

# Diagonalisation conjointe non-orthogonale de matrices complexes basée sur une décomposition LU\*

Victor MAURANDI<sup>1,2</sup>, Christophe DE LUIGI<sup>1,2</sup> et Eric MOREAU<sup>1,2</sup>

<sup>1</sup>Aix Marseille Université, CNRS, ENSAM, LSIS, UMR 7296  
13397 Marseille, France.

<sup>2</sup>Université de Toulon, CNRS, LSIS, UMR 7296  
83957 La Garde, France.

{maurandi, deluigi, moreau}@univ-tln.fr

**Résumé** – Dans cet article, nous considérons le problème de la diagonalisation conjointe non-orthogonale d’un ensemble de matrices complexes. Ce dernier apparaît dans de nombreux problèmes en traitement du signal et plus particulièrement en séparation de source. Nous proposons un nouvel algorithme de type Jacobi, basé sur la décomposition LU de la matrice de séparation recherchée. Cet algorithme consiste à estimer les paramètres de la matrice de séparation de façon découplée. Des simulations numériques sont fournies afin de comparer les performances avec un algorithme proche développé dans le cas réel et adapté au cas complexe.

**Abstract** – In this paper, we consider the problem of non-orthogonal joint diagonalization of a set of complex matrices. This appears in many signal processing problems, especially in source separation. We propose a new algorithm based on LU decomposition of the unknown matrix associated to a Jacobi like procedure. The algorithm is based on a decoupled parameter estimation. Numerical simulations are provided to compare the performances of our algorithm with a similar one existing in the real case and adapted to the complex case.

## 1 Introduction

La diagonalisation conjointe d’ensembles de matrices apparaît dans de nombreux problèmes de traitement du signal et plus particulièrement en séparation de sources [1]-[11]. Le cadre unitaire a d’abord été considéré [1][2]. Aujourd’hui, le cadre non unitaire est devenu important car il permet principalement de se passer d’une étape préalable de traitement (blanchiment des observations en séparation de sources) limitant dans une certaine mesure les performances en pratique. De nombreux algorithmes ont été récemment proposés [3]-[11]. Certains se basent sur un critère direct permettant l’estimation de la matrice de mélange tandis que d’autres se basent sur un critère inverse permettant l’estimation de la matrice de séparation sans inversion explicite.

Cet article est organisé de la manière suivante : tout d’abord nous exposerons le problème de la diagonalisation conjointe. Nous rappellerons le critère inverse classique que nous utilisons. Puis nous développerons un algorithme de diagonalisation conjointe de type Jacobi basé sur la décomposition LU de la matrice de séparation recherchée. Finalement, à l’aide de simulations numériques, nous évaluerons les performances de cet algorithme en le comparant à un algorithme de la littérature développé dans le cas réel [8] et que l’on a adapté ici au cas complexe.

---

\*Les auteurs remercient la Direction Générale de l’Armement pour son support financier.

## 2 Formulation du problème

On considère  $K$  ( $K \geq 2$ ) matrices complexes symétriques ou hermitiennes  $\mathbf{M}_k$  se décomposant sous la forme :

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

où  $(\cdot)^\dagger$  correspond soit à l’opérateur transposé dans le cas complexe symétrique, soit à l’opérateur conjugué-transposé dans le cas complexe hermitien. Dans toute la suite on ne considèrera uniquement que le cas de matrices carrés, toutes de dimension  $N \times N$ . Les matrices  $\mathbf{D}_k$  sont toutes diagonales. Dans le cas hermitien, elles sont réelles. La matrice  $\mathbf{A}$  est la matrice de mélange supposée inversible.

