



HAL
open science

Nonnegative Joint Diagonalization by Congruence Based on LU Matrix Factorization

Lu Wang, Laurent Albera, Amar Kachenoura, Huazhong Shu, Lotfi Senhadji

► **To cite this version:**

Lu Wang, Laurent Albera, Amar Kachenoura, Huazhong Shu, Lotfi Senhadji. Nonnegative Joint Diagonalization by Congruence Based on LU Matrix Factorization. *IEEE Signal Processing Letters*, 2013, 20 (8), pp.807 - 810. 10.1109/LSP.2013.2267797 . inserm-00841128

HAL Id: inserm-00841128

<https://inserm.hal.science/inserm-00841128v1>

Submitted on 7 Jul 2013

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.

Nonnegative Joint Diagonalization by Congruence Based on LU Matrix Factorization

Lu Wang, *Student Member, IEEE*, Laurent Albera, *Senior Member, IEEE*, Amar Kachenoura, Huazhong Shu, *Senior Member, IEEE*, and Lotfi Senhadji, *Senior Member, IEEE*

Abstract—In this letter, a new algorithm for joint diagonalization of a set of matrices by congruence is proposed to compute the nonnegative joint diagonalizer. The nonnegativity constraint is imposed by means of a square change of variables. Then we formulate the high-dimensional optimization problem into several sequential polynomial subproblems using LU matrix factorization. Numerical experiments on simulated matrices emphasize the advantages of the proposed method, especially in the case of degeneracies such as for low SNR values and a small number of matrices. An illustration of blind separation of nuclear magnetic resonance spectroscopy confirms the validity and improvement of the proposed method.

Index Terms—Blind source separation, independent component analysis, LU factorization, nonnegative joint diagonalization by congruence, nuclear magnetic resonance spectroscopy.

I. INTRODUCTION

Consider a set $\mathcal{C} = \{\mathbf{C}^{(k)}\}_{k=1}^K$ of K real symmetric matrices sharing the common structure:

$$\mathbf{C}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^\top \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{N \times P}$ denotes an unknown transformation matrix, and $\mathcal{D} = \{\mathbf{D}^{(k)}\}_{k=1}^K$ is a set of $(P \times P)$ unknown real diagonal matrices. \mathbf{A} is called a joint diagonalizer of \mathcal{C} . According to different applications, the K matrices $\mathbf{C}^{(k)}$ can be time-delayed covariance matrices, or higher order cumulant matrix slices. The objective of Joint Diagonalizing by Congruence (JDC) such matrices consists of estimating \mathbf{A} up to a scale factor and a permutation of columns. Such a problem appears in many signal processing contexts such as Blind Source Separation (BSS) and Independent Component Analysis (ICA). Some BSS applications involve a nonnegative mixing matrix \mathbf{A} . For instance, in Nuclear Magnetic Resonance

(NMR) spectroscopy, columns of \mathbf{A} represent the positive concentration of source metabolites [13]. Therefore we propose to impose a nonnegativity constraint on the matrix \mathbf{A} in the JDC problem. Generally, \mathbf{A} can be estimated either indirectly or directly [4]:

- 1) Indirect algorithms, such as JAD [3], FFDIAG [22], QDIAG [18], LUJ1D [1], FLEXJD [21], J-DI [14] and CVFFDIAG [19], estimate \mathbf{A} from the inverse of a transformation matrix \mathbf{V} , which minimizes the non-diagonal parts of $\mathbf{V}\mathbf{C}^{(k)}\mathbf{V}^\top$ using the following criterion:

$$U(\mathbf{V}) = \sum_{k=1}^K \left\| \mathbf{V}\mathbf{C}^{(k)}\mathbf{V}^\top - \text{diag}(\mathbf{V}\mathbf{C}^{(k)}\mathbf{V}^\top) \right\|_F^2 \quad (2)$$

where $\text{diag}(\cdot)$ eliminates the non-diagonal parts of its input and $\|\cdot\|_F$ denotes the Frobenius norm. These algorithms use additional constraint on \mathbf{V} , such as orthogonality [3] or unit-determinant [1], in order to avoid the trivial solution $\mathbf{0}$. There also exist some other criterions. For example, LUJ2D [1] minimizes a scale and permutation invariant criterion. An information theoretic criterion is proposed by Pham [12] and adopted by Joho [8] and Todros *et al.* [16], which requires the target set \mathcal{C} to be positive definite. However, the nonnegativity of \mathbf{A} can be hardly guaranteed in such approaches.

