



**HAL**  
open science

# A new class of block coordinate algorithms for the Joint EigenValue Decomposition of complex matrices

Rémi André, Xavier Luciani, Eric Moreau

## ► To cite this version:

Rémi André, Xavier Luciani, Eric Moreau. A new class of block coordinate algorithms for the Joint EigenValue Decomposition of complex matrices. *Signal Processing*, 2018, 145, pp.78-90. 10.1016/j.sigpro.2017.11.016 . hal-01862357

**HAL Id: hal-01862357**

**<https://hal.science/hal-01862357>**

Submitted on 27 Aug 2018

**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.

# A new class of block coordinate algorithms for the Joint EigenValue Decomposition of complex matrices

Rémi André<sup>a,b,c</sup>, Xavier Luciani<sup>a,b,c</sup>, Eric Moreau<sup>a,b,c</sup>

<sup>a</sup>*Laboratoire des Sciences de l'Information et des Systèmes, UMR CNRS 7296.*

<sup>b</sup>*Aix Marseille Université, CNRS, ENSAM, 13397 Marseille et Université de Toulon, 83957 La Garde, France.*

<sup>c</sup>*EURASIP member.*

---

## Abstract

Several signal processing problems can be written as the joint eigenvalue decomposition of a set of noisy matrices. This is notably the case of the canonical polyadic decomposition of tensors that is widely applied in blind identification and source separation. Most of the existing algorithms are based on a block coordinate procedure and require significant modifications to deal with complex-valued matrices. These modifications decrease algorithms performances either in terms of estimation accuracy of the eigenvectors or in terms of computational cost. Therefore, we propose a class of algorithms working equally with real- or complex-valued matrices. These algorithms are still based on a block coordinate procedure and multiplicative updates. The originality of the proposed approach lies in the structure of the updating matrix and in the way the optimization problem is solved in  $\mathbb{C}$ . This structure is parametrized and allows to define up to five different JEVD algorithms. Thanks to numerical simulations, we show that, with respect to the more accurate algorithms of the literature, this approach improves the estimation of the eigenvectors and has a computational cost significantly lower. Finally, as an application example, one of the proposed algorithm is successfully applied to the blind sources separation of DS-CDMA signals.

---

*Email addresses:* [randre@univ-tln.fr](mailto:randre@univ-tln.fr) (Rémi André), [luciani@univ-tln.fr](mailto:luciani@univ-tln.fr) (Xavier Luciani), [moreau@univ-tln.fr](mailto:moreau@univ-tln.fr) (Eric Moreau)

*Keywords:* Joint eigenvalue decomposition; block coordinate procedure; canonical polyadic decomposition; source separation; joint diagonalization; DS-CDMA.

---

## 1. Introduction

Joint EigenValue Decomposition (JEVD), also called joint diagonalization by similarity, is an important issue for a number of signal processing applications such as directions of arrival estimation [1], joint angle-delay estimation [2], multi-dimensional harmonic retrieval [3], Independent Component Analysis (ICA) [4, 5, 6, 7, 8] and Canonical Polyadic Decomposition (CPD) of tensors [9, 10, 11]. JEVD problems occur when a set of  $K$  non-defective matrices  $\mathbf{M}^{(k)}$  shares the same basis of eigenvectors:

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

where the invertible matrix  $\mathbf{A} \in \mathbb{C}^{N \times N}$  is the common matrix of eigenvectors and the  $K$  matrices  $\mathbf{D}^{(k)} \in \mathbb{C}^{N \times N}$  are all diagonal and contain the eigenvalues of corresponding  $\mathbf{M}^{(k)}$  matrices. The goal is then to estimate  $\mathbf{A}$  or  $\mathbf{A}^{-1}$  from matrices  $\mathbf{M}^{(k)}$ .

Naive approaches consist in considering the  $K$  eigenvalue decompositions separately or a linear combination of those. The main problem is then that if some eigenvalues are degenerated or very close, the corresponding eigenvectors cannot be correctly identified. Moreover, in practice this problem is accentuated by the presence of noise because small perturbations of a matrix can strongly affect its eigenvectors [9, 12]. As a consequence, a recommended solution is to decompose the whole matrix set jointly. One usual way of doing is to make matrices  $\mathbf{M}^{(k)}$  as diagonal as possible within the same change of basis. That's why JEVD can be seen as a joint diagonalization problem. More precisely we speak of joint diagonalization by similarity in contrast with the Joint Diagonalization by Congruence (JDC) problem for which the inverse of the matrix  $\mathbf{A}$  in (1) is replaced by the conjugate transpose of  $\mathbf{A}$ . Of course, JDC and JEVD are equivalent if

$\mathbf{A}$  is a unitary matrix but this is not necessary the case here. A second consequence is that JEVD algorithms are often inspired by JDC algorithms. Indeed most of them resort to an iterative block coordinate procedure adapted from the original Jacobi method [13]. This means that matrix  $\mathbf{A}$  (or  $\mathbf{A}^{-1}$ ) is built by successive multiplicative updates. Each update involves a small set of parameters (with respect to  $N^2$ ) that allows to build the updating matrix. JEVD algorithms differ one from another in the way these parameters are defined and computed. Several family of algorithms can then be identified. Algorithms of the first family look for the updating matrix in the form of a QR factorization [3, 14]. More recently, three algorithms based on the polar decomposition were proposed: SHear RoTation algorithm (SH-RT) [15], Joint Unitary Shear Transformation (JUST) [16] and Joint Diagonalization algorithm based on Targeting hyperbolic Matrices (JDTM) [11]. These algorithms build the updating matrix as a product of a unitary and an hermitian matrix. Finally a third family of algorithms based on the LU factorization and called Joint Eigenvalue decomposition algorithms based on Triangular matrices (JET) was introduced in [8]. Thus, in practice the updating matrix is computed as a product of two matrices having a particular structure: unitary, Hermitian or triangular. At this stage, it is worth mentioning that all of the previously mentioned algorithms compute these two matrices separately by solving two successive optimization problems. Furthermore, two versions of each algorithm have been consistently proposed by their authors in order to deal with real or complex-valued matrices respectively. Indeed, the extension of the previous methods to the complex case implies to work in  $\mathbb{R}^2$  instead of  $\mathbb{C}$  and this involves some significant modifications. According to the used approach, these modifications are not trivial and make the optimization step more intricate. The consequence is a performance decrease of the algorithms [8]. More precisely, in the complex case, algorithms based on polar decomposition are robust to the noise power but require many iterations to converge hence a high computational cost. Conversely algorithms based on the LU decomposition have a low computational cost but are less robust to the noise power. As a consequence we propose here a new class of algorithms that

work equally for real- or complex-valued matrices.

The proposed approach has common points with the previous methods: it is also based on a block coordinate procedure and the updating matrix is still computed as a product of factorization matrices. However its originality is twofold. First, these matrices are estimated conjointly from a simple eigenvalue decomposition of a  $2 \times 2$  matrix. This strategy is inspired from [17] for the JDC problem and allows to reduce the numerical complexity of the block coordinate step. Second we propose a parametrized expression of the updating matrix that cover different matrix factorizations. We then show that we can switch from one matrix factorization to another by changing only one parameter. This allows to define a class of five algorithms sharing the same structure. One of these algorithm was briefly presented in [18]. Moreover, we have the possibility to pass from one version to another at each new iteration of the optimization process. This approach has the advantage to work in  $\mathbb{C}$  throughout the process. In other words, no modifications are required to deal with real- or complex-valued matrices.

The paper is organized as follow: in section 2, we recall the principle of block coordinate JEVD algorithms. In section 3, we describe the proposed method and the algorithms. In section 4, we compare the numerical complexity of these algorithms to the ones of the existing algorithms. Section 5 is dedicated to numerical simulations. We have evaluated the performances of the proposed algorithms to compute the JEVD of complex matrices according to several scenarios. Comparison are made with all the other JEVD algorithms. Finally in section 6, we show how the proposed approach can be used to achieve the blind sources separation of telecommunication signals.

**Notations.** In the following scalars are denoted by a lower case ( $a$ ), vectors by a boldface lower case ( $\mathbf{a}$ ) and matrices by a boldface upper case ( $\mathbf{A}$ ).  $a_i$  is the  $i$ -th element of vector  $\mathbf{a}$  and  $A_{i,j}$  is the  $(i, j)$ -th element of matrix  $\mathbf{A}$ . Operator  $\|\bullet\|$  is the Frobenius norm of the argument matrix. Operator  $\text{ZDiag}\{\bullet\}$  sets to zero the diagonal of the argument matrix.  $\mathbf{I}$  is the identity matrix. Modulus

and conjugate of any complex number  $z$  are denoted by  $|z|$  and  $\bar{z}$  respectively.  $k \in [1, K]_{\mathbb{N}}$  is the sequence of natural integers from 1 to  $K$ .

## 2. A block coordinate procedure

The JEVD problem consists in finding a matrix,  $\mathbf{B}$ , which jointly diagonalizes the given set of matrices  $\mathbf{M}^{(k)}$  in equation (1).  $\mathbf{B}$  is called the diagonalizing matrix and can be considered as an estimate of  $\mathbf{A}^{-1}$  up to a permutation and a scaling indeterminacy. This indeterminacy is inherent to the JEVD problem. An important uniqueness result has been shown in [9]. Let us define matrix  $\mathbf{\Omega}$  as

$$\mathbf{\Omega} = \begin{pmatrix} D_{11}^{(1)} & \cdots & D_{11}^{(K)} \\ \vdots & \cdots & \vdots \\ D_{NN}^{(1)} & \cdots & D_{NN}^{(K)} \end{pmatrix}. \quad (2)$$

The JEVD is unique up to a permutation and a scaling indeterminacy if and only if the rows of  $\mathbf{\Omega}$  are two by two distinct (*i.e.*  $\forall m, n$  with  $m \neq n$ ,  $\omega_m - \omega_n \neq \mathbf{0}$  where  $\omega_m$  and  $\omega_n$  are the  $m^{\text{th}}$  and the  $n^{\text{th}}$  rows of  $\mathbf{\Omega}$  respectively). Obviously this uniqueness condition makes sense only for  $K > 1$ . In the following, we will always assume that this condition is satisfied.

