{B} = \mathbf{V}\mathbf{W}^{-T}. \quad (4)$$

Furthermore $(\mathbf{C} \odot \mathbf{B})^T$ can be seen as a horizontal block matrix:

$$(\mathbf{C} \odot \mathbf{B})^T = [\Phi^{(1)}(\mathbf{B})^T, \dots, \Phi^{(R)}(\mathbf{B})^T], \quad (5)$$

where $\Phi^{(1)}, \dots, \Phi^{(R)}$ are the R diagonal matrices built from the R rows of matrix \mathbf{C} . Then, (4) and (5) yield

$$\mathbf{V}^T = [\Gamma^{(1)T}, \dots, \Gamma^{(R)T}], \quad (6)$$

where $\Gamma^{(r_1)} = \mathbf{B}\Phi^{(r_1)}\mathbf{W}^T$ for $r_1 = 1, \dots, R$. Assuming that matrix \mathbf{B} is full column rank, we can define for all couples $k = (r_1, r_2)$ with $r_1 = 1, \dots, R$ and $r_2 = 1, \dots, R$, $r_1 \neq r_2$, a set of matrices jointly diagonalizable by similarity transformation :

$$\mathbf{M}^{(k)} = (\Gamma^{(r_1)} \# \Gamma^{(r_2)})^T, \quad (7)$$

$$= \mathbf{W}\mathbf{D}^{(k)}\mathbf{W}^{-1}, \quad (8)$$

where $\#$ is the Moore-Penrose inverse and $\mathbf{D}^{(k)} = \Phi^{(r_2)} \Phi^{(r_1)\#}$ so that $\{\mathbf{D}^{(k)}\}$ is a set of diagonal matrices. Hence, performing JEVD of the $K = R(R-1)$ matrices $\{\mathbf{M}^{(k)}\}$ allows to estimate \mathbf{W} . This quadratic relationship between the number of matrices in the JEVD step (K) and the dimension of the last mode (R) has two consequences when R is large. On one hand, it achieves a better estimation of \mathbf{W} . On the other hand, the numerical complexity quadratically increases with R (see section 4). Thus one has to find a good trade-off between speed and estimation accuracy. As a rule of thumb, if K is very large with respect to the matrix size N , one can reasonably leave out a significant fraction of the matrix set. Here, we apply an arbitrary matrix decimation as a naive approach but better strategies could be used. For instance, we could compute and use a basis of the vector subset spanned by the K matrices to be diagonalized.

At this stage, there are several ways to deduce the factor matrices from matrix \mathbf{W} and/or the set $\{\mathbf{D}^{(k)}\}$. DIAG resorts to the eigenvector matrix \mathbf{W} and equations (4) to immediately estimate matrices \mathbf{A} and $\mathbf{C} \odot \mathbf{B}$. Then \mathbf{B} and \mathbf{C} are deduced as explained in [16]. Conversely to the DIAG procedure, SSD-CP resorts to the eigenvalues contained in matrices $\{\mathbf{D}^{(k)}\}$. Indeed diagonal entries of these matrices are linearly linked to the entries of \mathbf{C} so that a least square estimate of \mathbf{C} can be obtained from an overdetermined linear system. $\mathbf{B} \odot \mathbf{A}$ is then easily deduced, hence \mathbf{B} and \mathbf{A} . Of course these different schemes could be combined in order to define hybrid versions of DIAG and SSD-CP but this is outside the scope of this paper.

3. JEVD based on joint symmetrization

3.1. A two-step framework

We consider a set of K matrices of size $N \times N$:

$$\mathbf{M}^{(k)} = \mathbf{W} \mathbf{D}^{(k)} \mathbf{W}^{-1} \quad \forall k. \quad (9)$$

Any non-singular matrix can be decomposed as a product of a lower triangular matrix and an orthogonal matrix. Let \mathbf{L} and \mathbf{Q} be the matrices of the \mathbf{QL} decomposition of \mathbf{W}^{-1} : \mathbf{L} is a lower triangular matrix, \mathbf{Q} is an orthogonal matrix and $\mathbf{W}^{-1} = \mathbf{QL}$. Thus the set of equations (9) can be rewritten as

$$\mathbf{M}^{(k)} = \mathbf{L}^{-1} \mathbf{Q}^T \mathbf{D}^{(k)} \mathbf{QL}, \quad \forall k. \quad (10)$$

Then, defining matrices $\mathbf{S}^{(k)}$ as

$$\mathbf{S}^{(k)} = \mathbf{LM}^{(k)}\mathbf{L}^{-1}, \quad \forall k, \quad (11)$$

leads to a JEVD with orthogonal transformation:

$$\mathbf{S}^{(k)} = \mathbf{Q}^T \mathbf{D}^{(k)} \mathbf{Q}, \quad \forall k. \quad (12)$$

As a consequence, in the noise free case, all matrices $\mathbf{S}^{(k)}$ are symmetric and equation (12) defines the OJD of $\{\mathbf{S}^{(k)}\}$. We thus propose a two step algorithm. First, we look for a lower triangular matrix $\widehat{\mathbf{L}}$ that jointly "symmetrizes" $\{\mathbf{M}^{(k)}\}$ in order to estimate the set $\{\mathbf{S}^{(k)}\}$ or an approximation of it in the noisy case. After this Joint Symmetrization (JS) step, JEVD problem

is transformed into an OJD problem. An estimate of \mathbf{Q} , denoted $\widehat{\mathbf{Q}}$, is then obtained by solving this OJD problem (using, for instance, the JAD algorithm). We call this approach, JDJS for Joint Diagonalization based on Joint Symmetrization. Nevertheless, we aim at using the algorithm introduced by Pham in [21]. Our motivation is to enjoy the nice convergence properties of this algorithm. Indeed, Pham has proven its convergence when the matrices of the set can be nearly jointly diagonalized. He has also provided a theoretical investigation showing the quadratic convergence of the algorithm close to the diagonalizing solution. Eventually, he has shown, through numerical simulations, that it requires very few iterations to reach the convergence. However, Pham's algorithm is designed for positive definite matrices. This is due to the used diagonality criterion arising from the Hadamard's inequality [22]. Let $\{\mathbf{S}_2^{(k)}\}$ be a set of positive definite matrices, the Hadamard's inequality leads then to

$$\det\{\text{Diag}\{\widehat{\mathbf{Q}} \mathbf{S}_2^{(k)} \widehat{\mathbf{Q}}^T\}\} \geq \det\{\widehat{\mathbf{Q}} \mathbf{S}_2^{(k)} \widehat{\mathbf{Q}}^T\}. \quad (13)$$

The equality being reached if and only if the matrix $\widehat{\mathbf{Q}} \mathbf{S}_2^{(k)} \widehat{\mathbf{Q}}^T$ is diagonal, a natural diagonality criterion based on the likelihood maximum method directly comes as

$$\frac{1}{2} \sum_{k=1}^K \log(\det\{\text{Diag}\{\widehat{\mathbf{Q}} \mathbf{S}_2^{(k)} \widehat{\mathbf{Q}}^T\}\}) - \log(\det\{\widehat{\mathbf{Q}} \mathbf{S}_2^{(k)} \widehat{\mathbf{Q}}^T\}). \quad (14)$$

In our approach, we can build the $\{\mathbf{S}_2^{(k)}\}$ matrix set directly after the JS step as:

$$\mathbf{S}_2^{(k)} = (\mathbf{S}^{(k)})^T \mathbf{S}^{(k)} \quad \forall k. \quad (15)$$

Hence, we obtain positive definite matrices jointly diagonalizable in a common orthogonal basis having the following expression:

$$\mathbf{S}_2^{(k)} = \mathbf{Q}^T (\mathbf{D}^{(k)})^2 \mathbf{Q}, \quad \forall k. \quad (16)$$

This second version of JDJS is called JDJS².

It is noteworthy that the matrix \mathbf{Q} is unique (up to permutation and scaling indeterminacy) in equation (12) if and only if the rows of the matrix

$$\mathbf{\Omega} = \begin{pmatrix} D_{11}^{(1)} & \cdots & D_{11}^{(K)} \\ \vdots & \ddots & \vdots \\ D_{NN}^{(1)} & \cdots & D_{NN}^{(K)} \end{pmatrix} \quad (17)$$

are distinct [23]. Hence, the transformation in equation (16) should not impact the identifiability of the matrix \mathbf{Q} excepted if and only if one row of the matrix $\mathbf{\Omega}$ is the opposite of another.

In the next subsection we focus on the JS step which is at the center of the proposed approach.