- 2) ACDC [20] and the subspace fitting algorithm [17] directly estimate \mathbf{A} and the set \mathcal{D} in order to best approximate the target set \mathcal{C} by minimizing the following criterion:

$$J(\mathbf{A}, \mathcal{D}) = \sum_{k=1}^K \left\| \mathbf{C}^{(k)} - \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^\top \right\|_F^2 \quad (3)$$

Herein the nonnegativity of \mathbf{A} can be enforced explicitly.

In this letter, we present a nonnegative JDC algorithm based on criterion (3), which is committed to seek a nonnegative joint diagonalizer \mathbf{A} . The nonnegativity constraint is imposed by means of a *square change of variable*. Then the high-dimensional optimization problem is formulated into several sequential polynomial subproblems using LU matrix factorization. Numerical experiments on simulated matrices emphasize the advantages of the proposed method, especially in the case of degeneracies such as for low Signal to Noise Ratio (SNR) values and a small number K of matrices. An illustration of blind separation of NMR spectroscopy confirms the validity and improvement of the proposed method.

II. NONNEGATIVE JOINT DIAGONALIZATION

A way of including the nonnegativity constraint in (3) is through a square change of variable $\mathbf{A} = \mathbf{B}^{\square 2}$ with $\mathbf{B} \in \mathbb{R}^{N \times P}$ as originally proposed in [5] for Nonnegative Matrix Factorization (NMF), where \square stands for Hadamard product. Then (3) can be reformulated as the following unconstrained problem:

$$J(\mathbf{B}, \mathcal{D}) = \sum_{k=1}^K \left\| \mathbf{C}^{(k)} - (\mathbf{B}^{\square 2})\mathbf{D}^{(k)}(\mathbf{B}^{\square 2})^\top \right\|_F^2 \quad (4)$$

Manuscript received April 02, 2013; revised June 02, 2013; accepted June 04, 2013. Date of publication June 12, 2013; date of current version June 18, 2013. The associate editor coordinating the review of this story and approving it for publication was Prof. Arrate Munoz-Barrutia.

L. Wang, A. Kachenoura and L. Senhadji are with the INSERM, U1099 and the LTSI, Université de Rennes 1, Rennes, 35042, France, and also with the Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), Rennes, France (e-mail: wanglyu1986@hotmail.com; amar.kachenoura@univ-rennes1.fr; lotfi.senhadji@univ-rennes1.fr).

L. Albera is with the INSERM, U1099 and the LTSI, Université de Rennes 1, Rennes 35042, France, with the Centre INRIA Rennes-Bretagne Atlantique, Rennes 35042, France, and also with the Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), Rennes 35042, France (e-mail: laurent.albera@univ-rennes1.fr).

H. Shu is with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and also with the Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), Nanjing 210096, China (e-mail: shu.list@seu.edu.cn).

Minimizing (4) is the main purpose of this paper. To estimate \mathbf{B} , based on LU matrix factorization, the high dimensional optimization is reduced to search a sequence of sparse triangular matrices. Let's recall the following definition:

Definition 1: A unit triangular matrix is a triangular matrix whose main diagonal elements are equal to 1.