We want to build  $\mathbf{B}$  such that matrices  $\widehat{\mathbf{D}}^{(k)}$  defined by:

$$\widehat{\mathbf{D}}^{(k)} = \mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{-1}, \quad \forall k = 1, \dots, K \quad (3)$$

are as diagonal as possible. The method is iterative. At each iteration,  $\mathbf{B}$  is multiplicatively updated with a matrix  $\mathbf{X}$  in the following way:

$$\mathbf{B} \leftarrow \mathbf{X}\mathbf{B} \quad (4)$$

and the set of matrices  $\widehat{\mathbf{D}}^{(k)}$  is consequently updated as:

$$\widehat{\mathbf{D}}^{(k)} \leftarrow \mathbf{X}\widehat{\mathbf{D}}^{(k)}\mathbf{X}^{-1}, \quad \forall k = 1, \dots, K. \quad (5)$$

We expect that at the end of the iterative process, matrices  $\widehat{\mathbf{D}}^{(k)}$  are diagonalized and  $\mathbf{B}\mathbf{A}$  is close to the product between an invertible diagonal matrix

and a permutation matrix.  $\mathbf{B}$  is initialized from an appropriate initial guess  $\mathbf{B}_0$  and before the first iteration we set  $\widehat{\mathbf{D}}^{(k)} \leftarrow \mathbf{B}_0 \mathbf{M}^{(k)} \mathbf{B}_0^{-1}$ ,  $\forall k \in [1, K]_{\mathbb{N}}$ . The choice of  $\mathbf{B}_0$  will be discussed in section 5.

At each iteration,  $\mathbf{X}$  is computed thanks to a block coordinate procedure. Here, it means that  $\mathbf{X}$  is built from a set of  $N(N-1)/2$  matrices, denoted  $\mathbf{X}^{(i,j)}$  ( $i = 1, \dots, N-1$  and  $j = i, \dots, N$ ) in such a way that:

$$\mathbf{X} = \prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{X}^{(i,j)} \quad (6)$$

where matrices  $\mathbf{X}^{(i,j)}$  are equal to the identity matrix up to four unknown parameters:  $x_1^{(i,j)}$ ,  $x_2^{(i,j)}$ ,  $x_3^{(i,j)}$ ,  $x_4^{(i,j)}$ :

$$\begin{cases} \forall (p, q) \notin \{(i, i), (i, j), (j, i), (j, j)\}, & X_{p,q}^{(i,j)} = \delta_{p,q}, \\ X_{i,i}^{(i,j)} = x_1^{(i,j)}, & X_{i,j}^{(i,j)} = x_2^{(i,j)}, \\ X_{j,i}^{(i,j)} = x_3^{(i,j)}, & X_{j,j}^{(i,j)} = x_4^{(i,j)}, \end{cases} \quad (7)$$

where  $\delta$  is the Kronecker delta. As a consequence the update in (4) and (5) consists in  $N(N-1)/2$  successive  $(i, j)$ -updates of  $\mathbf{B}$  and  $\widehat{\mathbf{D}}^{(k)}$ , defined as:

$$\begin{cases} \mathbf{B} & \leftarrow \mathbf{X}^{(i,j)} \mathbf{B}, \\ \widehat{\mathbf{D}}^{(k)} & \leftarrow \mathbf{X}^{(i,j)} \widehat{\mathbf{D}}^{(k)} \left( \mathbf{X}^{(i,j)} \right)^{-1}, \quad \forall k \in [1, K]_{\mathbb{N}}. \end{cases} \quad (8)$$

The fact of having  $N(N-1)/2$  matrices sharing the structure defined in (7) allows to equivalently affect all the off-diagonal terms of  $\widehat{\mathbf{D}}^{(k)}$  with elementary operations. Thus, the block coordinate procedure transforms the problem of estimating  $\mathbf{X}$  into  $N(N-1)/2$  successive simpler optimization problems. Indeed each  $\mathbf{X}^{(i,j)}$  matrix is estimated by minimizing a wisely chosen cost function. Obviously, this sequential optimization framework is not equivalent to a global optimization of  $\mathbf{X}$ . That's why it needs several iterations to converge. It has been successfully used to solve any kind of joint diagonalization problems. One can see for instance [4, 19, 15] for examples in orthogonal JDC, non-orthogonal JDC and JEVD contexts respectively.

For each  $(i, j)$ -update in equation (8), we define:

$$\mathbf{N}^{(k)} = \mathbf{X}^{(i,j)} \widehat{\mathbf{D}}^{(k)} \left( \mathbf{X}^{(i,j)} \right)^{-1}, \quad \forall k \in [1, K]_{\mathbb{N}} \quad (9)$$

and we use the following cost function to compute the current matrix  $\mathbf{X}^{(i,j)}$ :

$$C(\mathbf{X}^{(i,j)}) = \sum_{k=1}^K |N_{i,j}^{(k)}|^2 + |N_{j,i}^{(k)}|^2. \quad (10)$$

This diagonalization criterion has been introduced in [19] for the JDC problem and it was successfully adapted to solve the JEVD problem in [11] and [8]. Minimizing this criterion is not equivalent to minimizing all the off diagonal entries of  $\mathbf{N}^{(k)} \forall k \in [1, K]_{\mathbb{N}}$  but in practice it offers good, empiric, convergence properties. A possible justification is that the only off-diagonal entries affected by both matrix multiplications in (9) are  $N_{i,j}^{(k)}$  and  $N_{j,i}^{(k)}$ .

One of the advantages of criterion (10) is that it allows to rewrite the problem involving only  $2 \times 2$  matrices instead of  $N \times N$  matrices. Since we are now only interested in  $N_{i,j}^{(k)}$  and  $N_{j,i}^{(k)}$  in the left member of (9), we can define matrices  $\tilde{\mathbf{N}}^{(k)}$  and  $\tilde{\mathbf{X}}^{(i,j)}$  as

$$\tilde{\mathbf{N}}^{(k)} = \begin{pmatrix} N_{i,i}^{(k)} & N_{i,j}^{(k)} \\ N_{j,i}^{(k)} & N_{j,j}^{(k)} \end{pmatrix}, \quad \forall k \in [1, K]_{\mathbb{N}}; \quad (11)$$

$$\tilde{\mathbf{X}}^{(i,j)} = \begin{pmatrix} x_1^{(i,j)} & x_2^{(i,j)} \\ x_3^{(i,j)} & x_4^{(i,j)} \end{pmatrix}. \quad (12)$$

Then, we can reduce (9) into:  $\forall k \in [1, K]_{\mathbb{N}}$ ,

$$\tilde{\mathbf{N}}^{(k)} = \tilde{\mathbf{X}}^{(i,j)} \begin{pmatrix} \hat{D}_{i,i}^{(k)} & \hat{D}_{i,j}^{(k)} \\ \hat{D}_{j,i}^{(k)} & \hat{D}_{j,j}^{(k)} \end{pmatrix} (\tilde{\mathbf{X}}^{(i,j)})^{-1} \quad (13)$$

and the cost function becomes:

$$C(\tilde{\mathbf{X}}^{(i,j)}) = \sum_{k=1}^K |\tilde{N}_{1,2}^{(k)}|^2 + |\tilde{N}_{2,1}^{(k)}|^2. \quad (14)$$

In the next section, we propose a general framework to minimize an approximation of  $C(\tilde{\mathbf{X}}^{(i,j)})$  from which we derive five different ways to compute  $\tilde{\mathbf{X}}^{(i,j)}$  (and thus  $\mathbf{X}^{(i,j)}$ ).

### 3. A new class of JEVD algorithms

#### 3.1. General framework

First of all, we recall that at each iteration the process has to be repeated in the same way for each couple  $(i, j)$  with  $i < j$ . In order to simplify the notation, in this section we will drop  $(i, j)$  subscripts by referring to  $\tilde{\mathbf{X}}^{(i,j)}$ ,  $x_1^{(i,j)}$ ,  $x_2^{(i,j)}$ ,  $x_3^{(i,j)}$  and  $x_4^{(i,j)}$  as  $\tilde{\mathbf{X}}$ ,  $x_1$ ,  $x_2$ ,  $x_3$  and  $x_4$ .