3.2. Joint symmetrization

We call joint symmetrization any process that aims to jointly transform a set of square matrices into a set of symmetric matrices and we call symmetrized matrix any output of a symmetrization process.

The proposed method to build $\{\mathbf{S}^{(k)}\}$ from $\{\mathbf{M}^{(k)}\}$ is a block coordinate method based on multiplicative updates. Block coordinate methods consist in estimating several unknown parameter subsets instead of estimating the whole of them at each optimization step. Here, it means that the matrix $\widehat{\mathbf{L}}$ is decomposed as the product of elementary matrices. It is noteworthy that contrarily to JD problems, the JS problem is not invariant to diagonal matrix transformations. That is why, we have to resort to two kinds of elementary matrices:

- Real elementary lower triangular matrices, denoted $\mathbf{L}^{(ij)}(x)$, ($i > j$), are equal to the identity matrix at the exception of the $L_{i,j}^{(ij)}$ term which is equal to $x \in \mathbb{R}$.
- Real elementary diagonal matrices, denoted $\mathbf{\Lambda}^{(j)}(a)$, are equal to the identity matrix at the exception of the $D_{j,j}^{(j)}$ term which is equal to $a \in \mathbb{R}$.

The following proposition gives then a simple factorization of any lower triangular matrix :

Proposition 1. *Let \mathbf{L} be a lower triangular matrix of size N , then there exists $N(N-1)/2$ elementary lower triangular matrices $\{\mathbf{L}^{(ij)}(x^{(ij)})\}$ and N elementary diagonal matrices $\{\mathbf{\Lambda}^{(j)}(a^{(j)})\}$ such that*

$$\mathbf{L} = \left[\prod_{j=1}^{N-1} \left[\prod_{i=j+1}^N \mathbf{L}^{(ij)}(x^{(ij)}) \right] \mathbf{\Lambda}^{(j)}(a^{(j)}) \right] \mathbf{\Lambda}^{(N)}(a^{(N)}). \quad (18)$$

A constructive proof is trivial.

Thus, such a factorization gives us a natural choice for the construction of $\widehat{\mathbf{L}}$ by multiplicative updates. Computation of each elementary matrix will be done by successive optimization of a symmetrization criterion with respect of only one parameter. We call a multiplicative update of the set $\{\mathbf{M}^{(k)}\}$ by matrix \mathbf{X} the whole following **similarity** transformations:

$$\mathbf{M}^{(k)} \leftarrow \mathbf{X} \mathbf{M}^{(k)} \mathbf{X}^{-1} \quad \forall k. \quad (19)$$

In order to compute the matrix \mathbf{X} , we propose to minimize the following symmetrization criterion:

$$\begin{cases} C(\mathbf{X}) = \sum_{k=1}^K \sum_{p=1}^{N-1} \sum_{q=p+1}^N (N_{p,q}^{(k)} - N_{q,p}^{(k)})^2 \\ \text{where } \mathbf{N}^{(k)} = \mathbf{X} \mathbf{M}^{(k)} \mathbf{X}^{-1}. \end{cases} \quad (20)$$

This criterion is non-negative and is equal to zero if and only if all matrices $\mathbf{N}^{(k)}$ are symmetric. According to proposition 1 and equation (11), the joint symmetrization of $\{\mathbf{M}^{(k)}\}$ can be done thanks to a set of elementary lower triangular matrices denoted $\{\mathbf{L}^{(ij)}(x^{(ij)})\}$ and a set of elementary diagonal matrices denoted $\{\mathbf{\Lambda}^{(j)}(a^{(j)})\}$. More precisely, at each couple (i, j) , $j = 1, \dots, N-1, i = j+1, \dots, N$ corresponds a multiplicative update by $\mathbf{L}^{(ij)}(x^{(ij)})$. In addition, when $i = N$ the update by $\mathbf{L}^{(Nj)}(x^{(Nj)})$ is followed by an update by $\mathbf{\Lambda}^{(j)}$. Finally the last update is done by $\mathbf{\Lambda}^{(N)}(a^{(N)})$.

We now describe how the elementary triangular and diagonal updating matrices are computed. The goal of an update is to

make the whole set $\{\mathbf{M}^{(k)}\}$ closer to a set of symmetric matrices. In this purpose, each updating matrix is chosen by minimizing the symmetrization criterion (20). We first consider the update by elementary lower triangular matrix $\mathbf{L}^{(ij)}(x)$ i.e. $\mathbf{X} = \mathbf{L}^{(ij)}(x)$ in (20). Thus, we have

$$\mathbf{N}^{(k)} = \mathbf{L}^{(ij)}(x) \mathbf{M}^{(k)} (\mathbf{L}^{(ij)}(x))^{-1}. \quad (21)$$

We can easily check that this update only impacts the j^{th} row and the i^{th} column of $\mathbf{M}^{(k)}$ so that

$$C(\mathbf{L}^{(ij)}(x)) = \sum_{k=1}^K \left(\sum_{\substack{p \neq i \\ p \neq j}}^K ((M_{p,i}^{(k)} - N_{i,p}^{(k)})^2 + (N_{p,j}^{(k)} - M_{j,p}^{(k)})^2) + (N_{i,j}^{(k)} - M_{j,i}^{(k)})^2 \right) + \beta \quad (22)$$

where β is a constant independent of x and

$$\begin{cases} \forall p \neq j, N_{i,p}^{(k)} = -x M_{j,p}^{(k)} + M_{i,p}^{(k)}, \\ \forall p \neq i, N_{p,j}^{(k)} = x M_{p,i}^{(k)} + M_{p,j}^{(k)}, \\ N_{i,j}^{(k)} = -x^2 M_{j,i}^{(k)} + x(M_{i,i}^{(k)} - M_{j,j}^{(k)}) + M_{i,j}^{(k)}. \end{cases} \quad (23)$$

Thus, we have

$$C(\mathbf{L}^{(ij)}(x)) = P(x) = \alpha_4 x^4 + \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \beta' \quad (24)$$

with

$$\alpha_4 = \sum_{k=1}^K (M_{j,i}^{(k)})^2; \quad \alpha_3 = -2 \sum_{k=1}^K M_{j,i}^{(k)} (M_{i,i}^{(k)} - M_{j,j}^{(k)}); \quad (25)$$

$$\alpha_2 = \sum_{k=1}^K \left(-2 M_{j,i}^{(k)} (M_{i,j}^{(k)} - M_{j,i}^{(k)}) + (M_{i,i}^{(k)} - M_{j,j}^{(k)})^2 + \sum_{\substack{p \neq i \\ p \neq j}}^K ((M_{j,p}^{(k)})^2 + (M_{p,i}^{(k)})^2) \right); \quad (26)$$

$$\alpha_1 = 2 \sum_{k=1}^K \left((M_{i,i}^{(k)} - M_{j,j}^{(k)}) (M_{i,j}^{(k)} - M_{j,i}^{(k)}) + \sum_{\substack{p \neq i \\ p \neq j}}^K (M_{p,i}^{(k)} (M_{p,j}^{(k)} - M_{j,p}^{(k)}) - M_{j,p}^{(k)} (M_{i,p}^{(k)} - M_{p,i}^{(k)})) \right). \quad (27)$$

β' is a constant term thereby its expression is useless here. $P(x)$ is a coercive 4 degree polynomial, as a consequence we choose $x^{(ij)}$ as the real root of the derivative of $P(x)$ that minimize $P(x)$.