Le but de la diagonalisation conjointe consiste à estimer à partir des matrices  $\mathbf{M}_k$ , une matrice dite de séparation  $\mathbf{B}$  (égale à l’inverse de la matrice de mélange au produit près d’une matrice diagonale inversible et d’une matrice de permutation) de telle manière que les matrices  $\mathbf{B}\mathbf{M}_k\mathbf{B}^\dagger$  soient (approximativement) conjointement diagonales. Dans ce but, nous utilisons le critère inverse classique défini par

$$J(\mathbf{B}) = \sum_{k=1}^K \|\mathbf{B}\mathbf{M}_k\mathbf{B}^\dagger - \text{Diag}\{\mathbf{B}\mathbf{M}_k\mathbf{B}^\dagger\}\|_F^2, \quad (2)$$

où  $\text{Diag}\{\mathbf{M}\}$  est la matrice diagonale définie à partir de la diagonale de  $\mathbf{M}$  et où  $\|\cdot\|_F$  est la norme de Frobenius. Il s’agit maintenant de trouver une matrice  $\mathbf{B}$  minimisant  $J(\mathbf{B})$  tout en évitant la solution triviale  $\mathbf{B} = \mathbf{0}$ . A cette fin nous proposons

un algorithme de minimisation de type Jacobi se basant sur une décomposition LU de  $\mathbf{B}$ .

### 3 Algorithmes proposés

Toute matrice carrée peut toujours être décomposée sous la forme  $\mathbf{DPLU}$  où  $\mathbf{D}$  est une matrice diagonale,  $\mathbf{P}$  une matrice de permutation et  $\mathbf{L}$  et  $\mathbf{U}$  des matrices triangulaires, inférieure et supérieure respectivement, de coefficients diagonaux tous égaux à 1.

Les indéterminations classiques de ce problème font que les matrices  $\mathbf{D}$  et  $\mathbf{P}$  de la décomposition précédente ne pourront pas être identifiées. Ainsi il est suffisant de rechercher une matrice  $\mathbf{B}$  s'écrivant sous la forme  $\mathbf{B} = \mathbf{LU}$ . Un grand avantage de cette paramétrisation est que forcément  $\mathbf{B}$  sera une matrice inversible, donc a fortiori non nulle, car  $\mathbf{L}$  et  $\mathbf{U}$  ont des éléments diagonaux tous égaux à 1.

Il s'agit donc maintenant d'estimer l'ensemble des paramètres libres de  $\mathbf{L}$  et  $\mathbf{U}$ . Afin de simplifier cette estimation, nous allons considérer une méthode de type Jacobi, *i.e.* écrire le problème de dimension  $N \times N$  comme un produit de sous problèmes de "dimension"  $2 \times 2$  en considérant tous les couples de mêmes indices de lignes et de colonnes.

Par exemple, dans le cas  $3 \times 3$ , on considère donc la décomposition suivante :

$$\mathbf{B} = \begin{pmatrix} B_{11}^1 & B_{12}^1 & 0 \\ B_{21}^1 & B_{22}^1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} B_{11}^2 & 0 & B_{13}^2 \\ 0 & 1 & 0 \\ B_{31}^2 & 0 & B_{33}^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & B_{22}^3 & B_{23}^3 \\ 0 & B_{32}^3 & B_{33}^3 \end{pmatrix}, \quad (3)$$

où  $B_{ij}^q$  avec  $(i, j) \in \{1, \dots, N\}^2$ ,  $q = 1, \dots, \frac{N(N-1)}{2}$ , sont des coefficients complexes.

Chacune des sous-matrices  $2 \times 2$  sera écrite comme le produit des matrices  $\mathbf{L}$  et  $\mathbf{U}$  suivantes

$$\mathbf{L} = \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} \quad \mathbf{U} = \begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix} \quad (4)$$

où  $l$  et  $u$  sont deux paramètres complexes inconnus qu'il s'agit maintenant d'estimer.

#### 3.1 L'estimation découplée

Toujours dans un but de simplification et de manière similaire à [8], nous proposons de découpler l'estimation de  $l$  et  $u$ . C'est-à-dire que l'on va commencer par imposer la matrice de séparation  $\mathbf{B}$  égale simplement à  $\mathbf{U}$  pour estimer  $u$  puis rendre  $\mathbf{B}$  égale à  $\mathbf{L}$  pour estimer  $l$ . On itérera ensuite ce processus.