Thanks to the scaling indeterminacy of the JEVD problem, we can impose that  $\det\{\mathbf{X}^{(i,j)}\} = 1$ . Thereby we have:

$$\tilde{\mathbf{X}}^{-1} = \begin{pmatrix} x_4 & -x_2 \\ -x_3 & x_1 \end{pmatrix}. \quad (15)$$

As a consequence, the off diagonal elements of  $\tilde{\mathbf{N}}^{(k)}$  in (13) can be written:  $\forall k \in [1, K]_{\mathbb{N}}$ ,

$$\begin{cases} \tilde{N}_{1,2}^{(k)} = x_1 x_2 (\hat{D}_{j,j}^{(k)} - \hat{D}_{i,i}^{(k)}) + x_1^2 \hat{D}_{i,j}^{(k)} - x_2^2 \hat{D}_{j,i}^{(k)}, \\ \tilde{N}_{2,1}^{(k)} = x_3 x_4 (\hat{D}_{i,i}^{(k)} - \hat{D}_{j,j}^{(k)}) + x_4^2 \hat{D}_{j,i}^{(k)} - x_3^2 \hat{D}_{i,j}^{(k)}. \end{cases} \quad (16)$$

Let us define the two vectors

$$\mathbf{v} = \begin{pmatrix} x_1 x_2 \\ x_1^2 \\ x_2^2 \end{pmatrix}; \quad \mathbf{w} = \begin{pmatrix} x_3 x_4 \\ x_4^2 \\ x_3^2 \end{pmatrix} \quad (17)$$

and the two matrices

$$\mathbf{F}_{12} = \begin{pmatrix} \hat{D}_{j,j}^{(1)} - \hat{D}_{i,i}^{(1)} & \hat{D}_{i,j}^{(1)} & -\hat{D}_{j,i}^{(1)} \\ \vdots & \vdots & \vdots \\ \hat{D}_{j,j}^{(K)} - \hat{D}_{i,i}^{(K)} & \hat{D}_{i,j}^{(K)} & -\hat{D}_{j,i}^{(K)} \end{pmatrix}; \quad (18)$$

$$\mathbf{F}_{21} = \begin{pmatrix} \hat{D}_{i,i}^{(1)} - \hat{D}_{j,j}^{(1)} & \hat{D}_{j,i}^{(1)} & -\hat{D}_{i,j}^{(1)} \\ \vdots & \vdots & \vdots \\ \hat{D}_{i,i}^{(K)} - \hat{D}_{j,j}^{(K)} & \hat{D}_{j,i}^{(K)} & -\hat{D}_{i,j}^{(K)} \end{pmatrix}. \quad (19)$$

The two terms of the right hand member of (14) can be rewritten as:

$$\begin{cases} \sum_{k=1}^K |N_{1,2}^{(k)}|^2 = \mathbf{v}^H \mathbf{F}_{12}^H \mathbf{F}_{12} \mathbf{v}, \\ \sum_{k=1}^K |N_{2,1}^{(k)}|^2 = \mathbf{w}^H \mathbf{F}_{21}^H \mathbf{F}_{21} \mathbf{w} \end{cases} \quad (20)$$

These two terms can be minimized separately. Matrices  $\mathbf{F}_{12}^H \mathbf{F}_{12}$  and  $\mathbf{F}_{21}^H \mathbf{F}_{21}$  are semi-definite positive matrices, thereby both optimization problems can be seen as minor eigenvalue problems (as explained in [17]). Indeed it is well known that the norm-1 vector that minimize one of the quadratic form in (20) is a normalized eigenvector associated to the smallest eigenvalue of  $\mathbf{F}_{12}^H \mathbf{F}_{12}$  or  $\mathbf{F}_{21}^H \mathbf{F}_{21}$ . This eigenvector has to be unique up to the product by a scale coefficient. Consequently the smallest eigenvalue must not have a multiplicity superior to 1. However, by observing the structure of  $\mathbf{F}_{12}$  and  $\mathbf{F}_{21}$ , it is easy to check that if the diagonalizing solution is perfectly reached, these two matrices will be of rank 1 and so  $\mathbf{F}_{12}^H \mathbf{F}_{12}$  and  $\mathbf{F}_{21}^H \mathbf{F}_{21}$  will have a zero eigenvalue of multiplicity 2. Thus in order to avoid this problem, we firstly do the following assumption:

**Assumption 1.** *We assume that we are close to a stationary point. Here a stationary point is characterized by  $|x_2| \ll 1$  and  $|x_3| \ll 1$ . Moreover we assume that we are in the vicinity of the diagonalizing solution and so that  $\forall k \in [1, K]_{\mathbb{N}}, \|\text{ZDiag}\{\widehat{\mathbf{D}}^{(k)}\}\| \ll 1$ .*

Under assumption 1, we can neglect  $x_2^2 \widehat{D}_{j,i}^{(k)}$  and  $x_3^2 \widehat{D}_{i,j}^{(k)}$  in (16), yielding:  $\forall k \in [1, K]_{\mathbb{N}}$ ,

$$\begin{cases} \tilde{N}_{1,2}^{(k)} \simeq x_1 x_2 (\widehat{D}_{j,j}^{(k)} - \widehat{D}_{i,i}^{(k)}) + x_1^2 \widehat{D}_{i,j}^{(k)}, \\ \tilde{N}_{2,1}^{(k)} \simeq x_3 x_4 (\widehat{D}_{i,i}^{(k)} - \widehat{D}_{j,j}^{(k)}) + x_4^2 \widehat{D}_{j,i}^{(k)}. \end{cases} \quad (21)$$

Now we define the two vectors

$$\mathbf{v}_1 = \begin{pmatrix} x_1 x_2 \\ x_1^2 \end{pmatrix}; \quad \mathbf{w}_1 = \begin{pmatrix} x_3 x_4 \\ x_4^2 \end{pmatrix} \quad (22)$$

and the two matrices

$$\mathbf{E}_{12} = \begin{pmatrix} \widehat{D}_{j,j}^{(1)} - \widehat{D}_{i,i}^{(1)} & \widehat{D}_{i,j}^{(1)} \\ \vdots & \vdots \\ \widehat{D}_{j,j}^{(K)} - \widehat{D}_{i,i}^{(K)} & \widehat{D}_{i,j}^{(K)} \end{pmatrix}; \quad (23)$$

$$\mathbf{E}_{21} = \begin{pmatrix} \widehat{D}_{i,i}^{(1)} - \widehat{D}_{j,j}^{(1)} & \widehat{D}_{j,i}^{(1)} \\ \vdots & \vdots \\ \widehat{D}_{i,i}^{(K)} - \widehat{D}_{j,j}^{(K)} & \widehat{D}_{j,i}^{(K)} \end{pmatrix}. \quad (24)$$

The cost function to be minimized is then given by the right member of:

$$C(\tilde{\mathbf{X}}) \simeq \mathbf{v}_1^H \mathbf{E}_{12}^H \mathbf{E}_{12} \mathbf{v}_1 + \mathbf{w}_1^H \mathbf{E}_{21}^H \mathbf{E}_{21} \mathbf{w}_1. \quad (25)$$

We can now notice that even if at the convergence,  $\mathbf{E}_{12}^H \mathbf{E}_{12}$  and  $\mathbf{E}_{21}^H \mathbf{E}_{21}$  can be rank-1 matrices, their structure avoids the zero eigenvalue to be of multiplicity 2 because it exists at least one  $k \in [1, K]_{\mathbb{N}}$  such that  $\widehat{D}_{i,i}^{(k)} \neq \widehat{D}_{j,j}^{(k)}$ . Indeed, it is noteworthy that, at the convergence, this last condition is equivalent to the JEVD uniqueness condition given in introduction.

We denote  $\mathbf{e}$  and  $\mathbf{f}$  the normalized eigenvectors corresponding to the smallest eigenvalue of  $\mathbf{E}_{12}^H \mathbf{E}_{12}$  and  $\mathbf{E}_{21}^H \mathbf{E}_{21}$  respectively, and thus we have:

$$\begin{cases} \mathbf{e} = \alpha \mathbf{v}_1, \\ \mathbf{f} = \beta \mathbf{w}_1, \end{cases} \quad (26)$$

where  $\alpha$  and  $\beta$  are unknown scale parameters. Before going further, one should note that due to the scaling indeterminacy of the JEVD,  $\mathbf{B}$  is estimated up to the left product by a diagonal matrix. Thanks to the updating process, whatever the matrix  $\mathbf{X}$  chosen for the update, we still keep a JEVD problem that can be solved up to a diagonal matrix. Therefore, if  $\tilde{\mathbf{X}}$  is a minimizer of the right member, in practice we can seek for any scaled matrix of determinant 1 of the form  $\mathbf{\Lambda}^{-1} \tilde{\mathbf{X}}$  where  $\mathbf{\Lambda}$  is an unknown invertible diagonal matrix. Thus, at this stage we can sum up our problem as follow:

**Problem 1.** Let  $\tilde{\mathbf{X}}$  be an unknown complex-valued matrix whose determinant is equal to 1:

$$\tilde{\mathbf{X}} = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}; \quad (27)$$

and  $\mathbf{e}$  and  $\mathbf{f}$  two given non-zero vectors defined by :

$$\mathbf{e} = \alpha \begin{pmatrix} x_1 x_2 \\ x_1^2 \end{pmatrix}; \quad \mathbf{f} = \beta \begin{pmatrix} x_3 x_4 \\ x_4^2 \end{pmatrix} \quad (28)$$

where  $\alpha, \beta$  are unknown parameters, we want to find from  $\mathbf{e}$  and  $\mathbf{f}$  a matrix  $\mathbf{Y}$  verifying:  $\det\{\mathbf{Y}\} = 1$  and  $\tilde{\mathbf{X}} = \mathbf{\Lambda}\mathbf{Y}$  where  $\mathbf{\Lambda}$  is an invertible diagonal matrix. Then we will use  $\mathbf{Y}$  as the updating matrix.

We can easily check that problem 1 admits one degree of freedom. Furthermore, since  $\mathbf{e}$  and  $\mathbf{f}$  are non-zero we know that  $x_1 \neq 0$  and  $x_4 \neq 0$ . As a consequence, we can look for  $\mathbf{Y}$  having the following structure:

$$\mathbf{Y} = \frac{1}{\sqrt{y_4 - y_2 y_3}} \begin{pmatrix} 1 & y_2 \\ y_3 & y_4 \end{pmatrix}, \quad \text{with } y_4 \neq 0. \quad (29)$$

We then immediately deduce from the definition of  $\mathbf{e}$  and  $\mathbf{f}$  that

$$\begin{cases} y_2 = \frac{x_2}{x_1} = \frac{e_1}{e_2}, \\ y_3 = \frac{x_3}{x_4} y_4 = \frac{f_1}{f_2} y_4. \end{cases} \quad (30)$$

Therefore, with this formulation the key point is the choice of  $y_4$ . In the next subsections, we will propose five different solutions to problem 1 corresponding to different choices of  $y_4$  based on different factorizations of  $\tilde{\mathbf{X}}$ . In this purpose, before to proceed, we recall in the following propositions three classical factorizations of a square complex-valued matrix of size 2 and introduce three other ones.