We now explain how to compute matrix $\mathbf{\Lambda}^{(j)}(a^{(j)})$ for a given j . We follow the same idea as previously. The update by $\mathbf{\Lambda}^{(j)}(a)$ (i.e. taking $\mathbf{X} = \mathbf{\Lambda}^{(j)}(a)$ in equation (20)) gives

$$\mathbf{N}^{(k)} = \mathbf{\Lambda}^{(j)}(a) \mathbf{M}^{(k)} (\mathbf{\Lambda}^{(j)}(a))^{-1}. \quad (28)$$

We then easily show that

$$C(\Lambda^{(j)}(a)) = \sum_{k=1}^K \sum_{p \neq j} (N_{j,p}^{(k)} - N_{p,j}^{(k)})^2 + \beta'' \quad (29)$$

$$= \sum_{k=1}^K \sum_{p \neq j} \left(\frac{M_{j,p}^{(k)}}{a} - a M_{p,j}^{(k)} \right)^2 + \beta'' \quad (30)$$

$$= f(a) + \beta'' \quad (31)_{225}$$

where β'' is a constant independent of a . This yields

$$\forall a \neq 0, f'(a) = 0 \Leftrightarrow a^4 = \frac{\sum_{k=1}^K \sum_{p \neq j} (M_{j,p}^{(k)})^2}{\sum_{k=1}^K \sum_{p \neq j} (M_{p,j}^{(k)})^2}. \quad (32)_{230}$$

$f(a)$ being an even function, we can take

$$a^{(j)} = \left(\frac{\sum_{k=1}^K \sum_{p \neq j} (M_{j,p}^{(k)})^2}{\sum_{k=1}^K \sum_{p \neq j} (M_{p,j}^{(k)})^2} \right)^{\frac{1}{4}}. \quad (33)_{235}$$

Performing the $N(N+1)/2$ successive updates using the whole set of (i, j) couples is called a sweep. Of course, this optimization scheme is not equivalent to simultaneously estimate all the entries of \mathbf{L} . As a consequence, we do not expect that at the end of the first sweep the matrix set $\{\mathbf{M}^{(k)}\}$ is symmetrized. That is why, the sweeping procedure has to be repeated such as one iteration of the symmetrization algorithm corresponds to a sweep. The algorithm is stopped when a suitable stopping criterion reaches a defined threshold or when a maximum number of iterations is reached. A classical stopping criterion is given by the relative decrease of C between two successive iterations.

It is worth mentioning that (i.) The identity matrix belongs to the set of elementary lower triangular matrices and to the set of elementary diagonal matrices. This ensures that criterion C cannot increase after an update and thus after a sweep. (ii.) Elementary lower triangular matrices are invertible by construction, meaning that update (21) is always possible. (iii.) During the symmetrization process the numerator and the denominator in equation (33) tend to the same value. If the numerator and/or the denominator is equal to 0, one can impose $a^{(j)} = 1$ and let the algorithm run.

4. Computational complexity analysis

Although the ultimate goal of comparing JEVD or CPD techniques is to evaluate the quality of methods as reflected by the computed matrices, it is also interesting to assess the numerical complexity of these methods. Complexity is defined here as the number of floating point operations required to execute an algorithm (flops). A flop corresponds to a multiplication followed by an addition. But, in practice, only the number of multiplications is considered since, most of the time, there are about as many (and slightly more) multiplications as additions. In order to simplify the expressions, the complexity is generally approximated by its asymptotic limit, as the size of the problem tends to infinity. We shall subsequently denote, with some

small abuse of notation, the equivalence between two strictly positive functions f and g :

$$f(x) = O(g(x)) \quad \text{or} \quad g(x) = O(f(x)) \quad (34)$$

if and only if the ratio $f(x)/g(x)$ tends to 1 as $x \rightarrow +\infty$. In practice, knowing whether an algorithm is computationally heavy is as important as knowing its performance in terms of SNR.

This section first addresses the complexity of some JEVD methods needed by the DIAG algorithm to solve the CPD problem. Next the numerical complexity of DIAG is given. In order to have a comparative reference, we provide the complexity of the alternating FNNLS (A-FNNLS) technique proposed in [24] and implemented in the N -way toolbox [25, 26], which is used in section 5.

4.1. Complexity of JEVD methods

The costliest operation of JEVD algorithms is the multiplicative update step of the matrix set defined in equation (19). Now the numerical complexity of this step strongly depends of the structure of the updating matrix.

Regarding the JSJD² method, one sweep of the JS step involves $N(N-1)/2$ elementary lower triangular matrices and N elementary diagonal matrices as updating matrices. The numerical complexity of the JS step is then $O(KN^3)$. In a similar way, the JD step based on Pham's algorithm involves Givens matrices. Therefore its numerical complexity is also $O(KN^3)$. Consequently, our JEVD method costs $O(KN^3)$ flops.

Note that the numerical complexity of classical JEVD methods such as JUST [27], SH-RT [28], JDTM [16], JAPAM [29], SJDTE [30] and JET-U [31] is $O(KN^3)$ too.

4.2. Complexity of two CPD methods: DIAG and A-FNNLS

The computational complexity of DIAG is clearly dominated by the three following computations. First, the truncated SVD of the unfolding matrix of size $P \times QR$ requires $2QRP^2 + 5N^2(P + QR) - 2(N^3 + P^3)/3$ multiplications by assuming that $QR > P$. Then, the computation of the $\mathbf{M}^{(k)}$ matrices needs approximately $(NR)^2Q$ additional multiplications. Finally the cost of the JEVD procedure is $O(R^2N^3)$ according to section 4.1 since $K = R(R-1)$. Additional computations can be neglected. Now, let's consider a low-rank CPD, i.e. $N \ll \min(P, Q, R)$, as observed in most of the applications such as the fluorescence one. Thus, the numerical complexity of DIAG is $O(P^2(P^2 + PN^2 + N^3))$.

As far as the A-FNNLS algorithm is concerned, the estimation of the factors is performed by solving alternatively Non-Negative Least Squares (NNLS) problems. The first widely used algorithm for solving the NNLS problem is an active-set method published in [32]. A modification of the latter was proposed in [24] giving rise to the FNNLS (Fast NNLS) method. The effective benefit of FNNLS is from forming the normal equations and recognizing that the normal equations on the passive set are simply a subset of these. The computational advantages, then, are only applicable for overdetermined systems, which is generally the case for the CPD problems encountered in applications. Without utilizing the normal equations, the original NNLS algorithm has $O(MN^2)$ complexity for

each iteration where M denotes the second dimension of the unfolding matrix used to estimate some factors (e.g. QR for the unfolding matrix (3)). FNNLS, however, only has an initialization complexity of $O(MN^2)$, followed by $O(N^3)$ complexity in each iteration. One may recognize that the use of the normal equations comes at the price of numerical stability. This approach, therefore, may not be appropriate when presented with an ill-conditioned problem [33]. Under the same assumptions as for DIAG, the numerical complexity of A-FNNLS is $O(P(P^2N^2 + N^3))$.

This asymptotic analysis shows that when the tensor dimensions of the low-rank CPD tend to infinity, the DIAG method requires more flops than A-FNNLS due to the truncated SVD ($O(P^4)$ flops) computed at the beginning of DIAG. Nevertheless, for tensors of smaller size such as that considered in the following section, the truncated SVD is rarely the most costly step of DIAG. Indeed, for such scenarios, by considering cubic tensors, the term $2P^2N^3(I_{SD} + 2I_{JD})$ is generally dominant in the complexity of DIAG, where I_{SD} and I_{JD} denote the numbers of iterations of the two steps of JSJD². As far as A-FNNLS is concerned, its dominant term is $I_AP^3N^2$, where I_A denotes the number of iterations (all factors have been estimated once using the FNNLS method during one iteration). Since the number I_A of iterations is generally much larger than $I_{SD} + 2I_{JD}$ and since $P > N$, the term $I_AP^3N^2$ appears to be larger than $2P^2N^3(I_{SD} + 2I_{JD})$ in such a context. This will be confirmed in the following section by means of the computed CPU time: DIAG will appear to be less time-consuming than A-FNNLS for the considered fluorescence spectroscopy scenarios.

5. Application to the Canonical Polyadic Decomposition of Fluorescence spectra

5.1. CPD of fluorescence excitation emission matrices

In this section, we show how the proposed approach can be applied in the context of fluorescence spectroscopy.

Fluorescence is defined as the emission of light (fluorescence or emitted light) by a substance (molecule or atom) that has previously absorbed another light (excitation light). A fluorescing chemical component is called a fluorophore. The fluorescence intensity of a diluted solution of several fluorophores varies with the wavelength of the excitation light, the wavelength of the fluorescence light and the fluorophore concentrations. Spectrofluorimeters allow to measure the fluorescence intensity of a diluted solution at various excitation-emission wavelengths couples, providing a data matrix that characterizes the solution. Such a matrix is called Fluorescent Excitation-Emission Matrix (FEEM) or fluorescence 3D spectra.

In practice, chemists often have at their disposal a set of fluorescent solutions from which they measure a set of FEEM. We denote \mathbf{X}_r the FEEM characterizing the r^{th} chemical solution. Each FEEM can be seen as a frontal slice of a third order tensor. Therefore, measured fluorescence intensities can be stored in a tensor \mathcal{T} so that $\mathcal{T}_{p,q,r}$ is the fluorescence intensity of solution r at the excitation wavelength p and emission wavelength q . Each solution can be seen as a mixture of N fluorophores.