Then any matrix \mathbf{B} with full column rank can be factorized as $\mathbf{B} = \mathbf{L}\mathbf{U}\mathbf{\Lambda}\mathbf{\Pi}$, where $\mathbf{L} \in \mathbb{R}^{N \times P}$ is a unit lower triangular matrix, and \mathbf{U} , $\mathbf{\Lambda}$ and $\mathbf{\Pi}$ are $(P \times P)$ unit upper triangular matrix, diagonal matrix and permutation matrix, respectively. Hence, due to the fact that $(\mathbf{L}\mathbf{U}\mathbf{\Lambda}\mathbf{\Pi})^{\square 2} = (\mathbf{L}\mathbf{U})^{\square 2} \mathbf{\Lambda}^{\square 2} \mathbf{\Pi}$ and the indeterminacies of the JDC problem, the matrix \mathbf{B} solving (4) can be chosen as $\mathbf{B} = \mathbf{L}\mathbf{U}$ without loss of generality. Now, let's consider the following definition and lemma:

Definition 2: An elementary triangular matrix $\mathbf{T}^{(i,j)}(a)$ is a unit triangular matrix whose non-diagonal elements are zeros except the (i, j) -th entry, which is equal to a .

Lemma 1: Any unit lower (or upper) triangular matrix belonging to $\mathbb{R}^{N \times P}$ can be factorized as a product of $(2N - P - 1)P/2$ elementary lower (or upper, respectively) triangular matrices.

The proof of lemma 1 is straightforward by reducing \mathbf{L} (or \mathbf{U}) into an elementary triangular matrix whose all non-diagonal elements are zeros, using Gaussian elimination. Lemma 1 yields that \mathbf{B} can be written as a product of elementary triangular matrices:

$$\mathbf{B} = \prod_{j=1}^P \prod_{i=j+1}^N \mathbf{L}^{(i,j)}(\ell_{i,j}) \prod_{i=1}^P \prod_{j=i+1}^P \mathbf{U}^{(i,j)}(u_{i,j}) \stackrel{\text{def}}{=} \prod_{(i \neq j)}^M \mathbf{T}^{(i,j)}(t_{i,j}) \quad (5)$$

where $\mathbf{T}^{(i,j)}(t_{i,j})$ is defined for the sake of convenience: it stands for $\mathbf{L}^{(i,j)}(\ell_{i,j})$ when $i > j$ and for $\mathbf{U}^{(i,j)}(u_{i,j})$ when $i < j$ with $M = P(N - 1)$ being the total number of elementary lower and upper triangular matrices. The sizes of all the $\mathbf{L}^{(i,j)}(\ell_{i,j})$ and $\mathbf{U}^{(i,j)}(u_{i,j})$ matrices are $(N \times N)$ and $(P \times P)$, respectively, except that of $\mathbf{L}^{(N,P)}(\ell_{N,P})$ which is $(N \times P)$. Ideally the minimization of (4) consists of finding M matrices $\mathbf{T}^{(i,j)}(t_{i,j})$ and the diagonal matrix set \mathcal{D} such that $J(\left(\prod_{(i \neq j)}^M \mathbf{T}^{(i,j)}(t_{i,j})\right)^{\square 2}, \mathcal{D})$ is a global minimum. As Yeredor's ACDC (Alternating Columns and Diagonal Centers) algorithm [20], the algorithm proposed in this paper alternates between two minimization schemes:

- The AC ("Alternating Columns") phase¹ minimizes (4) with respect to (w.r.t) one matrix $\mathbf{T}^{(i,j)}(t_{i,j})$ of \mathbf{B} , while keeping its other factors as well as \mathcal{D} fixed.
- The DC ("Diagonal Centers") phase minimizes (4) w.r.t the diagonal matrix set \mathcal{D} while keeping \mathbf{B} fixed.

Hence, we name the proposed algorithm ACDC_{LU}^+ .

A. AC Phase

In this phase, we minimize (4) w.r.t $\mathbf{T}^{(i,j)}(t_{i,j})$. Instead of simultaneously computing these M matrices, each $\mathbf{T}^{(i,j)}(t_{i,j})$ with a selected (i, j) index is identified sequentially. Suppose that $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are the current estimate of \mathbf{A} and \mathbf{B} , respectively. The update of $\tilde{\mathbf{B}}$ by one $\mathbf{T}^{(i,j)}(t_{i,j})$ is $\tilde{\mathbf{B}}\mathbf{T}^{(i,j)}(t_{i,j})$. Then we can show the following lemma:

¹The update of \mathbf{B} by right multiplying $\mathbf{T}^{(i,j)}(t_{i,j})$ with different (i, j) index affects the j -th column of \mathbf{B} sequentially. Thus we retain the terminology 'Alternating Columns phase'.