In each of the following propositions,  $\tilde{\mathbf{X}}$  is an invertible complex-valued matrix of size  $2 \times 2$  with  $\tilde{X}_{1,1} \neq 0$  and  $\tilde{X}_{2,2} \neq 0$ .

**Proposition 1.** *It exists an invertible diagonal matrix  $\Lambda$  and two complex numbers  $y_2$  and  $y_3$  such that:*

$$\tilde{\mathbf{X}} = \Lambda \begin{pmatrix} 1 & y_2 \\ y_3 & 1 \end{pmatrix} \frac{1}{\sqrt{1 - y_2 y_3}}. \quad (31)$$

**Proposition 2** (LU factorization). *It exists an invertible diagonal matrix  $\Lambda$  and two complex numbers  $l$  and  $u$  such that:*

$$\tilde{\mathbf{X}} = \Lambda \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} \begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix}. \quad (32)$$

**Proposition 3** (RQ factorization). *It exists an invertible diagonal matrix  $\Lambda$  and two complex numbers  $r$  and  $q$  such that:*

$$\tilde{\mathbf{X}} = \Lambda \begin{pmatrix} 1 & r \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\bar{q} \\ q & 1 \end{pmatrix} \frac{1}{\sqrt{1 + |q|^2}}. \quad (33)$$

**Proposition 4** (Algebraic RQ factorization). *It exists an invertible diagonal matrix  $\Lambda$  and two complex numbers  $r$  and  $q$  such that:*

$$\tilde{\mathbf{X}} = \Lambda \begin{pmatrix} 1 & r \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix} \frac{1}{\sqrt{1 + q^2}} \quad (34)$$

*if and only if  $\left(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}}\right)^2 \neq -1$ .*

**Proposition 5** (Algebraic LQ factorization). *It exists an invertible diagonal matrix  $\Lambda$  and two complex numbers  $l$  and  $q$  such that:*

$$\tilde{\mathbf{X}} = \Lambda \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix} \frac{1}{\sqrt{1 + q^2}} \quad (35)$$

*if and only if  $\left(\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}}\right)^2 \neq -1$ .*

**Proposition 6** (Algebraic polar decomposition). *If  $\left(\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}}\right)^2 \neq -1$  and  $\left(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}}\right)^2 \neq -1$ , then it exists an invertible diagonal matrix  $\Lambda$  and two complex numbers  $h$  and  $q$  such that:*

$$\tilde{\mathbf{X}} = \Lambda \begin{pmatrix} 1 & h \\ h & 1 \end{pmatrix} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix} \frac{1}{\sqrt{(1 - h^2)(1 + q^2)}} \quad (36)$$

Propositions 1 and 2 are well known results. Proposition 3 is a simpler formulation of the  $RQ$  factorization of  $\tilde{\mathbf{X}}$  taking advantage of  $\tilde{X}_{1,1} \neq 0$  and  $\tilde{X}_{2,2} \neq 0$ . Proofs of propositions 4, 5 and 6 are simple and are given in Appendix A.

### 3.2. Trivial solution

Proposition 1 provides a trivial solution to problem 1 by choosing  $y_4 = 1$  and thus  $y_3 = \frac{f_1}{f_2}$ .

### 3.3. Solution based on the LU factorization

Proposition 2 allows us to look for  $\mathbf{Y}$  in the form:

$$\mathbf{Y} = \begin{pmatrix} 1 & u \\ l & lu + 1 \end{pmatrix}. \quad (37)$$

Then by identifying (29) and (37) we obtain immediately:

$$\begin{cases} y_4 = 1 + y_2 y_3 \\ y_3 = \frac{f_1 e_2}{f_2 e_2 - f_1 e_1} \end{cases} \quad (38)$$

### 3.4. Solution based on the RQ factorization

Proposition 3 allows us to look for  $\mathbf{Y}$  in the form:

$$\mathbf{Y} = \frac{1}{\sqrt{1 + |q|^2}} \begin{pmatrix} 1 + qr & r - \bar{q} \\ q & 1 \end{pmatrix} \quad (39)$$

$$= \frac{1 + qr}{\sqrt{1 + |q|^2}} \begin{pmatrix} 1 & \frac{r - \bar{q}}{1 + qr} \\ \frac{q}{1 + qr} & \frac{1}{1 + qr} \end{pmatrix}. \quad (40)$$

Identifying (29) and (40) yields :

$$\frac{1}{y_4} = 1 + \frac{y_3 y_2}{y_4^2} + \left| \frac{y_3}{y_4} \right|^2 \quad (41)$$

and finally:

$$\begin{cases} y_4 = \frac{|f_2|^2 (f_2 e_2 - f_1 e_1)}{f_2 e_2 (|f_1|^2 + |f_2|^2)} \\ y_3 = \frac{f_1}{f_2} y_4 \end{cases} \quad (42)$$

3.5. *Solution based on the algebraic RQ/LQ factorization*

If we have  $\left(\frac{f_1}{f_2}\right)^2 \neq -1$ , proposition 4 allows us to look for  $\mathbf{Y}$  in the form:

$$\mathbf{Y} = \frac{1}{\sqrt{1+q^2}} \begin{pmatrix} 1+qr & r-q \\ q & 1 \end{pmatrix} \quad (43)$$

$$= \frac{1+qr}{\sqrt{1+q^2}} \begin{pmatrix} 1 & \frac{r-q}{1+qr} \\ \frac{q}{1+qr} & \frac{1}{1+qr} \end{pmatrix}. \quad (44)$$

Identifying (29) and (44) yields :

$$\frac{1}{y_4} = 1 + \frac{y_3 y_2}{y_4^2} + \left(\frac{y_3}{y_4}\right)^2 \quad (45)$$

and finally:

$$\begin{cases} y_4 = \frac{f_2(f_2 e_2 - f_1 e_1)}{e_2(f_1^2 + f_2^2)} \\ y_3 = \frac{f_1}{f_2} y_4 \end{cases} \quad (46)$$

It is noteworthy that in the real case this solution is equivalent to the previous one. Furthermore if  $\left(\frac{f_1}{f_2}\right)^2 = -1$  but  $\left(\frac{e_1}{e_2}\right)^2 \neq -1$  one can still use the LQ factorization of proposition 5 leading to:

$$y_4 = \frac{f_2(e_1^2 + e_2^2)}{e_2(e_2 f_2 - e_1 f_1)}. \quad (47)$$

3.6. *Solution based on the algebraic polar decomposition*

If we have  $\left(\frac{e_1}{e_2}\right)^2 \neq -1$  or  $\left(\frac{e_1}{e_2}\right)^2 = -1$  and  $\left(\frac{f_1}{f_2}\right)^2 = -1$ , proposition 6 allows us to look for  $\mathbf{Y}$  in the form:

$$\mathbf{Y} = \frac{1}{\sqrt{(1-h^2)(1+q^2)}} \begin{pmatrix} 1+qh & h-q \\ h+q & 1-qh \end{pmatrix} \quad (48)$$

$$= \frac{1+qh}{\sqrt{(1-h^2)(1+q^2)}} \begin{pmatrix} 1 & \frac{h-q}{1+qh} \\ \frac{h+q}{1+qh} & \frac{1-qh}{1+qh} \end{pmatrix}. \quad (49)$$

Identifying (29) and (49) yields :

$$y_4^2 + y_4 + \frac{(y_3^2 - y_2^2)}{2} - 2 = 0 \quad (50)$$

and finally:

$$\begin{cases} (2f_2^2 e_2^2 + f_1^2 e_2^2)y_4^2 + 2e_2^2 f_2^2 y_4 - f_2^2 (e_1^2 + 4e_2^2) = 0 \\ y_3 = \frac{f_1}{f_2} y_4. \end{cases} \quad (51)$$

Since  $y_3$  should be close to zero at the convergence, we can choose the root of the binomial that leads to the smallest modulus of  $y_3$ .

These solutions define a class of JEVD algorithms called JAPAM (Joint eigenvalue decomposition Algorithms using a PArameterized Matrix) sharing the same structure. In the following, we will call JAPAM-1, JAPAM-2, JAPAM-3, JAPAM-4 and JAPAM-5 the algorithms based on the parametrizations proposed in section 3.2, 3.3, 3.4, 3.5 and 3.6 respectively. The structure of JAPAM algorithms is summed up in algorithm 1. We clearly see that we can switch from one JAPAM algorithm to another by changing only the expression of  $y_4$  and it is worth mentioning that this can even be done during the iterative procedure. This feature will not be studied here.

One should note that in JAPAM-5 matrix  $\mathbf{Y}$  is decomposed as the product of a complex symmetric matrix and a complex orthogonal matrix. We recall that a complex symmetric matrix is any complex matrix  $\mathbf{H}$  verifying  $\mathbf{H}^T = \mathbf{H}$  and a complex orthogonal matrix is any complex matrix  $\mathbf{Q}$  verifying  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$ . This decomposition is thus a particular case of the algebraic polar decomposition introduced and studied in [20, 21]. In the present context, it is more appropriate than the usual polar decomposition (defined as the product of an Hermitian matrix and a unitary matrix). Indeed the algebraic polar decomposition does not involve product and cross product between the parameters and their complex conjugates. This allows their computation in a very simple way as it has been shown in section 3.6. Moreover, equations are the same in the real and complex cases. The same reasoning has led to the factorizations used for JAPAM-4 that involve the product of a triangular matrix and a complex orthogonal matrix. In the best of our knowledge, we found no mention of such a decomposition so far, so we propose to refer to it as an algebraic RQ or LQ decomposition. The counter part of these simplified algebraic decompositions is that they do not

always exist. However, the existence conditions, given in 3.4, 3.5 and 3.6, are very weak in practice.