La différence principale avec [8] est que l'on va uniquement se restreindre au cas  $2 \times 2$  pour l'estimation successive de  $l$  et  $u$  de manière identique au processus de Jacobi classique dans le cas unitaire. Cependant ici puisque l'on ne considère pas le cas unitaire, cela conduit à ne pas minimiser directement le critère global mais le "sous" critère  $2 \times 2$  correspondant.

#### 3.1.1 Estimation de $\mathbf{U}$

En ne considérant donc uniquement que le cas  $2 \times 2$ , on pose

$$\mathbf{M}_k = \begin{pmatrix} M_{k,11} & M_{k,12} \\ M_{k,21} & M_{k,22} \end{pmatrix}. \quad (5)$$

On a de plus  $M_{k,12} = M_{k,21}$  dans le cas symétrique ou bien  $M_{k,12} = M_{k,21}^*$  où  $(\cdot)^*$  est l'opérateur conjugué, dans le cas hermitien. Dans ce dernier cas hermitien, on a aussi  $M_{k,11}$  et  $M_{k,22}$  réels.

En posant

$$\mathbf{B} = \mathbf{U} = \begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix}, \quad (6)$$

le critère  $J(\mathbf{B})$  en (2) restreint au cas  $2 \times 2$  correspondant et alors noté  $J_{22}(\mathbf{B})$ , s'écrit dans le cas symétrique

$$J_{22}(\mathbf{U}) = 2 \sum_{k=1}^K |M_{k,12} + uM_{k,22}|^2, \quad (7)$$

et dans le cas complexe hermitien

$$J_{22}(\mathbf{U}) = \sum_{k=1}^K |M_{k,12} + uM_{k,22}|^2 + |M_{k,12}^* + u^*M_{k,22}|^2. \quad (8)$$

En raison des symétries de  $\mathbf{M}_k$ , la valeur de  $u$  minimisant  $J_{22}(\mathbf{U})$  prend la même expression dans les deux cas

$$u = - \frac{\sum_{k=1}^K M_{k,12} M_{k,22}^*}{\sum_{k=1}^K |M_{k,22}|^2}. \quad (9)$$

#### 3.1.2 Estimation de $\mathbf{L}$

En posant maintenant

$$\mathbf{B} = \mathbf{L} = \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix}, \quad (10)$$

le critère  $J(\mathbf{B})$  en (2) restreint au cas  $2 \times 2$  correspondant s'écrit dans le cas symétrique

$$J_{22}(\mathbf{L}) = \sum_{k=1}^K |M_{k,12} + lM_{k,11}|^2, \quad (11)$$

et dans le cas complexe hermitien

$$J_{22}(\mathbf{L}) = \sum_{k=1}^K |M_{k,21} + lM_{k,11}|^2 + |M_{k,21}^* + l^*M_{k,11}|^2. \quad (12)$$

En raison des symétries de  $\mathbf{M}_k$ , la valeur de  $l$  minimisant  $J_{22}(\mathbf{L})$  prend la même expression dans les deux cas

$$l = - \frac{\sum_{k=1}^K M_{k,21} M_{k,11}^*}{\sum_{k=1}^K |M_{k,11}|^2}. \quad (13)$$

### 3.2 Deux stratégies différentes

On peut considérer deux stratégies d'optimisation itératives pour résoudre le problème  $N \times N$ . La première consiste à estimer les matrices  $2 \times 2$   $\mathbf{U}$  et à mettre à jour les matrices à diagonaliser, cela pour l'ensemble des paires, puis faire de même pour les matrices  $2 \times 2$   $\mathbf{L}$ . On attribue à cet algorithme qui utilise une stratégie de type bloc, l'acronyme DLU1. La seconde stratégie consiste, pour chaque paire, à estimer dans un premier temps la matrice  $2 \times 2$   $\mathbf{U}$ , à mettre à jour les matrices à diagonaliser, puis faire de même avec la matrice  $2 \times 2$   $\mathbf{L}$ . On attribue à cet algorithme qui utilise une stratégie de type alterné, l'acronyme DLU2.