Lemma 2: The Hadamard square of the update is given by:

$$\left(\tilde{\mathbf{B}}\mathbf{T}^{(i,j)}(t_{i,j})\right)^{\square 2} = (\tilde{\mathbf{B}}^{\square 2})\mathbf{T}^{(i,j)}(t_{i,j}^2) + 2t_{i,j}(\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j)\mathbf{e}_j^\top \quad (6)$$

where $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{b}}_j$ denote the i -th and j -th columns of $\tilde{\mathbf{B}}$ respectively, and \mathbf{e}_j is the j -th column of the identity matrix.

The proof of lemma 2 is omitted due to lack of space. The optimization of (4) w.r.t $\mathbf{T}^{(i,j)}(t_{i,j})$ only consists of minimizing the following cost function:

$$J_{AC}^+(t_{i,j}) = \sum_{k=1}^K \left\| \mathbf{C}^{(k)} - \left[(\tilde{\mathbf{B}}\mathbf{T}^{(i,j)}(t_{i,j}))^{\square 2} \right] \mathbf{D}^{(k)} \times \left[(\tilde{\mathbf{B}}\mathbf{T}^{(i,j)}(t_{i,j}))^{\square 2} \right]^\top \right\|_F^2 \quad (7)$$

Inserting (6) into (7), the only unknown variable $t_{i,j}$ can be computed by minimizing the following cost function:

$$J_{AC}^+(t_{i,j}) = \sum_{k=1}^K \left\| \mathbf{F}_4^{(k)} t_{i,j}^4 + \mathbf{F}_3^{(k)} t_{i,j}^3 + \mathbf{F}_2^{(k)} t_{i,j}^2 + \mathbf{F}_1^{(k)} t_{i,j} + \mathbf{F}_0^{(k)} \right\|_F^2 \quad (8)$$

with:

$$\begin{aligned} \mathbf{F}_4^{(k)} &= -d_j^{(k)} \left[(\tilde{\mathbf{b}}_i^{\square 2}) (\tilde{\mathbf{b}}_i^{\square 2})^\top \right] \\ \mathbf{F}_3^{(k)} &= -2d_j^{(k)} \left[(\tilde{\mathbf{b}}_i^{\square 2}) (\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j)^\top + (\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j) (\tilde{\mathbf{b}}_i^{\square 2})^\top \right] \\ \mathbf{F}_2^{(k)} &= -d_j^{(k)} \left[(\tilde{\mathbf{b}}_i^{\square 2}) (\tilde{\mathbf{b}}_j^{\square 2})^\top + (\tilde{\mathbf{b}}_j^{\square 2}) (\tilde{\mathbf{b}}_i^{\square 2})^\top + 4(\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j) (\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j)^\top \right] \\ \mathbf{F}_1^{(k)} &= -2d_j^{(k)} \left[(\tilde{\mathbf{b}}_j^{\square 2}) (\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j)^\top + (\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j) (\tilde{\mathbf{b}}_j^{\square 2})^\top \right] \\ \mathbf{F}_0^{(k)} &= \mathbf{C}^{(k)} - (\tilde{\mathbf{B}}^{\square 2}) \mathbf{D}^{(k)} (\tilde{\mathbf{B}}^{\square 2})^\top \end{aligned} \quad (9)$$

where $d_j^{(k)}$ denotes the (j, j) -th entry of $\mathbf{D}^{(k)}$. Equation (8) shows that $J_{AC}^+(t_{i,j})$ is a 8-th degree polynomial in $t_{i,j}$. The global minimum $t_{i,j}$ can be obtained by computing the roots of its derivative and selecting the root yielding the smallest value of $J_{AC}^+(t_{i,j})$. Once the optimal $t_{i,j}$ is computed, the matrix $\tilde{\mathbf{A}}$ is updated by computing $(\tilde{\mathbf{B}}\mathbf{T}^{(i,j)}(t_{i,j}))^{\square 2}$.