Finally we can notice that JAPAM algorithms work with complex and real data without changes. This is an important feature of the proposed approach in comparison with the existing algorithms.

---

**Algorithm 1** The JAPAM algorithms

---

Let  $S_C$  be a stopping criterion and  $It_{max}$  the maximal number of iterations;  
Initialize  $\mathbf{B}$  with  $\mathbf{I}$  or any clever choice;  
 $it \leftarrow 1$ ;  
**while**  $S_C$  is *false* and  $it < It_{max}$  **do**  
  **for**  $i = 1$  to  $N - 1$  **do**  
    **for**  $j = i + 1$  to  $N$  **do**  
      Compute  $\mathbf{E}_{12}^H \mathbf{E}_{12}$  and  $\mathbf{E}_{21}^H \mathbf{E}_{21}$  by using (23) and (24);  
      Compute the eigenvalue decomposition of  $\mathbf{E}_{12}^H \mathbf{E}_{12}$  and  $\mathbf{E}_{21}^H \mathbf{E}_{21}$  to find  $\mathbf{e}$  and  $\mathbf{f}$ ;  
      Compute  $y_2 = e_1/e_2$ ;  
      Compute  $y_4$  according to one of the proposed solutions and deduce  $y_3 = y_4 f_1/f_2$ ;  
      Compute  $\mathbf{X}^{(i,j)}$  from (7) with  $x_1^{(i,j)} = \frac{1}{\sqrt{y_4 - y_2 y_3}}$ ,  $x_2^{(i,j)} = \frac{y_2}{\sqrt{y_4 - y_2 y_3}}$ ,  
 $x_3^{(i,j)} = \frac{y_3}{\sqrt{y_4 - y_2 y_3}}$  and  $x_4^{(i,j)} = \frac{y_4}{\sqrt{y_4 - y_2 y_3}}$ ;  
      Update  $\mathbf{B} \leftarrow \mathbf{X}^{(i,j)} \mathbf{B}$ ;  
      Update  $\hat{\mathbf{D}}^{(k)} \leftarrow \mathbf{X}^{(i,j)} \hat{\mathbf{D}}^{(k)} (\mathbf{X}^{(i,j)})^{-1} \forall k \in [1, K]_{\mathbb{N}}$ ;  
    **end for**  
  **end for**  
  Update  $S_C$   
   $it \leftarrow it + 1$ ;  
**end while**

---

#### 4. Numerical complexity

We define here the numerical complexity as the number of real multiplications computing during one iteration. We consider that a complex multiplication is equivalent to four real multiplications (even if it could be done in only three real multiplications). Numerical complexity of JEVD algorithms resorting to block coordinate procedure are dominated by the updating step (equation (8)). Therefore, the numerical complexity of existing polar decomposition based algorithms is about  $32KN^3$  for SH-RT and JDTM and  $44KN^3$  for JUST. The proposed approach allows to reduce this number to only  $16KN^3$ . One can argue that JET-O and JET-U still have the lowest complexity with  $8KN^3$  and  $4KN^3$ . However it has been shown in [8] that, in the complex case, these algorithms are less efficient than the others in the medium-low SNR range. This will be confirmed in the next section.

#### 5. Numerical Simulations

We now compare the performances the JAPAM approach with those of references JEVD algorithms: SH-RT, JDTM, JUST, JET-U and JET-O. The three first algorithms define the family of polar decomposition based algorithms. In the comparisons the JAPAM family will be represented by JAPAM-2, JAPAM-4 and JAPAM-5 algorithms. Indeed preliminary studies were led (results not shown) and revealed that they consistently perform better than JAPAM-1 and JAPAM-3. The set of matrices to decompose is built from complex valued matrices. In order to evaluate the accuracy of the eigenvector matrix estimation, we use an indicator which quantifies the relative deviation between the inverse of the estimated diagonalizing matrix  $\mathbf{B}$  and the actual matrix of eigenvectors  $\mathbf{A}$ . After removing scaling and permutation indeterminacy, the indicator is defined as:

$$r_A = \frac{\|\mathbf{A} - \mathbf{B}^{-1}\|}{\|\mathbf{A}\|}. \quad (52)$$

A second criterion is used to measure the average computational cost of the algorithms. This criterion, called  $\Gamma$ , is the product of the numerical complexity

with the average number of iterations that an algorithm needs to converge.

We study the algorithms behavior according to two scenarios. In the first one, we focus on the impact of the noise level. In the second one, we study the impact of the matrix size. For both scenarios, we build the set of matrices as follow:

$$\mathbf{M}^{(k)} = \frac{\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}}{\|\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}\|} + \sigma \frac{\mathbf{E}^{(k)}}{\|\mathbf{E}^{(k)}\|}, \quad \forall k = 1, \dots, K. \quad (53)$$

Where matrices  $\mathbf{A}$ ,  $\mathbf{D}^{(k)}$  and  $\mathbf{E}^{(k)} \in \mathbb{C}^{N \times N}$  are drawn according to a standard normal distribution and  $\sigma$  is a parameter which allows to set the Signal to Noise Ratio (SNR). We now define the following function:

$$S(\mathbf{B}) = \sum_{k=1}^K \|\text{ZDiag}\{\mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{-1}\}\|^2. \quad (54)$$

All the algorithms are stopped when the criterion  $|S(\mathbf{B}_{it+1}) - S(\mathbf{B}_{it})|$  falls bellow  $10^{-6}$  (where  $\mathbf{B}_{it}$  is the estimated diagonalizing matrix at iteration  $it$ ) or when they have computed 150 iterations. For each value of the varying parameters (SNR value or matrix size), we perform 200 Monte Carlo (MC) runs and for each MC runs, we build a new set of matrices.

### 5.1. Scenario 1

**Settings:** we take a matrix size  $N = 5$ , a number of matrices  $K = 20$  and we vary the SNR values from 0 dB to 100 dB by step of 10 dB. We initialize  $\mathbf{B}$  with the identity matrix.

**Results:** the results of scenario 1 are plotted on figure 1.

JAPAM-5 provides the best estimation of the matrix of eigenvectors from 30 dB to 100 dB, indeed it has the lowest average and median values of  $r_A$  and also the lowest standard deviation (respectively figures 1a, 1b and 1c). Under 30 dB the only reference JEVD algorithm which competes with JAPAM-5 is JD TM. About the average computational cost (figure 1d), it can be seen that JAPAM-5 is about 2 to 10 times less costly than polar decomposition based algorithms (SH-RT, JUST and JD TM).

JAPAM-4 reaches a level of performances close to JAPAM-5 in term of accuracy and stability. However it is more costly, in particular for the intermediate SNR values. Indeed, at 0 dB and 100 dB JAPAM-4 and JAPAM-5 have about the same average computational cost while, for instance, JAPAM-4 is 1.9 times more costly than JAPAM-5 at 30 dB.

It is noteworthy that JAPAM-2 has the same  $r_A$  median value as JAPAM-5 and JAPAM-4 above 30 dB. On the other hand its  $r_A$  average value and its  $r_A$  standard deviation are greater. It means that JAPAM-2 converges more frequently to absurd solutions than JAPAM-5 and JAPAM-4. Thereby JAPAM-2 is not recommended for low SNR values. In addition, JAPAM-2 is costlier than JAPAM-4 and JAPAM-5 (but it remains less costly than the polar decomposition based methods).

Thus we show with this first scenario that the JAPAM method is interesting for diagonalizing a set of small matrices for any tested SNR value at the exception of 0 dB but we can consider that at this value no algorithms give satisfying results. Indeed JAPAM-5 and JAPAM-4 improve the estimation of the matrix of eigenvectors with a limited computational cost and also appears as the stablest algorithms. JET-U is clearly the least costly algorithms but it is also much less accurate and stable than JAPAM-5 and JAPAM-4.

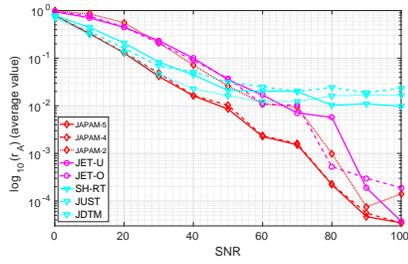
### 5.2. Scenario 2.a

**Settings:** we now set the SNR value to 50 dB, the number of matrices to  $K = 20$  and we vary the matrix size from  $N = 3$  to  $N = 15$  by step of 1. We still initialize  $\mathbf{B}$  with the identity matrix.

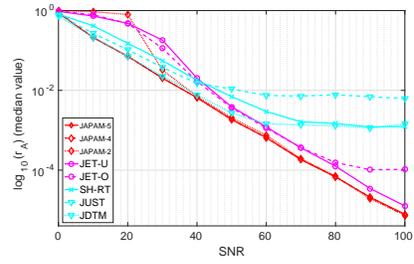
**Results:** results of scenario 2.a are plotted on figure 2.

By observing the  $r_A$  average and the  $r_A$  median value (figures 2a and 2b) of the JAPAM methods, it can be seen that these values dramatically increase from a critical value of  $N$  which depends on the JAPAM version. Actually, in this scenario, we can define a working range for each of the proposed algorithm.

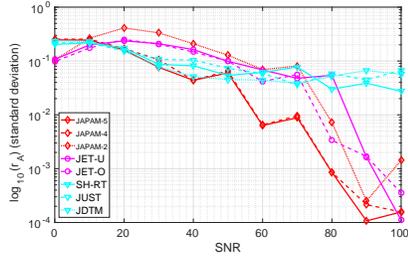
The working range of JAPAM-5 is located under  $N = 13$  (figure 2b). However JAPAM-5 is clearly the most accurate of all the algorithms for  $N < 12$  (figure



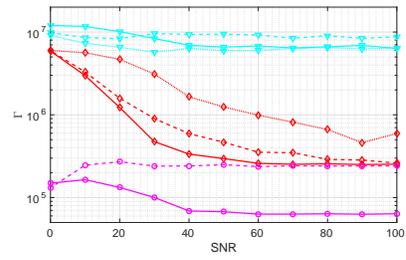
(a) Average value of  $r_A$  w.r.t SNR.