## 4 Simulations numériques

Afin d'évaluer les performances des algorithmes, l'indice de performance utilisé est celui proposé dans [2][10][11]. Il compare la matrice globale  $\mathbf{S} = \mathbf{B}\mathbf{A} = (S_{ij})$  au produit d'une matrice de permutation et d'une matrice diagonale inversible. Il est défini suivant :

$$I(\mathbf{S}) = \frac{1}{2N(N-1)} \left( \sum_{i=1}^N \left( \sum_{j=1}^N \frac{|S_{ij}|^2}{\max_{\ell} |S_{i\ell}|^2} - 1 \right) + \sum_{j=1}^N \left( \sum_{i=1}^N \frac{|S_{ij}|^2}{\max_{\ell} |S_{\ell j}|^2} - 1 \right) \right). \quad (14)$$

Cet indice non-négatif est égal à zéro si  $\mathbf{S}$  satisfait  $\mathbf{B} = \mathbf{DPA}^{-1}$ . Dans cette partie consacrée aux simulations numériques, nous nous concentrerons sur le cas complexe hermitien, le cas complexe symétrique donnant des performances similaires.

Au moyen de simulations de type Monte Carlo (100 tirages indépendants), nous comparons les performances des algorithmes DLU1 et DLU2 proposés ici avec l'algorithme LUJ1D proposé par B. Afsari dans le cas réel [8] et adapté dans ce papier au cas complexe hermitien. Nous considérons 25 matrices complexes hermitiennes  $10 \times 10$  dans les cas :

- Non-bruité
- Bruité de la forme  $\mathbf{M}_k + t \mathbf{N}_k$  avec  $\mathbf{N}_k$ ,  $k = 1, \dots, 25$ , matrices  $10 \times 10$  complexes hermitiennes et  $t$  un facteur réel pondérant le niveau de bruit.

Pour la génération des matrices  $\mathbf{M}_k$ , les coefficients des matrices  $\mathbf{D}_k$  et  $\mathbf{N}_k$  suivent tous une loi normale centrée-réduite pour leur partie réelle et leur partie imaginaire. La matrice  $\mathbf{A}$  est construite suivant le modèle LU appliqué à la matrice de séparation  $\mathbf{B}$ . On se réfère donc au schéma de Jacobi (cf (3)), pour chaque paire  $(i, j)$  avec  $i < j$ , on prend  $B_{ii}^q = 1$ ,  $B_{ij}^q = u_q$ ,  $B_{ji}^q = l_q$ ,  $B_{jj}^q = 1 + l_q u_q$ . Chaque  $l_q$  et chaque  $u_q$  suivent une loi uniforme sur l'intervalle  $\left] \frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right[$  pour leur partie réelle et leur partie imaginaire. Enfin, on initialise la matrice de séparation  $\mathbf{B}$  avec la matrice identité. Finalement, on affichera la valeur médiane du critère global  $J(\mathbf{B})$  (cf fig. 1) puis celle de l'indice de performance  $I(\mathbf{S})$  (cf fig. 2, fig. 3 et tab. 1) en fonction du nombre d'itération.

### Convergence

En premier lieu, nous nous intéressons à la convergence des algorithmes pour le critère global  $J(\mathbf{B})$  défini en (2). En effet, étant donné que les algorithmes DLU1 et DLU2 sont construits non pas sur la minimisation de  $J(\mathbf{B})$  mais sur celle du "sous" critère  $2 \times 2$  correspondant, on peut se demander s'ils minimisent le critère global. Au moyen de simulations numériques (25 matrices de dimension  $10 \times 10$  dans le cas bruité avec  $t = 10^{-3}$ ), on observe que DLU1 et DLU2 minimisent effectivement le critère  $J(\mathbf{B})$  (cf fig. 1), de plus ils présentent une meilleure vitesse de convergence et une meilleure précision que l'algorithme LUJ1D.