However, it is observed that if the matrix set \mathcal{C} well fulfills the model (1), classical non-constrained JDC methods can give nonnegative \mathbf{A} . In this situation, imposing the nonnegativity constraint explicitly is unnecessary and increases the computational burden. Therefore, the nonnegativity constraint can be relaxed by decomposing $\mathbf{A} = \prod_{(i \neq j)} \mathbf{T}^{(i,j)}(t_{i,j})$ directly instead of using \mathbf{B} . Each parameter $t_{i,j}$ can be found by minimizing the following 4-th degree polynomial:

$$J_{AC}(t_{i,j}) = \sum_{k=1}^K \left\| \mathbf{F}_4^{(k)} t_{i,j}^4 + \left(\mathbf{F}_2^{(k)} + 4d_j^{(k)} (\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j) (\tilde{\mathbf{b}}_i \square \tilde{\mathbf{b}}_j)^\top \right) t_{i,j} + \mathbf{F}_0^{(k)} \right\|_F^2 \quad (10)$$

In practice, it is suggested to compute $t_{i,j}$ by minimizing (10) first. If all the elements in the j -th column of $\tilde{\mathbf{A}}\mathbf{T}^{(i,j)}(t_{i,j})$ have the same sign, the update $\tilde{\mathbf{A}} \leftarrow |\tilde{\mathbf{A}}\mathbf{T}^{(i,j)}(t_{i,j})|$ is adopted, where $|\cdot|$ denotes the absolute value. Otherwise, $t_{i,j}$ is computed by minimizing (8). Then the AC procedure is repeated to estimate

$\mathbf{T}^{(i,j)}(t_{i,j})$ with the next (i, j) index. The processing of all the M factors is called a full AC iteration.

B. DC Phase

In this phase, the minimization of (4) w.r.t the diagonal matrices $\{\mathbf{D}^{(k)}\}_{k=1}^K$ can be separated into K distinct linear least square subproblems:

$$J_{\text{DC}}^{(k)}(\mathbf{D}^{(k)}) = \left\| \mathbf{C}^{(k)} - \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^\top \right\|_F^2 \quad (11)$$

The optimal solution of $\mathbf{D}^{(k)}$ given by [20] is:

$$\mathbf{D}^{(k)} = \text{diag} \left((\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{c}^{(k)} \right) \quad (12)$$

where $\mathbf{H} = (\mathbf{A}^\top \mathbf{A})^{\square 2}$, $\mathbf{A} = (\mathbf{B})^{\square 2}$ and $\mathbf{c}^{(k)} = \text{vec}(\mathbf{C}^{(k)})$. $\text{vec}(\cdot)$ converts a matrix into a column vector by stacking its columns vertically.

In this context, one sweep is referred to the combination of one full AC iteration and one DC phase. The overall numerical complexity per sweep is $\mathcal{O}(N^3 PK)$, which is higher than $\mathcal{O}(N^2 PK)$ of ACDC [20]. In practice, several sweeps are necessary to ensure convergence. One can stop the algorithm when the value of $J(\mathbf{B}, \mathbf{D})$, or the decrease of $J(\mathbf{B}, \mathbf{D})$ between two successive sweeps falls below a fixed positive threshold. Such a stopping criterion is guaranteed to be met since $J(\mathbf{B}, \mathbf{D})$ is non-increasing in each sweep. We observed empirically that the proposed algorithm converges linearly.

III. SIMULATION RESULTS

In this section, the proposed ACDC_{LU}^+ method is compared with several state-of-the-art JDC methods and BSS algorithms. The performance is measured in terms of the error between the true joint diagonalizer \mathbf{A} and its estimate $\tilde{\mathbf{A}}$, as well as the source \mathbf{s} and its estimate $\tilde{\mathbf{s}}$ when a BSS context is considered. So the scale and permutation invariant distance defined in [7] is chosen as the preferred measure:

$$D(\mathbf{A}, \tilde{\mathbf{A}}) = \min_{\mathbf{\Pi}} \Psi(\mathbf{A}, \tilde{\mathbf{A}}\mathbf{\Pi}) \quad (13)$$