(b) Median value of  $r_A$  w.r.t SNR.



(c) Standard deviation of  $r_A$  w.r.t SNR.



(d)  $\Gamma$  w.r.t SNR.

Figure 1: Algorithm performances computed from 200 MC runs on sets of 20 matrices  $5 \times 5$  (initialization with Identity matrix).

2a). Inside its working range it has a quite good stability. It is notably the stablest for  $N < 7$  (figure 2c). Average computational cost is displayed on figure 2d. Inside its working range, JAPAM-5 is much less costly than the polar decomposition based algorithms but JET-U is still the less costly for any matrix size.

The working range of JAPAM-4, its working range is located under  $N = 10$ . It can be considered as the second best algorithm for  $N < 7$  in term of estimation precision of  $\mathbf{A}$ . For  $N = 3$  and  $N = 4$ , it is as stable as JAPAM-5 then its  $r_A$  standard deviation increases until the end of its working range. JAPAM-4 is less costly than the polar decomposition based algorithms, but it is costlier than JAPAM-5.

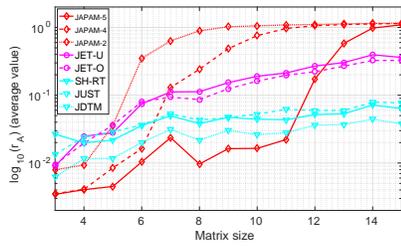
The working range of JAPAM-2 is located under  $N = 7$ . It is competitive with the reference JEVD algorithms for  $N < 6$  but it is one of the least stable algorithms in its working range. Moreover, in this range, the average computational cost of JAPAM-2 is the most important of the JAPAM methods but it is still clearly lower than those of classical polar decomposition algorithms.

Thus in this scenario we show that the JAPAM algorithms are interesting for small and medium matrix sizes in terms of accuracy, stability and computational cost but with this kind of initialization (using the identity matrix) they are not adapted to big matrices. This behaviour may be explained by the combination of the chosen diagonalization criterion and the approximation derived from assumption 1. In the following, we will show that this issue can be corrected easily.

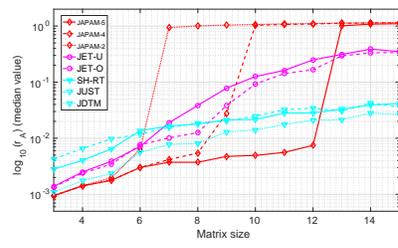
The common framework of the JAPAM methods allows to highlight that the LU factorization is less robust to approximation errors than the algebraic QR factorization which is itself less robust than the algebraic polar decomposition.

### 5.3. Scenario 2.b

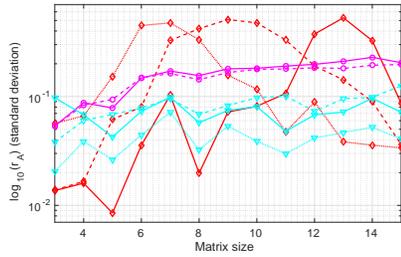
**Settings:** we take the same settings as for scenario 2.a but here we initialize  $\mathbf{B}$  (for all the algorithms) with the results of a General EigenValue Decomposition (GEVD). The GEVD is the exact joint eigenvalue decomposition of two



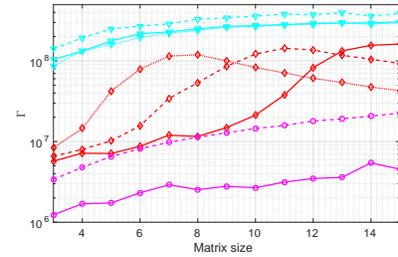
(a) Average value of  $r_A$  w.r.t  $N$ .



(b) Median value of  $r_A$  w.r.t  $N$ .



(c) Standard deviation of  $r_A$  w.r.t  $N$ .



(d)  $\Gamma$  w.r.t  $N$ .

Figure 2: Algorithm performances computed from 200 MC runs on sets of 20 matrices with a SNR value of 50 dB (initialization of  $\mathbf{B}$  with the identity matrix).

matrices, which are here two matrices of the set  $\mathbf{M}^{(k)}$  randomly chosen (but we could choose the two matrices with the lower conditional number). The GEVD helps the algorithms to get closer to the diagonalizing solution.

**Results:** Results of scenario 2.b are plotted on figure 3.

We can notice that, this time, the JAPAM algorithms provide very good results for any matrix size in term of average (figure 3a) and of median (figure 3b) estimation of  $\mathbf{A}$ .

Indeed by observing the  $r_A$  average value and the  $r_A$  median value of JAPAM-5, we can see that it is now the best method for estimating the matrix of eigenvectors for any value of  $N$ . Concerning the standard deviation (figure 3c), we can notice that for  $N < 12$  JAPAM-5 and JD TM are the stablest algorithms, but for  $N \geq 12$  JD TM is more stable. About the average computational cost, the gap between the algorithms is constant. Once again JAPAM-5 is much less costly than SH-RT, JUST and JD TM but costlier than JET-U and JET-O.

Regarding the average value of  $r_A$  JAPAM-4 is as accurate as JAPAM-5 for the estimation of the matrix of eigenvectors for  $N < 5$ . Then it is slightly less accurate. On the other hand, regarding the median value, JAPAM-4 is as good as JAPAM-5 for any matrix size. Concerning the standard deviation of JAPAM-4, we can notice that it provides the same results as JAPAM-5 for  $N < 6$ , then it is outperformed by JAPAM-5 and by the polar decomposition based algorithms. In conclusion, JAPAM-4 is almost as accurate as JAPAM-5 but it is less stable. Its average computational cost is also slightly more important than the one of JAPAM-5.

Finally JAPAM-2 is as accurate as JAPAM-5 and JAPAM-4 on median. But it is not as good regarding the average value for  $N > 5$ . Its  $r_A$  standard deviation is similar to the one of JAPAM-5 and JAPAM-4 for  $N < 6$ , then it competes only with JET-O and JET-U which are the least stable algorithms. So JAPAM-2 is globally as accurate as JAPAM-4 and JAPAM-5, but it is less stable. The average computational cost of JAPAM-2 is slightly higher than those of JAPAM-4.

Thus, we have shown that initializing our problem with a GEVD significantly

improves the performances of the coupled algorithms for the largest matrix sizes. This makes of JAPAM-5 a suitable solution if we look for the algorithm that achieves the best performances in terms of estimation of the matrix of eigenvectors, whatever the considered combination of SNR and matrix sizes. Furthermore its average computational cost is limited with respect to all other algorithms at the exception of JET-O and JET-U.

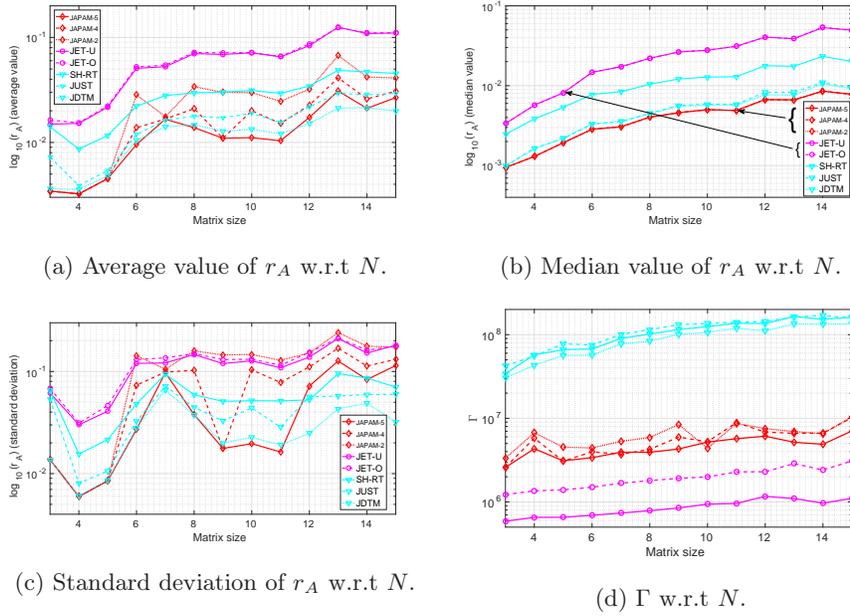


Figure 3: Algorithm performances computed from 200 MC runs on sets of 20 matrices with a SNR value of 50 dB (initialization of  $\mathbf{B}$  with a GEVD).

## 6. Application example in numerical telecommunications

As mentioned in introduction of this paper, canonical polyadic decomposition of tensors is one the main motivation of JEVD algorithms since various authors showed that the CPD can be rewritten as a JEVD problem, giving birth to fast CPD algorithms in comparison to classical iterative approaches.