Then, assuming that non-linear effects such as Rayleigh and Raman scattering or inner filter effects have been corrected, it has been showed in [34] that

$$\mathcal{T}_{p,q,r} \simeq \sum_{n=1}^N A_{p,n} B_{q,n} C_{r,n} \quad (35)$$

where

- $A_{p,n}$ is the capacity of the fluorophore n to absorb light at excitation wavelength number p . Column n of \mathbf{A} defines the excitation spectra of the fluorophore n .
- $B_{q,n}$ is the capacity of the fluorophore n to emit light at emission wavelength number q . Column n of \mathbf{B} defines the emission spectra of the fluorophore n .
- $C_{r,n}$ is proportional to the concentration of the fluorophore n in solution r . Column n of \mathbf{C} defines the concentration profile of the fluorophore n .

Here, we also impose that

$$\|\mathbf{a}_{:,n}\|_2 = \|\mathbf{b}_{:,n}\|_2 = 1, \quad \forall n. \quad (36)$$

As a consequence, $c_{:,n}$ represents the contributions of fluorophore n to the different mixtures. We then define

$$C_n = \|\mathbf{c}_{:,n}\|_2, \quad \forall n \quad (37)$$

as the contribution of fluorophore n to the fluorescence tensor \mathcal{T} . In order to gather all the spectral information about fluorophore n in an unique spectral signature, we define its individual 3D spectrum as the rank one matrix $\mathbf{a}_{:,n} \mathbf{b}_{:,n}^T$. Finally, the CPD model considers that each FEEM \mathbf{X}_r is modeled as

$$\mathbf{X}_r = \mathbf{A} \text{diag}\{\mathbf{c}_{:,r}\} \mathbf{B}^T, \quad \forall r. \quad (38)$$

5.2. Simulations on synthetic data sets

We first propose to evaluate our approach by using several synthetic data sets designed to cover some of the main issues encountered with real unknown fluorescence data:

1. some fluorophores can have similar excitation or emission spectra,
2. the actual number of fluorophores is unknown and it is overestimated (overfactoring).

5.2.1. Data sets and algorithm settings

Our data sets correspond here to different synthetic fluorescence tensors of size $(P \times Q \times R)$. These tensors are built from a noisy CPD model of rank 4. Hence, each tensor gathers simulated fluorescence measurements of R random mixtures of 4 synthetic fluorophores and each data set contains 100 realizations of these synthetic fluorescence tensors for each considered noise level. Synthetic data sets are built as follows.

We first build four synthetic excitation and emission spectra defining matrices \mathbf{A} and \mathbf{B} in equation (38). For all data sets, excitation and emission spectra are of size $P = 41$ and

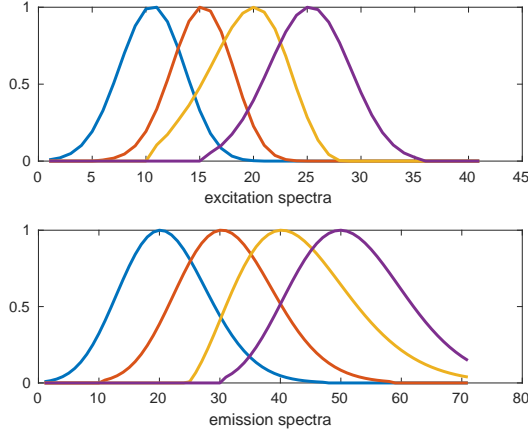


Figure 1: Synthetic excitation and emission spectra used to build data set 1.

$Q = 71$ respectively and are norm 1. In the proposed numerical simulations, excitation and emission spectra are unimodal spectra similar to those presented in figure 1. Entries of \mathbf{C} are randomly drawn from a uniform distribution. Columns of \mathbf{C} are also normalized so that all fluorophores have unit contributions. Then, R synthetic noise-free FEEM are built one after another according to equation (38). In order to simulate model errors due to the FEEM acquisition and signal processing, we build from each noise-free FEEM the corresponding slice of the noisy tensor $\mathbf{T}_{\dots r}$ as follows:

$$\mathbf{T}_{\dots r} = \mathcal{A}\{\mathbf{G}_r \boxtimes \tilde{\mathbf{X}}_r\} \quad (39)$$

where $\tilde{\mathbf{X}}_r$ and \mathbf{G}_r are two random matrices of size $P \times Q$ that simulate the photon detection counting and the possible variations of the photomultiplier gain respectively. \boxtimes is the Hadamard product and \mathcal{A} is a smoothing operator. More precisely:

- Entry (p, q) of $\tilde{\mathbf{X}}_r$ is randomly drawn from a Poisson distribution of mean $(X_r)_{p,q}$: $\forall (p, q), (\tilde{X}_r)_{p,q} \sim \text{Pois}((X_r)_{p,q})$. It is noteworthy that before applying Poisson noise, the matrix \mathbf{X}_r is normalized in order to set its lowest value to one.
- Entries of \mathbf{G}_r are randomly drawn from a uniform distribution in the interval $[1 - \alpha; 1 + \alpha]$: $\forall (p, q), (\tilde{X}_r)_{p,q} \sim \text{unif}(1 - \alpha, 1 + \alpha)$. Hence, parameter α allow us to vary the amplitude of the multiplicative noise.
- Here \mathcal{A} is a Savitzky–Golay filter. This filter is commonly used by spectrofluorometer manufacturer softwares.

Other non linear deviations such as diffusion scattering or inner filter effects are not considered here. This process is repeated 100 times for six different values of the noise amplitude α from 0.05 to 0.5. Note that $\alpha = 0.5$ means that the photomultiplier gain varies from 50% to 150% around its mean value (giving a relative standard deviation of 29%). Thus each data set consists in six subsets of 100 noisy tensors. As it will be explained in the dedicated subsections below, data sets will differ in the number

of mixtures (R) and in the correlation coefficients between the factors of the two spectral modes.

Now regarding the algorithms, we compare three different approaches. Each approach is represented by two algorithms. The first approach resorts to the DIAG algorithm (introduced in section 2) to compute the CPD. Two implementations are considered. The first implementation, referred to as DIAG-JDJS², corresponds to the proposed JDJS² algorithm: after the symmetrization step, the symmetry of the matrices to diagonalize is imposed by construction and the joint diagonalization step is performed thanks to Pham algorithm. Indeed, thanks to preliminary comparisons (results not shown), we obtained best results with this configuration. The second implementation corresponds to a reference JEVD algorithm: the JEVD step is computed with the SJDTE algorithm recently introduced in [30]. This algorithm belongs to the family of JDTE algorithm. Its main advantage in the present context of small real matrices is its rapidity. This implementation is referred to as DIAG-SJDTE. Both JEVD algorithms are initialized with the identity matrix. The JEVD step is stopped when a maximum number of iterations is reached or when the relative deviation between two successive values of the algorithms criterion falls under a defined threshold. In JDJS², JS and JD steps are stopped when the relative deviation between two successive values of the criterion is lower than 10^{-6} . Maximum number of iterations are set to 100 for both steps. The same stopping criterion is used for SJDTE but the maximum number of iterations is set to 50. Here, we have the *prior* knowledge that entries of the factor matrices are nonnegative. Therefore, after the JEVD step in the DIAG procedure, we take $\mathbf{A} = [\mathbf{U}\mathbf{S}\mathbf{W}]_+$ and $\mathbf{C} \odot \mathbf{B} = [\mathbf{V}\mathbf{W}^T]_+$.

The second approach resorts to alternating CPD algorithms with non-negative constraints. The first algorithm of this approach is the A-FNNLS algorithm mentioned in section 4 of the N-way toolbox¹ (parafac function) which is commonly used to analyze fluorescent excitation-emission matrices. The second algorithm is the recent AO-ADMM algorithm [35]. This algorithm has proved to be very efficient to deal with overfactoring and correlated factors. We used the AO-ADMM implementation of Jeremy Cohen² [36] (AOadmm function). A-FNNLS is initialized using the best-fitting model scheme implemented in the parafac function: algorithm is run from 5 different random starting values and keeps the starting values that has led to the best fit after few iterations. We keep the default values of the convergence criterion (10^{-6}) and the maximal iteration number (2500). The starting values selected by the best-fitting model scheme are reused as starting values for the AO-ADMM algorithm. We keep the default settings (least-square loss, non-negative constraint, tolerance of 10^{-7}) of the AOadmm function but the maximal number of iterations is set to 1000.