where the solution (13) requires to sweep all the $(P \times P)$ permutation matrices $\mathbf{\Pi}$, and:

$$\Psi(\mathbf{M}, \tilde{\mathbf{M}}) = \frac{1}{P} \sum_{p=1}^P \left\| \mathbf{m}_p - \frac{\tilde{\mathbf{m}}_p^\top \mathbf{m}_p}{\tilde{\mathbf{m}}_p^\top \tilde{\mathbf{m}}_p} \tilde{\mathbf{m}}_p \right\|_F \quad (14)$$

with \mathbf{m}_p and $\tilde{\mathbf{m}}_p$ denoting the p -th columns of \mathbf{M} and $\tilde{\mathbf{M}}$, respectively. Moreover, we repeat all the experiments with 500 independent Monte Carlo trials. All the algorithms stop either when the relative error of the corresponding criterion between two successive sweeps is less than 10^{-4} or when the number of sweeps exceeds 2000. In the following experiments, all the algorithms are initialized by identity matrices. All the simulations are implemented in Matlab v7.14 and run on Intel Dual-Core 2.8 GHz CPU with 32 Gb memory.

A. Simulated JDC Model

In this test, the behavior of the proposed ACDC_{LU}^+ method is evaluated and compared with several classical nonorthogonal JDC methods, including one direct-type method, namely ACDC [20], and three indirect-type methods, namely FFDIAG [22], LUJ1D [1] and UWEDGE [15]. The synthetic matrix set \mathbf{C} is generated randomly according to (1) with $N = 3$, $P = 3$ and

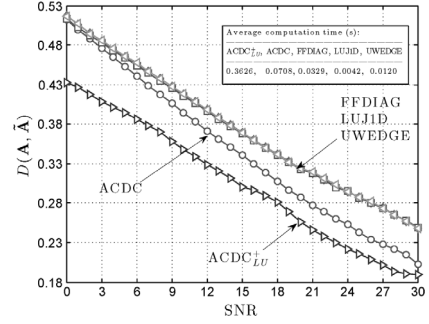


Fig. 1. Error $D(\mathbf{A}, \tilde{\mathbf{A}})$ evolution versus SNR.

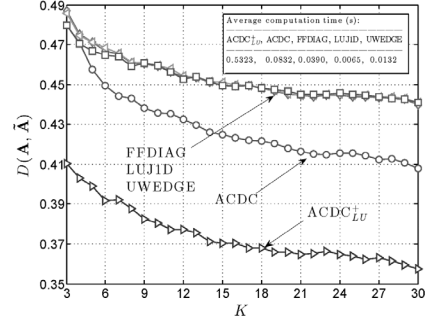


Fig. 2. Error $D(\mathbf{A}, \tilde{\mathbf{A}})$ evolution versus the number K of matrices.

$\mathbf{A} = \mathbf{B}^{\square 2}$. \mathbf{B} and the diagonal part of $\mathbf{D}^{(k)}$ are drawn from the zero-mean unit-variance Gaussian distribution. Each resulting target matrix $\mathbf{C}_N^{(k)}$ is perturbed by a random symmetric “noise” matrix as follows [4], [22]:

$$\mathbf{C}_N^{(k)} = \mathbf{C}^{(k)} + \sigma_N^{(k)} \left(\mathbf{N}^{(k)} \right) \left(\mathbf{N}^{(k)} \right)^\top \quad (15)$$

where $\sigma_N^{(k)}$ is a scalar controlling the noise level and $\mathbf{N}^{(k)}$ is drawn from a standard normal distribution. Then the SNR is defined as $\text{SNR} = -20 \log_{10}(\sigma_N^{(k)} \|\mathbf{N}^{(k)} \mathbf{N}^{(k)\top}\|_F / \|\mathbf{C}^{(k)}\|_F)$. We study the influence of SNR and the effect of the number K of matrices to be jointly diagonalized.