Complex-valued tensor and CPD are commonly used in digital telecommunications for blind source separation purpose. In particular, in [22] Sidiropoulos *et al* showed how a Direct-Sequence Code Division Multiple Access (DS-CDMA) system can be modeled by using a Canonical Polyadic Decomposition (CPD or PARAFAC) model and then proposed a "PARAFAC DS-CDMA receiver" that allows to estimate the source signals in a deterministic way. Indeed, in the absence of intersymbol-interference the complex envelop of the source signals appear as the columns of one of the CPD factor matrices. Since this pioneer work, deterministic tensor approaches have been successfully applied to several DS-CDMA devices [23, 24, 25, 26]. In this short section, we want to evaluate the suitability of the JAPAM-5 algorithm to address this kind of problem. In this purpose we simulate the following DS-CDMA system of transmission: At the emission,  $N$  users transmit at the same time sequences of  $P$  QPSK symbols spread by hadamar codes of  $Q$  symbols. At the reception, we consider an array of  $R$  antennas and the transmission canal is modeled by  $R$  linear mixtures of the  $N$  emitted signals. A white Gaussian noise is added to the mixed signals. Thus each receiving antenna receives a sequence of  $PQ$  symbols. After shifting and downsampling operation, the collected symbols are gathered in a  $P \times Q \times R$  complex-valued tensor  $\mathcal{X}$  and the rank  $N$  CPD of  $\mathcal{X}$  gives:

$$\mathcal{X}_{pqr} = \sum_{n=1}^N S_{pn} C_{qn} H_{rn} + \mathcal{E}_{pqr} \quad (55)$$

where  $S_{pn}$  is the  $p^{th}$  symbol of source  $n$ ,  $C_{qn}$  is the  $q^{th}$  symbol of code sequence  $n$ ,  $H_{rn}$  is the  $n^{th}$  coefficient of the linear mixture received by antenna  $r$  and tensor  $\mathcal{E}$  represents the noise. Hence, CPD factors matrices  $\mathbf{S}$ ,  $\mathbf{C}$  and  $\mathbf{H}$  define the source, code and mixture matrices respectively and can be estimated by the rank  $N$  CPD of  $\mathcal{X}$ . In the following simulation we compare the performances of 3 CPD algorithms: ALS, ALS+ELSCS<sup>1</sup> and DIAG-JAPAM-5 which is a version of the DIAG algorithm [11] that use the proposed JAPAM-5 algorithm

---

<sup>1</sup>we used the matlab code provided at the author web-page:  
[http://dimitri.nion.free.fr/Codes/Tensor\\_Decompositions.html](http://dimitri.nion.free.fr/Codes/Tensor_Decompositions.html)

to compute the JEVD step. ALS (Alternating Least Squares) is still one of the most popular CPD algorithm. An Enhanced Line Search procedure (ELS) has been proposed in [27] for real-value tensor CPD in order to improve ALS performances. This procedure was then extended to complex-valued tensors in [28], giving birth to the ALS+ELSCS algorithm. It is worth mentioning that in the same paper both algorithms were successfully applied for blind separation of DS-CDMA signals. We propose here a simple test scenario that illustrates the interest of the proposed method. We choose  $N = 7$ ,  $P = 512$ ,  $Q = 16$  and  $R = 5$  so that we have less receiving antenna (4) than emitting antenna (7) and we vary the signal to noise ratio of the system from 0 to 12 dB. Comparisons are done by means of MC simulations: for each SNR value 200 test tensors are built according to equation (55) from 200 random draws of  $\mathbf{S}$ ,  $\mathbf{H}$  and  $\mathcal{E}$ . More precisely, each column of  $\mathbf{S}$  is built as the QPSK modulation of a random binary signal of length  $2P$ . Test tensors are then decomposed using the three algorithms. ALS and ALS+ELS algorithms are initialized randomly and the JEVD step of DIAG-JAPAM-5 is initialized by the identity matrix. Our first comparison criteria is the average value over 200 MC runs of the Binary Error Ratio (BER) computed after demodulation from the source matrix estimated by each algorithm. Results are plotted on figure 4. It clearly appears that the best results are obtained with DIAG-JAPAM-5, whatever the SNR value. On figure 5, we plot the average value of the normalized mean squared error, denoted  $r_H$ , between the actual mixing matrix and the mixing matrices estimated by the three algorithms. Again, the best estimates of the mixing matrix are consistently provided by DIAG-JAPAM-5. Actually, in this particular scenario iterative algorithm ALS and ALS+ELS algorithms provide poor average results. This is mainly explained by the sensibility of algorithms to the random initialisation, especially when the rank of the CPD is high and this is the case here.

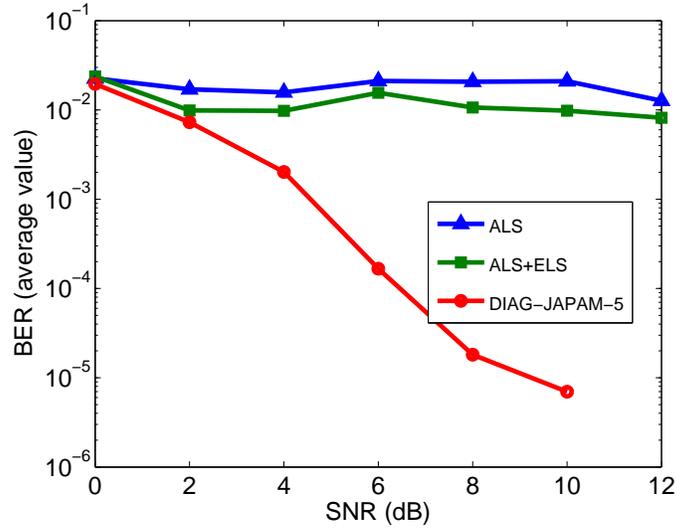


Figure 4: Evolution of the average value of the BER with the SNR (at 12 dB the value obtained with DIAG-JAPAM-5 is zero)

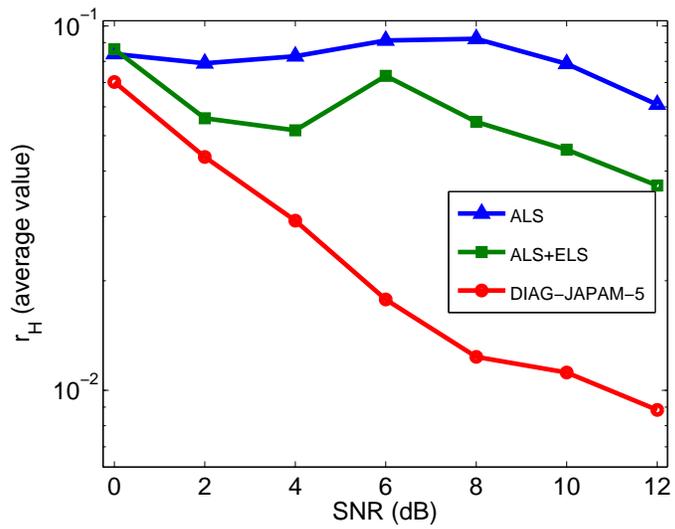


Figure 5: Evolution of the average estimation error of the mixing matrix ( $r_H$ ) with the SNR

## 7. Conclusion

Thus we introduced a new class of algorithms to compute the joint eigenvalue decomposition of a set of matrices. Five algorithms that share a common structure are described: JAPAM-1, JAPAM-2, JAPAM-3, JAPAM-4 and JAPAM-5. In comparison with existing algorithms, the proposed approach offers several immediate advantages. First of all, it allows to deal with real or complex matrix sets without change. Second, our numerical simulations show that JAPAM-5 and JAPAM-4 are the most accurate and the stablest algorithms to decompose a set of small complex-valued matrices, whatever the SNR value. Finally, in terms of computational cost JAPAM-5 and JAPAM-4 are usually much less costly than polar decomposition based algorithms. JET algorithms have a lower computational cost but are significantly less accurate and stable.

Numerical simulations also highlighted that when the JAPAM algorithms are initialized with the identity matrix, these are very sensitive to the matrix size. This is particularly true for JAPAM-2 and JAPAM-4. However we then show that this issue can be easily circumvented by using a generalized EVD for initialization. Indeed the more we increase the size of the matrices the more JAPAM-2, JAPAM-4 and JAPAM-5 takes benefit of the GEVD so that finally JAPAM-4 and JAPAM-5 provide the best results of all the tested algorithms whatever the evaluated matrix sizes.

Globally, JAPAM-5 appears as the best of the JAPAM algorithms, whatever the tested scenario: the performance criterion or the initialization. It is followed by JAPAM-4 and JAPAM-2. The gap between the JAPAM algorithms is reduced when using a GEVD for initialization. The only difference between the proposed algorithms is the used matrix factorization. Therefore, if a more general conclusion had to be drawn, this study would advocate for the use of the polar decomposition for joint diagonalization problems.

The present work also includes a practical application of JAPAM-5 for the blind source separation of telecommunication signals. Indeed we have shown that JAPAM-5 can be efficiently used for the canonical polyadic decomposition

of complex DS-CDMA signals with respect to classical approaches. Finally, one interesting feature of the JAPAM approach is that the algorithms can be interchanged during the iterative process. In other word, one iteration of an algorithm can be followed by an iteration of another one then by one iteration of a third one and so on. This point has not been exploited here and investigated in future works.

### Appendix A. Proof of propositions 4, 5 and 6

*Proof of proposition of 4.* Equation (34) is equivalent to

$$\tilde{\mathbf{X}} \begin{pmatrix} 1 & q \\ -q & 1 \end{pmatrix} \begin{pmatrix} 1 & -r \\ 0 & 1 \end{pmatrix} \frac{1}{\sqrt{1+q^2}} = \mathbf{\Lambda}. \quad (\text{A.1})$$

Thereby, we have to find a necessary and sufficient condition for which the following system in  $q$  and  $r$  has a solution:

$$\begin{cases} \tilde{X}_{1,1}(q-r) + \tilde{X}_{1,2}(1+qr) = 0 \\ \tilde{X}_{2,1} - q\tilde{X}_{2,2} = 0 \\ 1+q^2 \neq 0. \end{cases} \quad (\text{A.2})$$

Recalling that  $\tilde{X}_{2,2} \neq 0$  and  $\det(\tilde{\mathbf{X}}) = \tilde{X}_{1,1}\tilde{X}_{2,2} - \tilde{X}_{2,1}\tilde{X}_{1,2} = 1$ , the system is solved by  $q = \tilde{X}_{2,1}/\tilde{X}_{2,2}$  and  $r = \frac{\tilde{X}_{2,2}\tilde{X}_{1,2} + \tilde{X}_{1,1}\tilde{X}_{2,1}}{\tilde{X}_{1,1}\tilde{X}_{2,2} - \tilde{X}_{2,1}\tilde{X}_{1,2}}$  if and only if  $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}})^2 \neq -1$ . Obviously, if  $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}})^2 \neq -1$ ,  $\mathbf{\Lambda}$  being a product of invertible matrices, it is invertible too.  $\square$

Proposition 5 can be proved in the same way.