The last approach consists in using the factors estimated by the algorithms of the first approach to initialize an algorithm of the second approach and then perform a very limited number of iterations as proposed in [16]. The main advantage of this

¹<http://www.models.life.ku.dk/nwaytoolbox>

²<http://www.gipsa-lab.grenoble-inp.fr/pierre.comon/TensorPackage/tensorPackage.htm>

hybrid approach is to improve the results obtained by the DIAG algorithm while keeping a limited computational cost. Here, we use the factors estimated by DIAG-JDJS² and DIAG-SJDTE⁴⁶⁵ as starting values and we perform a maximum of 10 iterations of AO-ADMM. These algorithms are referred to as JDJS²-AO-ADMM and SJDTE-AO-ADMM respectively.

In order to evaluate the robustness of each algorithm with respect to overfactoring, we compute CPD of rank 6 instead⁴⁷⁰ of 4. Then, the 6 computed factors are sorted according to their contributions (fluorophores contribution are defined in equation (37)). We keep the 4 factors with the highest contributions as the estimation of the actual factors. Scaling and permutation indeterminacy are then removed as explained in [37]. We denote⁴⁷⁵ the estimated factors linked to fluorophore n by $\hat{\mathbf{a}}_{:,n}$, $\hat{\mathbf{b}}_{:,n}$, $\hat{\mathbf{c}}_{:,n}$. Algorithms are then compared according three criteria:

- The mean least square error between the 3D spectra of the 4 fluorophores and their estimation denoted e_{AB} :

$$e_{AB} = \frac{1}{4} \sum_{n=1}^4 \frac{\|\hat{\mathbf{a}}_{:,n} \hat{\mathbf{b}}_{:,n}^T - \mathbf{a}_{:,n} \mathbf{b}_{:,n}^T\|_2}{\|\mathbf{a}_{:,n} \mathbf{b}_{:,n}^T\|_2}, \quad (40)$$

- The mean least square error between the concentration profiles of the 4 fluorophores and their estimation denoted⁴⁸⁵ e_C :

$$e_C = \frac{1}{4} \sum_{n=1}^4 \frac{\|\hat{\mathbf{c}}_{:,n} - \mathbf{c}_{:,n}\|_2}{\|\mathbf{c}_{:,n}\|_2} \quad (41)$$

- The computation time t_c .

All the following numerical simulations were performed using matlab R2017a on a linux system with an Intel Core i7-7700T CPU and 8GB RAM.

5.2.2. Synthetic data set 1 : small tensors and uncorrelated spectra

For this first data set, we consider a (quite) small number of mixtures by taking $R = 30$. Synthetic spectra used to build the data set are plotted on figure 1. These spectra are moderately⁴⁹⁰ correlated. More precisely the higher correlation coefficient between two columns of \mathbf{A} or \mathbf{B} is 0.63. Figure 2 shows the evolution of the average values of the three criteria with α . Average values are computed from the 100 tensors of each subset. First of all, recall we are in an over-factoring situation with⁵⁰⁵ two extra-components out of 6. In this context, it is noteworthy that all the considered algorithms performed quite well with estimation errors falling under 12 %.

Regarding both estimation error criteria (e_{AB} and e_C) DIAG-JDJS² provides better results than DIAG-SJDTE, A-FNNLS and AO-ADMM whatever the considered value of α . Hybrid algorithms improve significantly DIAG results. Among them the best results are clearly achieved by JDJS²-AO-ADMM. Now regarding the computing time, DIAG-JDJS² and JDJS²-AO-ADMM are about two to seven times faster than A-FNNLS and AO-ADMM but slower than SJDTE based algorithms. The gap⁵¹⁵ with SJDTE is mainly explained by the number of iterations computed during the symmetrization procedure.

5.2.3. Synthetic data set 2 : large tensors and uncorrelated spectra

We now show what happen on bigger data tensors. FEEM dimensions used for data set 1 are common while the number of mixtures can vary a lot from one real word application to another. Therefore, here we use the same spectra as synthetic data set 1 but we now consider a larger number of mixtures by taking $R = 200$. The JEVD step in data set 1 involved 870 matrices of size 6. Thus, we had $K/N = 145$ and $KN^3 \approx 2.10^5$. Here, the situation is different since we have now : $K = 39800$, $K/N \approx 6633$ and $KN^3 \approx 9.10^6$. In other words, the numerical complexity has been multiplied by 45 whereas the size of the matrix set was already significantly greater than the matrix size. We thus decimate the matrix set by a factor 10 (we leave out 90% of the matrices) in order to speed up the DIAG algorithm as explained in section 2. The same decimation will be applied to the next two synthetic data sets. Results are plotted on figure 3. These results are similar to data set 1. In particular, JDJS²-AO-ADMM still provides the lowest estimation errors while keeping a very limited computing time. Indeed thanks to the decimation JDJS²-AO-ADMM is now about ten and fifteen times faster than A-FNNLS and AO-ADMM respectively.

5.2.4. Synthetic data set 3 : large tensors and correlated excitation spectra

Here, we keep $R = 200$ but we modify the position (in the wavelength range) of the excitation spectra of fluorophore 2 and 4 in order to increase the overlap with the excitation spectra of fluorophore 1 and 3 (the shapes of the spectra remain unchanged). Correlation coefficients between fluorophores 1 and 2 on one side and between fluorophores 3 and 4 on the other side are now equal to 0.98. Results are plotted on figure 4. If we consider the whole range of α values, JDJS²-AO-ADMM, A-FNNLS and AO-ADMM outperform the three other algorithms in term of estimation error. The correlations affect the estimation performances of JDJS²-AO-ADMM when $\alpha \geq 0.4$. In this range, A-FNNLS and AO-ADMM provide now the best results. Under this critical value JDJS²-AO-ADMM performs slightly better than A-FNNLS and AO-ADMM. It is worth mentioning that at the lowest values of α (0.05 and 0.1) DIAG-JDJS² competes with those algorithms. Regarding the computing times, t_c (JDJS²-AO-ADMM) is about 20 to 30 times lower than t_c (A-FNNLS) and t_c (AO-ADMM) whatever the value of α . This indicate that we could decrease the decimation factor for $\alpha \geq 0.4$ in order to improve the performances of JDJS²-AO-ADMM.

5.2.5. Synthetic data sets 4 : large tensors and correlated excitation and emission spectra

Here, we modify both the excitation and the emission spectra to obtain significant correlations in matrices \mathbf{A} and \mathbf{B} . Fluorophores 1 and 2 have a coefficient correlation of 0.98 in excitation and fluorophores 3 and 4 have a coefficient correlation of 0.99 in emission. Results are plotted on figure 5. JDJS²-AO-ADMM, A-FNNLS and AO-ADMM still outperform the other algorithms in term of estimation errors. Regarding the 3D-spectra error, they provide similar results. Mean values of e_{AB} are 0.065 for JDJS²-AO-ADMM, 0.072 for AO-ADMM

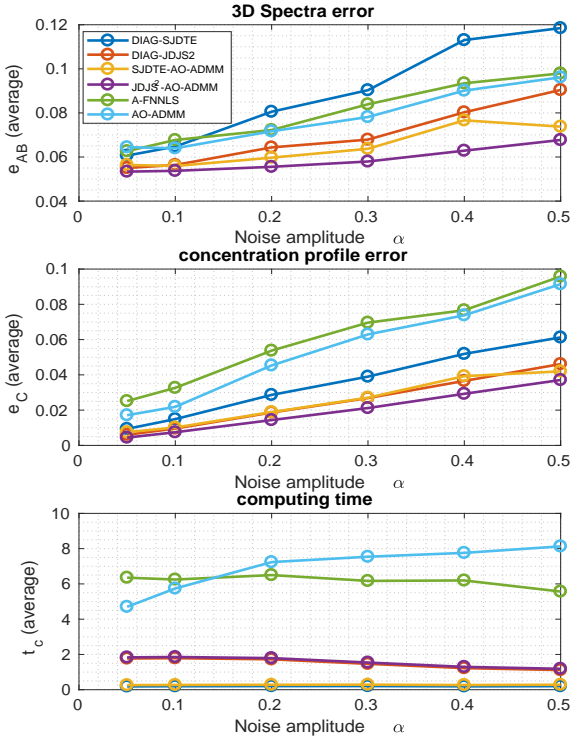


Figure 2: Synthetic data set 1: evolution of the average values of the three comparison criteria with respect to the noise amplitude.