Fig. 1 shows the average error $D(\mathbf{A}, \tilde{\mathbf{A}})$ as a function of SNR ranging from 0 dB to 30 dB with $K = 5$. One can notice that the direct-type algorithms ACDC and the proposed method ACDC_{LU}^+ achieve better estimations than indirect-type methods. ACDC_{LU}^+ maintains a competitive advantage over ACDC, especially when SNR values are below 15 dB. Fig. 2 depicts the average error $D(\mathbf{A}, \tilde{\mathbf{A}})$ of the compared algorithms as a function of the quantity K varying from 3 to 30 with an SNR of 5 dB. As the quantity K grows, direct-type algorithms produce a larger performance gain than indirect-type. It can be seen that the proposed ACDC_{LU}^+ method gives the smallest estimation errors. The interest of using nonnegativity is then significant in this experimental context. The average running time of the compared algorithms in the above experiments is presented in the upper right corners of Fig. 1 and 2, respectively. It shows that the proposed algorithm requires more time but can achieve better results.

B. BSS Application on NMR Spectroscopy Data

In addition to the good performance of the proposed method in simulated JDC model, we further illustrate its capability for BSS. ACDC_{LU}^+ is compared with three well-known BSS algorithms, namely the ICA methods CoM2 [6] and SOBI [2], and a

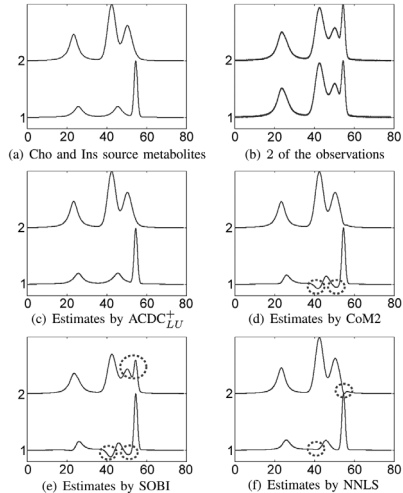


Fig. 3. NMR spectroscopy source metabolites, mixtures and estimated metabolites by $ACDC_{LU}^+$, CoM2, SOBI and NNLS.

TABLE I
AVERAGE ESTIMATING ERRORS OF MIXING MATRIX $D(\mathbf{A}, \tilde{\mathbf{A}})$,
SOURCES $D(\mathbf{s}^T, \tilde{\mathbf{s}}^T)$ AND COMPUTATION TIME OF FOUR METHODS
FOR BSS OF NMR SPECTROSCOPY SIGNALS

	$ACDC_{LU}^+$	CoM2	SOBI	NNLS
$D(\mathbf{A}, \tilde{\mathbf{A}})$	0.0370	0.1102	0.4277	0.0945
$D(\mathbf{s}^T, \tilde{\mathbf{s}}^T)$	0.0330	0.1521	0.4226	0.0794
Computation time (s)	1.5730	0.0821	3.2834	1.5700

NMF method based on alternating NonNegativity Least Squares (NNLS) [9], through an experiment carried out on simulated NMR spectroscopy data [11]. Two source metabolites, namely the Choline (Cho) and Myo-inositol (Ins) (see Fig. 3(a)), are generated by means of Lorenzian functions with fixed parameters for a realistic representation [10]. It is assumed that the linear mixing model $\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\nu}$ is valid, where $\boldsymbol{\nu}$ is an additive white Gaussian noise with an SNR of 30 dB. 10 linear observations (see Fig. 3(b)) are created with \mathbf{A} similarly generated as in the previous section. The matrix set \mathcal{C} is built using 100 4-th order cumulant matrix slices.

Table I shows the average estimation errors $D(\mathbf{A}, \tilde{\mathbf{A}})$ of the mixing matrices, that $D(\mathbf{s}^T, \tilde{\mathbf{s}}^T)$ of the source metabolites and the computation time of the compared methods. From the results of NNLS and $ACDC_{LU}^+$, it is obvious that the nonnegativity constraint allows us to improve the estimation since the two sources are not totally independent. In addition, the proposed $ACDC_{LU}^+$ method gives the smallest estimating errors both for \mathbf{A} and \mathbf{s} . The estimated sources displayed in Fig. 3(c) to 3(f) coincide with the results shown in Table I. Regarding the unconstrained methods, SOBI can not separate both sources and CoM2 can not eliminate the influence of the Ins peak in the estimated Cho metabolite. NNLS gives a good result, but the two metabolites still slightly exist in each other's estimate. As far as $ACDC_{LU}^+$ is concerned, the estimated source metabolites are quasi-perfect. It appears that, in this experimental context, the proposed $ACDC_{LU}^+$ technique based on a nonnegativity constraint improves the BSS performance in an acceptable running time.