*Proof of proposition 6.* We use the same reasoning as for proposition 4: thus we show that it exists  $h$  and  $q$  such that the matrix

$$\tilde{\mathbf{X}} \begin{pmatrix} 1 & q \\ -q & 1 \end{pmatrix} \begin{pmatrix} 1 & -h \\ -h & 1 \end{pmatrix} \frac{1}{\sqrt{(1-h^2)(1+q^2)}}$$

is diagonal and invertible. In other words we want to check that if  $(\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}})^2 \neq -1$  and  $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}})^2 \neq -1$ , then the following system has a solution:

$$\begin{cases} \tilde{X}_{1,2}qh + \tilde{X}_{1,1}q - \tilde{X}_{1,1}h + \tilde{X}_{1,2} = 0 \\ -\tilde{X}_{2,1}qh - \tilde{X}_{2,2}q - \tilde{X}_{2,2}h + \tilde{X}_{2,1} = 0 \\ 1 + q^2 \neq 0 \\ 1 - h^2 \neq 0. \end{cases} \quad (\text{A.3})$$

We distinguish four cases:

**case 1.**  $\tilde{X}_{1,2} = \tilde{X}_{2,1} = 0$

This case is trivial since  $\tilde{\mathbf{X}}$  is already an invertible diagonal matrix so  $q = h = 0$  is a solution.

**case 2.**  $\tilde{X}_{1,2} = 0$  and  $\tilde{X}_{2,1} \neq 0$

One can easily check that in this case,  $q = h$  with  $q$  verifying:

$$-\tilde{X}_{2,1}q^2 - 2\tilde{X}_{2,2}q + \tilde{X}_{2,1} = 0. \quad (\text{A.4})$$

This polynomial has at least one solution. Then,  $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}})^2 \neq -1$  implies  $q^2 \neq -1$  and  $\tilde{X}_{2,2} \neq 0$  implies  $h^2 \neq 1$ .

**case 3.**  $\tilde{X}_{1,2} \neq 0$  and  $\tilde{X}_{2,1} = 0$

The same reasoning leads to the following conclusion:  $q = -h$  and  $q$  is a solution of

$$-\tilde{X}_{1,2}q^2 + 2\tilde{X}_{1,1}q + \tilde{X}_{1,2} = 0. \quad (\text{A.5})$$

Then  $(\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}})^2 \neq -1$  implies  $q^2 \neq -1$  and  $\tilde{X}_{1,1} \neq 0$  implies  $h^2 \neq 1$ .

**case 4.**  $\tilde{X}_{1,2} \neq 0$  and  $\tilde{X}_{2,1} \neq 0$

Here we have:

$$\begin{cases} \alpha q - \beta h + \gamma = 0 \\ \beta h^2 + 2h - \beta = 0 \\ q^2 \neq -1 \\ h^2 \neq 1. \end{cases} \quad (\text{A.6})$$

with

$$\begin{cases} \alpha = \tilde{X}_{2,1}\tilde{X}_{1,1} - \tilde{X}_{2,2}\tilde{X}_{1,2} \\ \beta = \tilde{X}_{1,1}\tilde{X}_{2,1} + \tilde{X}_{2,2}\tilde{X}_{1,2} \\ \gamma = 2\tilde{X}_{2,1}\tilde{X}_{1,2}. \end{cases}$$

We have four subcases according to the value of  $\alpha$  and  $\beta$ :

1.  $\alpha = 0$  and  $\beta = 0$

Impossible because  $\tilde{X}_{2,1} \neq 0$  and  $\tilde{X}_{1,2} \neq 0$ .

2.  $\alpha \neq 0$  and  $\beta = 0$

Here, a solution is given by  $q = -\frac{\gamma}{\alpha} = -\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}}$  and  $h = 0$  thus if  $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{1,1}})^2 \neq -1$  then  $q^2 \neq -1$ .

3.  $\alpha = 0$  and  $\beta \neq 0$

Here, a solution is given by  $q = 0$  and  $h = \frac{\gamma}{\beta} = \frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}} = \frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}}$ . We can then easily verify that  $h^2 = 1$  is incompatible with  $\det\{\tilde{\mathbf{X}}\} = 1$ .

4.  $\alpha \neq 0$  and  $\beta \neq 0$

Here it is clear that  $h \neq 1$  and  $h \neq -1$  (otherwise the second equation in (A.6) would be false). The first equation of (A.6) gives  $h = \frac{\alpha q + \gamma}{\beta}$ , reporting this expression in the second equation yields:

$$\alpha^2 q^2 + (2\alpha\gamma + 2\alpha)q + \gamma^2 + 2\gamma - \beta^2 = 0 \quad (\text{A.7})$$

which has at least one solution. After some straightforward computations, we can show that:

$$q^2 \neq -1 \Leftrightarrow \alpha \neq \pm i(1 + \gamma) \quad (\text{A.8})$$

$$\text{and that } \begin{cases} (\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}})^2 \neq -1 \\ (\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}})^2 \neq -1 \end{cases} \Rightarrow \alpha \neq \pm i(1 + \gamma).$$

□

## References

- [1] A. J. Van der Veen, P. B. Ober, E. F. Deprettere, Azimuth and elevation computation in high resolution doa estimation, *IEEE Transactions on Signal Processing* 40 (7) (1992) 1828–1832.

- [2] A. N. Lemma, A. J. Van der Veen, E. F. Deprettere, Analysis of joint angle-frequency estimation using ESPRIT, *IEEE Transactions on Signal Processing* 51 (5) (2003) 1264–1283.
- [3] M. Haardt, J. Nossek, Simultaneous schur decomposition of several non-symmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems, *IEEE Transactions on Signal Processing* 46 (1) (1998) 161–169.
- [4] J. F. Cardoso, A. Souloumiac, Jacobi angles for simultaneous diagonalization, *SIAM Journal Matrix Analysis and Applications* 17 (1) (1996) 161–164.
- [5] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, E. Moulines, A blind source separation technique using second-order statistics, *IEEE Transactions on Signal Processing* 45 (2) (1997) 434–444.
- [6] E. Moreau, A generalization of joint-diagonalization criteria for source separation, *IEEE on Signal Processing* 49 (3) (2001) 530–541.
- [7] A. Yeredor, Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation, *IEEE Transactions on Signal Processing* 50 (7) (2002) 1545–1553.
- [8] 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.
- [9] 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.
- [10] F. Roemer, M. Haardt, A closed-form solution for multilinear parafac decompositions, in: *SAM 08, Fifth IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2008, pp. 487–491.

- [11] X. Luciani, L. Albera, Canonical polyadic decomposition based on joint eigenvalue decomposition, *Chemometrics and Intelligent Laboratory Systems* 132 (0) (2014) 152 – 167.
- [12] Y. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1982.
- [13] C. G. J. Jacobi, Uber ein leichtes verfahren die in der theorie der secularstorungen vorkommenden gleichungen numerisch aufzulosen, *Journal fur die reine und angewandte Mathematik* 1846 (30) (1846) 51–94.
- [14] P. Strobach, Bi-iteration multiple invariance subspace tracking and adaptive esprit, *IEEE Transactions on Signal Processing* 48 (2000) 442–456.
- [15] 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.
- [16] 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.
- [17] V. Maurandi, E. Moreau, C. D. Luigi, Jacobi like algorithm for non-orthogonal joint diagonalization of hermitian matrices, in: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2014)*, Florence, Italy, 2014.
- [18] R. Andre, X. Luciani, E. Moreau, A coupled joint eigenvalue decomposition algorithm for canonical polyadic decomposition of tensors, in: *Sensor Array and Multichannel signal processing workshop (SAM’2016)*, Rio de Janeiro, Brazil, 2016.
- [19] A. Souloumiac, Nonorthogonal joint diagonalization by combining givens and hyperbolic rotations, *IEEE Transactions on Signal Processing* 57 (6) (2009) 2222–2231.

- [20] D. Choudhury, R. Horn, A complex orthogonal-symmetric analog of the polar decomposition, *SIAM J. Algebraic Discrete Methods* 8 (1987) 219–225.
- [21] I. Kaplansky, Algebraic polar decomposition, *SIAM Journal on Matrix Analysis and Applications* 11 (2) (1990) 213–217.
- [22] 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.
- [23] D. Nion, L. De Lathauwer, A Block Component Model based Blind DS-CDMA Receiver, *IEEE Trans. Signal Proc.* 56 (11) (2008) 5567–5579.
- [24] A. de Almeida, G. Favier, J. Mota, Space-time spreading mimo-cdma downlink systems using constrained tensor modeling, *Signal Processing* 88 (10) (2008) 2403 – 2416.
- [25] M. da Costa, A. de Almeida, G. Favier, J. Romano, Tensor Coding for CDMA-MIMO Wireless Communication Systems, in: *19th European Signal Processing Conference (EUSIPCO)*, Barcelone, Spain, 2011, pp. 101–105.
- [26] A. de Almeida, A. Kibangou, S. Miron, D. Araùjo, Joint data and connection topology recovery in collaborative wireless sensor networks, in: *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, 2013, pp. 5303–5307.
- [27] M. Rajih, P. Comon, R. Harshman, Enhanced line search: A novel method to accelerate PARAFAC, *SIAM Journal on Matrix Analysis and Applications*.
- [28] D. Nion, L. D. Lathauwer, An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA, *Signal Processing, Elsevier* 88 (3) (2008) 749–755.