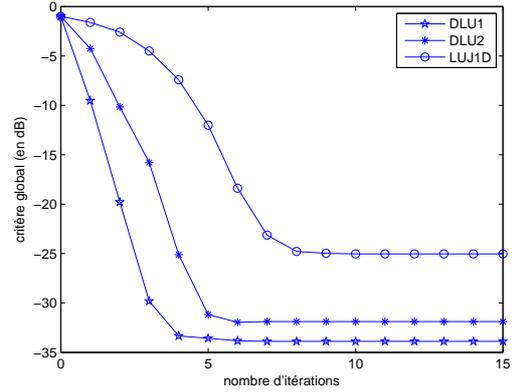


FIGURE 1 – Convergence du critère global pour 25 matrices  $10 \times 10$  en présence de bruit avec  $t = 10^{-3}$ .

### Cas non-bruité

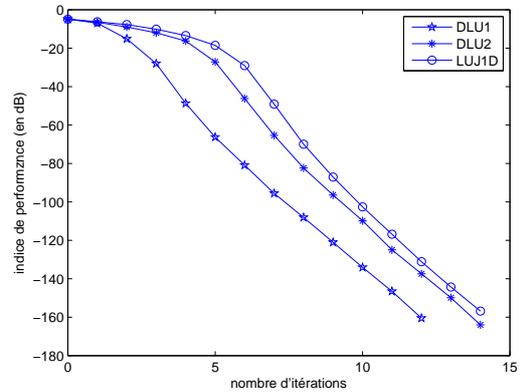


FIGURE 2 – Comparaison des algorithmes pour 25 matrices  $10 \times 10$  sans bruit.

Dans le cas sans bruit (cf fig. 2), on vérifie bien que les deux algorithmes diagonalisent parfaitement les matrices  $\mathbf{M}_k$ . Cependant, l'algorithme DLU1 possède la plus grande vitesse de convergence, en particulier sur les premières itérations (balayage de l'intégralité des paires). L'algorithme DLU2 converge aussi plus rapidement que LUJ1D mais la différence est moindre.

## Cas bruité

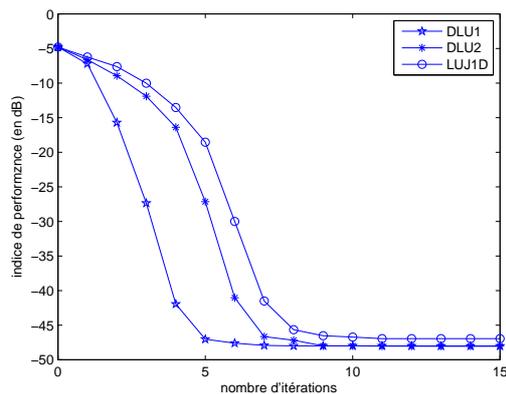


FIGURE 3 – Comparaison des algorithmes pour 25 matrices  $10 \times 10$  en présence de bruit avec  $t = 10^{-3}$ .

En présence de bruit (cf fig. 3), l’algorithme DLU1 montre aussi une plus grande vitesse de convergence. De plus dans ce cas, les deux algorithmes proposés convergent vers la même valeur (environ  $-47\text{dB}$ ) tandis que LUJ1D est  $2\text{dB}$  moins précis. Ainsi DLU1 possède globalement les meilleures performances.

## Influence de la taille des matrices

Nous nous intéressons ici à l’influence de la taille des matrices sur la valeur de l’indice de performance après convergence. On génère 25 matrices de taille  $N$  allant de 3 à 15 avec un paramètre de bruit  $t = 10^{-3}$ . Les résultats sont donnés dans le tableau 1. Pour des petites tailles de matrices, les trois al-

TABLE 1 – Performance des algorithmes (en dB) en fonction de la taille des matrices ( $N$ ).