IV. CONCLUSION

In this letter, we propose a novel nonnegative joint diagonalization by congruence algorithm named $ACDC_{LU}^+$ which computes

a nonnegative joint diagonalizer. Simulation results illustrate its improvement in dealing with degenerate JDC models and its potential usefulness in BSS applications.

REFERENCES

- [1] B. Afsari, "Simple LU and QR based non-orthogonal matrix joint diagonalization," in *ICA 2006, Springer LNCS Series*, 2006.
- [2] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, 1997.
- [3] J. F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 17, pp. 161–164, 1996.
- [4] G. Chabriel and J. Barrère, "A direct algorithm for nonorthogonal approximate joint diagonalization," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 39–47, 2012.
- [5] M. Chu, F. Diele, R. Plemmons, and S. Ragni, *Optimality, Computation, Interpretation of Nonnegative Matrix Factorizations* Wake Forest Univ., Winston-Salem, NC, USA, 2004, Tech. Rep..
- [6] P. Comon, "Independent component analysis, a new concept?," *Signal Process.*, vol. 36, no. 3, pp. 287–314, 1994.
- [7] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometr.*, vol. 23, pp. 393–405, 2009.
- [8] M. Joho, "Newton method for joint approximate diagonalization of positive definite hermitian matrices," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1205–1218, 2008.
- [9] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 713–730, 2008.
- [10] S. Moussaoui, "Séparation de sources non-négatives: application au traitement des signaux de spectroscopie," Ph.D. dissertation, Univ. Henri Poincaré, Nancy, France, 2005.
- [11] H. Parastara, M. Jalali-Heravia, and R. Taulerb, "Is independent component analysis appropriate for multivariate resolution in analytical chemistry?," *Trends Analyt. Chem.*, vol. 31, pp. 134–143, 2012.
- [12] D. T. Pham, "Joint approximate diagonalization of positive definite hermitian matrices," *SIAM J. Matrix Anal. Appl.*, vol. 22, pp. 1837–1848, 2001.
- [13] P. Sajda, S. Du, T. R. Brown, R. Stoyanova, D. C. Shungu, X. Mao, and L. C. Parra, "Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain," *IEEE Trans. Med. Imag.*, vol. 23, no. 12, pp. 1453–1465, 2004.
- [14] A. Souloumiac, "Nonorthogonal joint diagonalization by combining Givens and hyperbolic rotations," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2222–2231, 2009.
- [15] P. Tichavský and A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 878–891, 2009.
- [16] K. Todros and J. Tabrikian, "QML-based joint diagonalization of positive-definite hermitian matrices," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4656–4673, 2010.
- [17] A. J. van der Veen, "Joint diagonalization via subspace fitting techniques," in *Proc. ICASSP'01*, 2001, vol. 5, pp. 2773–2776.
- [18] R. Vollgraf and K. Obermayer, "Quadratic optimization for simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3270–3278, 2006.
- [19] X. F. Xu, D. Z. Feng, and W. X. Zheng, "A fast algorithm for nonunitary joint diagonalization and its application to blind source separation," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3457–3463, 2011.
- [20] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1545–1553, 2002.
- [21] G. Zhou, S. Xie, Z. Yang, and J. Zhang, "Nonorthogonal approximate joint diagonalization with well-conditioned diagonalizers," *IEEE Trans. Neural Netw.*, vol. 20, no. 11, pp. 1810–1819, 2009.
- [22] A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller, "A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation," *J. Mach. Learn. Res.*, vol. 5, pp. 777–800, 2004.