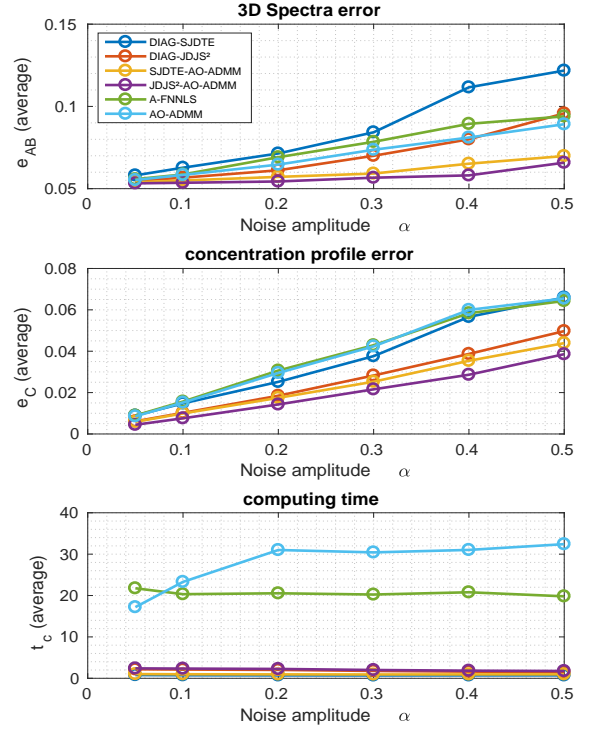


Figure 3: Synthetic data set 2: evolution of the average values of the three comparison criteria with respect to the noise amplitude.

and 0.075 for A-FNNLS. If we look at e_C plots, the gap between JDS²-AO-ADMM and AO-ADMM increases slightly: mean values of e_C are 0.03 for JDS²-AO-ADMM, 0.048 for AO-ADMM and 0.035 for A-FNNLS. If we compare these results with those of data set 3, it seems that two correlations in one mode (data set 3) affect more severely the proposed approach than one correlation in two modes (data set 4). Finally, as for data set 3, the main difference between the different approaches is the computation speed. Indeed, mean values of t_c are 1.35s for JDS²-AO-ADMM, 32.6s for AO-ADMM and 30.6s for A-FNNLS.

5.3. Simulations on real data sets

5.3.1. Real data set 1

This data set is composed of 11 FEEM, measured from diluted mixtures of fluorescein, quinine sulfate and tryptophane. Fluorophore spectra and concentration profiles are plotted on figure 6. These FEEM were used as reference spectra (no inner filter effect) in a previous study [38]. Excitation and emission wavelengths vary from 270 to 550 nm with a step of 5 nm. The data set is thus a tensor of size $(57 \times 57 \times 11)$. Rayleigh and Raman scatters were removed numerically from each measured FEEM using the method proposed by Zepp [39]. We refer to the above mentioned paper for details about sample preparation and FEEM acquisition. We compare the results obtained from various CPD computed by the previous algorithms. Indeed, in order to evaluate the influence of possible over-factoring,

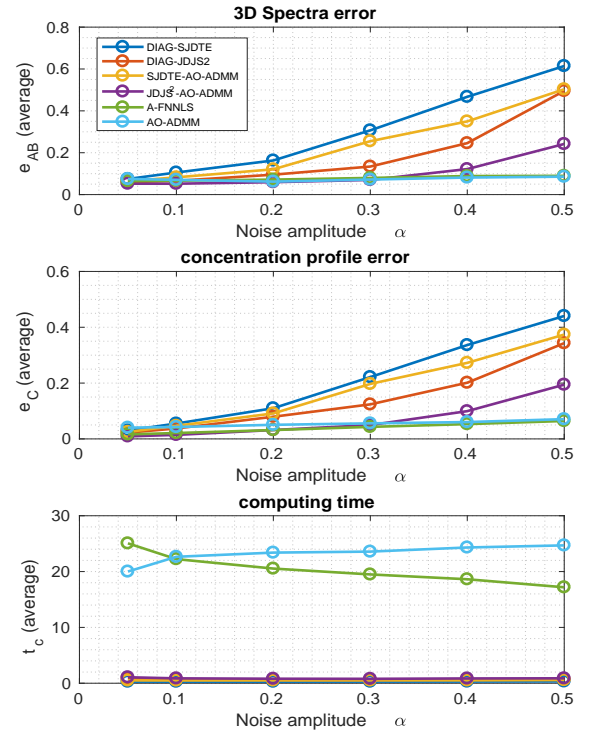


Figure 4: Synthetic data set 3: evolution of the average values of the three comparison criteria with respect to the noise amplitude.

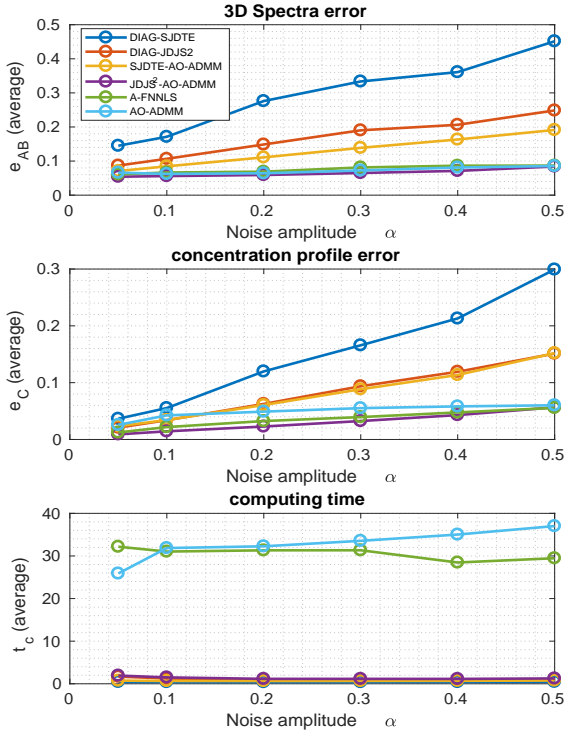


Figure 5: Synthetic data set 4: evolution of the average values of the three comparison criteria with respect to the noise amplitude.

we vary the CPD rank from 3 to 7, thus the number of extra-components, F , varies from 0 to 4. Algorithms settings are the same as for synthetic data set 1 and the same comparison criteria are used. Results are plotted on figure 7. When the correct number of components is used ($F = 0$), all algorithms perform well and give similar results in term of factor estimation error. From $F = 1$ to $F = 3$, DIAG and hybrid algorithms clearly outperform the alternating algorithms. Finally at $F = 4$, all algorithms fail. If we focus on e_{AB} , JDJS² algorithm helps to improve DIAG performances. However, here, when the hybrid approach is used, we cannot discriminate between SJDTE and the proposed algorithm. Thus, these results mainly highlight the ability of the DIAG and hybrid approaches to deal with overfactoring on a real data set. These algorithms are also clearly faster than A-FNNLS and AO-ADMM with random initialization in particular for $F > 0$.

5.3.2. Real data set 2

We performed the same comparison study on an other data set which is publicly available at <http://www.models.life.ku.dk/Fluorescence> [40, 41]. This data set consists in 16 mixtures of resorcinol, tryptophane and tyrosine and corresponds to dataset #1 in the web page³. Fluorescence intensity was measured for 19 different excitation wavelengths and

³We chose this data set because, according to the publisher of the data, in all other data sets one of the components "contains some impurities, and thus gives rise to an extra component"

136 different emission wavelengths. We removed the data corresponding to excitation wavelength shorter than 250 nm in order to focus on the main excitation peaks of the 3 fluorophores. We also removed the first sample because it corresponds to a pure fluorophore. Thus our final data set is a tensor of size $16 \times 136 \times 15$. Rayleigh and Raman scatters were removed as explained previously. Fluorophore spectra and concentration profiles are plotted on figure 8. Fluorophore spectra are highly correlated both in excitation and emission making this data set quite difficult. Results are plotted on figure 9. First of all, performances of DIAG and hybrid approaches are remarkably stable with respect to the number of extra-components, whatever the considered criterion. In contrast, A-FNNLS and AO-ADMM error plots are quite fluctuating and A-FNNLS t_c plot shows a clear increase of the computing time with the number of extra-components. Above all, DIAG-JDJS² and JDJS²-AO-ADMM clearly provide the lowest estimation errors whatever the number of extra-components. For instance at $F = 0$, e_{AB} is about 0.04 for DIAG-JDJS² and JDJS²-AO-ADMM, about 0.18 for DIAG-SJDTE and SJDTE-AO-ADMM and about 0.3 for A-FNNLS and AO-ADMM. Finally, DIAG and hybrid algorithms are much faster than A-FNNLS and AO-ADMM (at $F = 0$ we have $t_c(\text{JDJS}^2\text{-AO-ADMM})=0.22\text{s}$, $t_c(\text{A-FNNLS})=1.6\text{s}$ and $t_c(\text{AO-ADMM})=3.9\text{s}$).