$N$	3	5	8	10	13	15
DLU1	-68.9	-63.2	-55.0	-46.7	-35.1	-29.6
DLU2	-68.9	-63.2	-55.0	-46.7	-35.1	-29.6
LUJ1D	-68.9	-63.2	-55.0	-44.8	-30.8	-27.1

gorithmes convergent exactement vers la même valeur. Cependant lorsque la taille des matrices est plus grande ( $N \geq 10$ ), on constate que LUJ1D perd en précision par rapport à DLU1 et DLU2 qui, eux, continuent à converger vers la même valeur.

## 5 Conclusion

Dans le cadre de la diagonalisation conjointe non-orthogonale de matrices complexes, nous avons proposé un algorithme basé sur une décomposition LU découplée. Il s’inspire de celui proposé dans [8], mais repose sur l’optimisation d’une modification du critère classique utilisé dans [8] afin de se focaliser sur l’annulation de couples successifs d’éléments hors de la diag-

onale. Nous avons proposé deux stratégies dépendant du calcul des matrices  $L$  et  $U$  effectué de manière bloc ou alternée. Les simulations numériques illustrent que notre approche possède à la fois une meilleure vitesse de convergence et de meilleures performances après convergence dans le cas bruité. Ces mêmes simulations démontrent que la stratégie de type bloc offre de meilleures performances en terme de vitesse de convergence par rapport à la stratégie de type alterné.

## Références

- [1] J.-F. Cardoso and A. Souloumiac, “Blind beamforming for non Gaussian signals”, *IEE Proceedings-F*, Vol. 40, pp 362-370, 1993.
- [2] E. Moreau, “A generalization of joint-diagonalization criteria for source separation”, *IEEE Trans. Signal Processing*, Vol. 49, No. 3, pp 530-541, March 2001.
- [3] A. Yeredor, “Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation”, *IEEE Trans. Signal Processing*, Vol. 50, No. 7, pp 1545-1553, July 2002.
- [4] H. Ghennioui, E.-M. Fadaili, N. Thirion-Moreau, A. Adib and E. Moreau, “A non-unitary joint block diagonalization algorithm for blind separation of convolutive mixtures of sources,” *IEEE Signal Process. Letters*, Vol. 14, No. 11, pp. 860-863, Nov. 2007.
- [5] X.-L. Li and X.-D. Zhang, “Nonorthogonal joint diagonalization free of degenerate solution,” *IEEE Trans. on Sig. Pro.*, Vol. 55, No. 5, May 2007
- [6] X. Guo, S. Zhu, S. Miron and D. Brie, “Approximate joint diagonalization by nonorthogonal nonparametric Jacobi transformations,” in *Proc. ICASSP’2010*, Dallas, USA, March 2010.
- [7] A. Souloumiac, “Nonorthogonal joint diagonalization by combining Givens and hyperbolic rotations,” *IEEE Trans. Signal Processing*, vol. 57, no. 6, pp. 2222-2231, June 2009.
- [8] B. Afsari, “Simple LU and QR based non-orthogonal matrix joint diagonalization”, in *Proc. ICA’2006*, Springer LNCS series, pp 1-7, 2006.
- [9] T. Trainini and E. Moreau, “Variations around gradient like algorithms for joint diagonalization of Hermitian matrices,” in *Proc. EUSIPCO’2012*, Bucharest, Romania, August 2012.
- [10] E. Moreau and O. Macchi, “A one stage self-adaptive algorithm for source separation,” in *Proc. ICASSP*, april 1994, vol. 3, pp. 49-52.
- [11] E. Moreau and O. Macchi, “High order contrasts for self-adaptive source separation,” *International Journal of Adaptive Control and Signal Processing*, vol. 10, no. 1, pp. 19-46, January 1996.