6. Conclusion

In this study, we have first recalled how the canonical polyadic decomposition of a multiway array can be computed by means of joint eigenvalue decomposition. At this occasion, we also explain how this kind of CPD algorithm can be easily speed-up by matrix decimation in order to obtain fast algorithms. We have then showed how the JEVD can be rewritten into the simpler problem of orthogonal joint diagonalization of a set of positive definite matrices. This is done by joint symmetrization of the original matrix set. We have thus proposed an *ad hoc* joint symmetrization algorithm based on a sweeping procedure and multiplicative updates. This allowed us to introduce an original JEVD algorithm called JDJS² based on a two step process: joint diagonalization after joint symmetrization. This approach can thus take benefit of efficient existing OJD algorithms. We have then proposed to apply this approach to the CPD of data tensors containing real or synthetic fluorescence measurements by including JDJS² into a JEVD-based CPD algorithm. We call the resulting algorithm algorithm DIAG-JDJS². Both synthetic and real data sets were built to be challenging. Two main difficulties were considered: the number of component (i.e. the rank of the decomposition) is unknown and overestimated and several factors can be highly correlated. The proposed DIAG-JDJS² algorithm was compared to an other version of DIAG using a recent reference JEVD algorithm and with two reference CPD algorithms based on an alternating approach namely A-FNNLS and AO-ADMM. Our results first show that in most situations, the proposed algorithm significantly improves DIAG performances in term of factor estimation accuracy. Second, DIAG-JDJS² competes with and

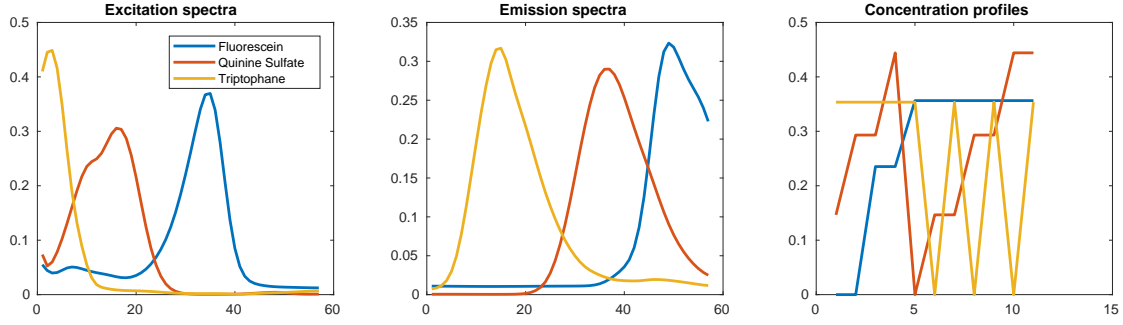


Figure 6: Real data set 1: Fluorophore spectra and concentration profiles.

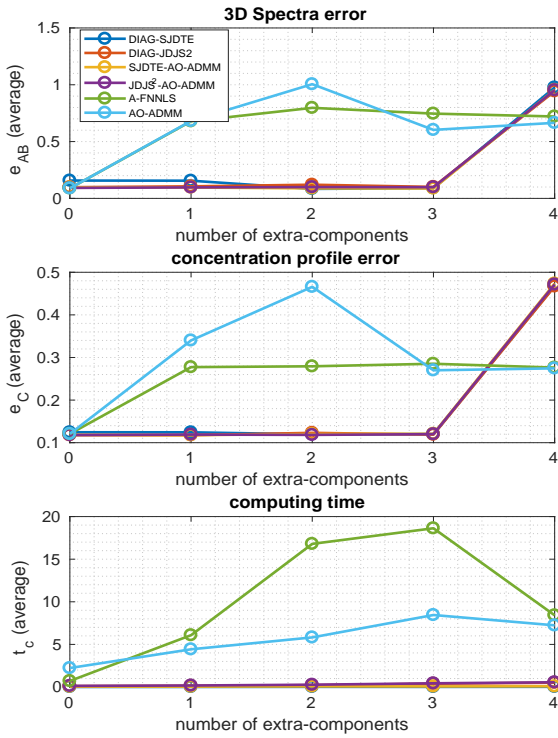


Figure 7: Real data set 1: evolution of the average values of the three comparison criteria with respect to the number of extra-components.

sometimes outperforms the A-FNNLS and AO-ADMM algorithms in the simplest situations (in terms of noise level and factor correlations) while having a computing time significantly lower. We then show that A-FNNLS and AO-ADMM are more robust to the noise level and correlations between factors. However, we also show that DIAG-JDJS² provides a very good starting point for alternating algorithms. This hybrid approach not only achieves the best performances in terms of estimation accuracy (in most of the considered situations) but it is also very fast in comparison to a classical random initialization strategy such as the best fitting scheme.

This study also opens some perspectives. In particular, the proposed JEVD algorithm allows us to impose a non-negative constraint on the eigenvalues. This feature is not really exploited here. An extension to the complex case is also worth considering. Regarding the CPD, here we use matrix decimation in the DIAG algorithm to build a smaller JEVD problem. Better selection strategies may be investigated. Finally it may be interesting to include JDJS² in other CPD algorithms based on JEVD.

- [1] F. L. Hitchcock, Multiple invariants and generalized rank of a p-way matrix or tensor, *J. Math. and Phys.* 7 (1) (1927) 39–79.
- [2] R. A. Harshman, Foundation of PARAFAC procedure: Models and conditions for an 'explanatory' multi-mode factor analysis, *UCLA working papers in Phonetics* (16) (1970) 1–84.
- [3] J. B. Kruskal, Three-way arrays: Rank and uniqueness of trilinear decompositions, *Linear Algebra and Applications* 18 (1977) 95–138.
- [4] N. Sidiropoulos, R. Bro, On the uniqueness of multilinear decomposition of n-way arrays, *Journal of Chemometrics* 14 (2000) 229–239.
- [5] B. Hunyadi, D. Camps, L. Sorber, W. V. Paesschen, M. D. Vos, S. V. Huffel, L. D. Lathauwer, Block term decomposition for modelling epileptic seizures, *EURASIP Journal on Advances in Signal Processing*, Special Issue on Recent Advances in Tensor Based Signal and Image Processing 139.
- [6] H. Becker, L. Albera, P. Comon, R. Gribonval, F. Wendling, I. Merlet, Brain source imaging: from sparse to tensor models, *IEEE Signal Processing Magazine*, special issue on Brain-Computer Interfaces 32 (6) (2015) 100–112.
- [7] N. Hao, M. E. Kilmer, K. Braman, R. C. Hoover, Facial recognition using tensor-tensor decompositions, *SIAM Journal on Imaging Sciences* 6 (1) (2013) 437–463.
- [8] N. D. Sidiropoulos, G. B. Giannakis, R. Bro, Blind PARAFAC receivers for DS-CDMA systems, *IEEE Transactions On Signal Processing* 48 (8) (2000) 810–823.
- [9] P. Comon, C. Jutten, *Handbook of blind source separation*, Academic Press, 2010.
- [10] R. Bro, Parafac, tutorial and applications, *Chemom. Intel. Lab. Syst.* 38 (1997) 149–171.
- [11] X. Luciani, S. Mounier, H. M. Paraquetti, R. Redon, Y. Lucas, A. Bois,

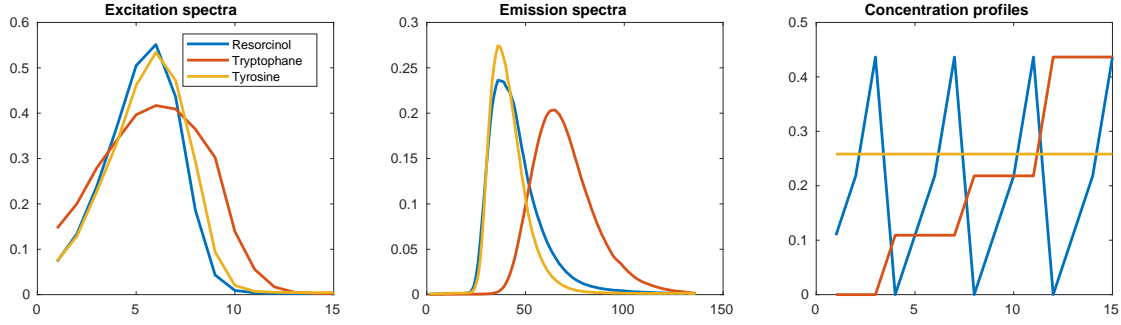


Figure 8: Real data set 2: Fluorophore spectra and concentration profiles.

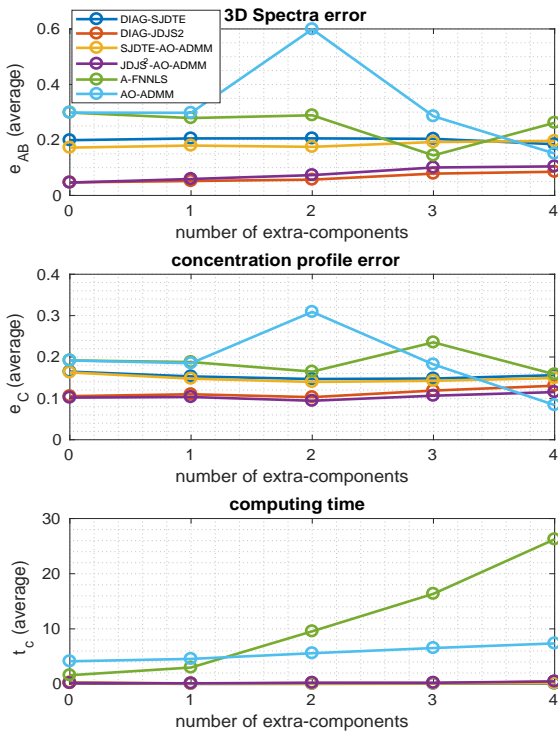


Figure 9: Real data set 2: evolution of the average values of the three comparison criteria with respect to the number of extra-components.

- 670 L. Lacerda, M. Raynaud, M. Ripert, Tracing of dissolved organic matter from the sepetiba bay (brazil) by parafac analysis of total luminescence matrices 65 (2008) 148–57.
- [12] E. Acar, D. Dunlavy, T. Kolda, Scalable tensor factorizations with missing data, 2010, pp. 701–712.
- 675 [13] G. Tomasi, R. Bro, A comparison of algorithms for fitting the parafac model, Computational Statistics & Data Analysis 50 (7) (2006) 1700–1734.
- [14] L. D. Lathauwer, A link between the canonical decomposition in multi-linear algebra and simultaneous matrix diagonalization, SIAM Journal on matrix Analysis and Applications 28 (3) (2006) 642–666.
- 680 [15] F. Roemer, M. Haardt, A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalization (SECSI), Elsevier Signal Processing 93 (2013) 2462–2473.
- [16] X. Luciani, L. Albera, Canonical polyadic decomposition based on joint eigenvalue decomposition, Chemometrics and Intelligent Laboratory Systems 132 (0) (2014) 152 – 167.
- 685 [17] S. Hajipour, L. Albera, M. B. Shamsollahi, I. Merlet, Canonical polyadic decomposition of complex-valued multi-way arrays based on simultaneous schur decomposition, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada, 2013, pp. 4178–4182.
- 690 [18] L. De Lathauwer, B. De Moor, J. Vandewalle, Computation of the canonical decomposition by means of a simultaneous Schur decomposition, SIAM Journal on Matrix Analysis and Applications 26 (2) (2004) 295–327.
- 695 [19] J. F. Cardoso, A. Souloumiac, Jacobi angles for simultaneous diagonalization, SIAM Journal Matrix Analysis and Applications 17 (1) (1996) 161–164.
- [20] B. N. Flury, W. Gautschi, An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form, SIAM J. Sci. Stat. Comput. 7 (1).
- 700 [21] D.-T. Pham, Joint approximate diagonalization of positive definite Hermitian matrices, SIAM J. Matrix Anal. Appl. 22 (4) (2001) 1136–1152.
- [22] T. M. Cover, J. A. Thomas, Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing), Wiley-Interscience, New York, NY, USA, 2006.
- 705 [23] L. D. Lathauwer, B. D. Moor, J. Vandewalle, Computation of the canonical decomposition by means of a simultaneous generalized schur decomposition, SIAM Journal on Matrix Analysis and Applications 26 (2001) 295–327.
- 710 [24] R. Bro, S. De Jong, A fast non-negativity-constrained least squares algorithm, Journal of Chemometrics 59 (1997) 393–401.
- [25] R. Bro, Parafac, tutorial and applications, Chemometrics and Intelligent Laboratory Systems 38 (2) (1997) 149–171.
- 715 [26] The n-way toolbox for matlab, Chemometrics and Intelligent Laboratory Systems 52 (1) (2000) 1 – 4.
- [27] R. Iferroudjene, K. Abed-Meraim, A. Belouchrani, A new jacobi-like method for joint diagonalization of arbitrary non-defective matrices, Applied Mathematics and Computation 211 (2) (2009) 363–373.
- 720 [28] T. Fu, X. Gao, Simultaneous diagonalization with similarity transformation for non-defective matrices, in: ICASSP 2006, 2006 IEEE International Conference on Acoustics Speech and Signal Processing, Vol. 4, 2006, pp. 1137–1140.

- [29] R. André, X. Luciani, E. Moreau, A new class of block coordinate algorithms for the joint eigenvalue decomposition of complex matrices, *Signal Processing* 145 (2018) 78 – 90.
- [30] R. André, X. Luciani, E. Moreau, Joint eigenvalue decomposition algorithms based on first-order Taylor expansion, *IEEE Transactions on Signal Processing* 68 (2020) 1716–1727.
- [31] X. Luciani, L. Albera, Joint eigenvalue decomposition of non-defective matrices based on the LU factorization with application to ICA, *IEEE Transactions on Signal Processing* 63 (17) (2015) 4594–4608.
- [32] C. L. Lawson, R. Hanson, Solving least squares problems, *Classics in Appl. Math.*, Vol. 15, SIAM, 1995.
- [33] A. Mosesov, Adaptive non-negative least squares with applications to non-negative matrix factorization, M.Sc. report under N. Sidiropoulos' supervision.
- [34] J. Lakowicz, "Principles of Fluorescence Spectroscopy", Plenum Press, New York and London, 1983.
- [35] K. Huang, N. D. Sidiropoulos, A. P. Liavas, A flexible and efficient algorithmic framework for constrained matrix and tensor factorization, *IEEE Transactions on Signal Processing* 64 (19) (2016) 5052–5065.
- [36] J. E. Cohen, Environmental Multiway Data Mining, Theses, Université Grenoble Alpes (Sep. 2016).
URL <https://hal.archives-ouvertes.fr/tel-01371777>
- [37] P. Comon, X. Luciani, A. L. F. De Almeida, Tensor decompositions, alternating least squares and other tales, *Journal of Chemometrics* 23.
- [38] X. Luciani, R. Redon, S. Mounier, How to correct inner filter effects altering 3d fluorescence spectra by using a mirrored cell, *Chemometrics and Intelligent Laboratory Systems* 126 (2013) 91 – 99.
doi:<https://doi.org/10.1016/j.chemolab.2013.04.014>.
URL <http://www.sciencedirect.com/science/article/pii/S0169743913000804>
- [39] R. G. Zepp, W. M. Sheldon, M. A. Moran, Dissolved organic fluorophores in southeastern US coastal waters: correction method for eliminating Rayleigh and Raman scattering peaks in excitation–emission matrices, *Marine Chemistry* 89 (1) (2004) 15 – 36, cDOM in the Ocean: Characterization, Distribution and Transformation.
doi:<https://doi.org/10.1016/j.marchem.2004.02.006>.
URL <http://www.sciencedirect.com/science/article/pii/S0304420304000799>
- [40] A. Rinnan, Application of PARAFAC on spectral data, Ph.D. thesis, Royal Veterinary and Agricultural University (2004).
- [41] R. Bro, A. Rinnan, N. K. M. Faber, Standard error of prediction for multilinear PLS - 2. practical implementation in fluorescence spectroscopy, *Chemometrics and Intelligent Laboratory Systems* 75 (1) (2005) 69–76.
doi:[10.1016/j.chemolab.2004.04.014](https://doi.org/10.1016/j.chemolab.2004.04.014).

- Canonical Polyadic Decomposition (CPD) also known as PARAFAC has become an important chemometric tool to decompose multiway data sets, notably in fluorescence spectroscopy.
- Joint eigenvalue decomposition and joint diagonalization are at the center of efficient CPD algorithms.
- We propose an original joint eigenvalue decomposition algorithm based on joint symmetrization.
- This algorithm is successfully applied to the CPD of synthetic fluorescence data sets.
- We show that our approach improves the estimation of the CPD factors in difficult but realistic cases (correlated factors and unknown number of factors.)

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: