



HAL
open science

Algorithmes pour la diagonalisation conjointe de tenseurs sans contrainte unitaire. Application à la séparation MIMO de sources de télécommunications numériques

Victor Maurandi

► **To cite this version:**

Victor Maurandi. Algorithmes pour la diagonalisation conjointe de tenseurs sans contrainte unitaire. Application à la séparation MIMO de sources de télécommunications numériques. Autre. Université de Toulon, 2015. Français. NNT : 2015TOUL0009 . tel-01376683

HAL Id: tel-01376683

<https://theses.hal.science/tel-01376683>

Submitted on 5 Oct 2016

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.



**ECOLE DOCTORALE 548
MER ET SCIENCES**

Laboratoire des Sciences de l'Information et des Systèmes (LSIS)
Equipe Signal et Image (SIIM)

THÈSE

pour l'obtention du grade de

Docteur en Sciences de l'Université de Toulon

Mention : Automatique, signal, productique, robotique

présentée et soutenue le 30 novembre 2015 par

Victor MAURANDI

**ALGORITHMES POUR LA DIAGONALISATION CONJOINTE DE TENSEURS
SANS CONTRAINTTE UNITAIRE. APPLICATION À LA SÉPARATION MIMO DE
SOURCES DE TÉLÉCOMMUNICATIONS NUMÉRIQUES**

Thèse dirigée par Eric MOREAU

Jury :

M. JEAN-FRANÇOIS CARDOSO	Directeur de recherche au CNRS à Télécom Paris, LTCI	Président
M. CHRISTIAN JUTTEN	Professeur à l'Université Joseph Fourier de Grenoble, GIPSA-lab	Rapporteur
M. LAURENT ALBERA	Maître de conférences HDR à l'Université de Rennes 1, LTSI	Rapporteur
M. SALAH BOURENNANE	Professeur à l'Institut Fresnel	Examineur
M. ERIC MOREAU	Professeur à l'Université de Toulon, LSIS	Directeur de Thèse
MME VÉRONIQUE SERFATY	Responsable DGA du domaine I2R	Invitée

Remerciements

Toutes les bonnes choses ont une fin. Il est donc temps de clôturer ce chapitre de ma vie et ces trois années de doctorat. J'en profite pour écrire ces quelques lignes afin de remercier toutes les personnes qui ont partagé cette expérience avec moi.

Tout d'abord, je tiens à remercier la Direction Générale de l'Armement ainsi que le Centre National de la Recherche Scientifique qui ont co-financé cette thèse et sans qui mes travaux de recherche n'auraient pas pu aboutir.

Je tiens aussi à exprimer toute ma reconnaissance aux membres de mon jury de thèse pour l'intérêt qu'ils ont porté à mes travaux. Je remercie mes rapporteurs Christian Jutten et Laurent Albera pour leur travail de relecture très consciencieux. Merci aussi à mes examinateurs Jean-François Cardoso et Salah Bourennane pour leur présence et leur questions très pertinentes. Merci, enfin, à ma tutrice DGA Véronique Serfaty qui a suivi l'évolution de mes recherches au cours de ces trois années.

J'adresse mes plus sincères remerciements à mon directeur de thèse, Eric Moreau, qui m'a donné les clés pour en arriver là, aujourd'hui. Il a su à la fois me guider, me pousser à me dépasser et me transmettre sa passion. Je suis très heureux d'avoir pu échanger avec lui durant ces trois ans tant sur le plan scientifique que sur le plan humain, et je suis très fier qu'il m'ait permis de mettre un pied à l'étrier du monde de la recherche.

Je remercie l'équipe des enseignants-chercheurs de l'« aile Télécom » qui, non contents de m'avoir supporté pendant trois ans en tant qu'étudiant en école d'ingénieur, m'ont accueilli à bras ouverts lorsque j'ai débuté ma thèse : Nadège Thirion-Moreau, Audrey Roman, Cyril Prissette (avec qui j'ai beaucoup trop de T-shirts en commun), Sylvain Maire qui devrait réellement ouvrir une boutique de cartes Magic (*La carte Maire*), Xavier Luciani, Bernard Xerri, Jean Barrère, Olivier Derrien, Frédéric Bouchara. Merci pour toutes ces discussions hautement

scientifiques ou non, merci pour vos conseils avisés et surtout merci pour votre bonne humeur.

J'en arrive à notre joyeuse bande de « non permanents », j'ai nommé les doctorants, post-doctorants et autres ingénieurs recherche. Je pense les avoir suffisamment emmouscaillés, comme on dit par chez moi, pour qu'ils se souviennent de moi ! Ce qu'ils ne savent peut-être pas, c'est que moi non plus, je ne les oublierai pas de sitôt. Je pense, notamment, à Jean-Philip, mon partenaire de pétanque avec qui j'ai battu la légende Sylvain Maire ; Diogone Bolt qui a réussi à me faire croire qu'il était sénégalais ; Manchun qui ne m'aura jamais livré sa fameuse recette du chat-lami ; Giang (qui a eu l'outrecuidance de ne pas appeler son fils Victor) à qui je souhaite plein de bonheur avec sa petite famille ; Xuan, la reine du Da Cau et du yoga ; Chéchile, *a minha favorita portuguesa*, que j'aurai traumatisée avec ma balle en mousse ; Rémi avec un « i » et nos débats houleux autour d'une bière ; Amine *el mejor aficionado del Barca* ; Omar notre rayon de soleil au LSIS, même depuis le Maroc, on peut ressentir sa bonne humeur ; Emilien, le pongiste de l'extrême, la plus grande progression que l'histoire du ping-pong n'ait jamais connue (d'aucun raconte qu'il viendrait s'entraîner toutes les nuits) ; Vincente et nos débats philosophiques jusqu'à pas d'heure ; Thibault que je n'aurai pas côtoyé très longtemps mais avec lequel j'aurai quand même bien rigolé ; Zineddine que j'espère très heureux avec sa famille en Algérie ; mais aussi Justine, Ikhlef, Kheir-Eddine, Vincent, Céline. Sans oublier mes doctorantes préférées du MIO : Gwen et Amandine.

Je remercie aussi, mes amis de l'entretien, Aurélie qui nous a toujours chouchoutés, Sandro (un tiers homme, un tiers nounours) et Mikapalão ainsi que Claudine de la cafétéria et les étudiants de l'école d'ingénieur SeaTech : Kemar, Eti, Marco, Gégé, Hélène, Anaïs, Thomas, Geoff, Benjamin et j'en oublie certainement beaucoup.

Je pense aussi à tous mes irréductibles amis qui partagent ma vie depuis bien longtemps maintenant et qui sont toujours là pour moi quand j'ai besoin de décompresser. Il y a, bien entendu, mes amis des années collège-lycée : les quatre autres fantastiques, à savoir Léão, Steph, Patou et P-A ainsi que Julie, Sophie, Laura, Jeff, Marcus et Alix. Mes amis de la fac : Ronan, Mathieu et Rémi. Et mes amis de l'ISITV : Mahieu, Séb, Esteb, 'tite Sandra, Mauween, Terry, Toinou et Thom'glom.

Enfin, je n'en serai certainement pas là sans ma famille qui a su me conseiller et me soutenir dans tous les choix que j'ai eu à prendre dans ma vie. Ils m'ont insufflé la curiosité et la soif de connaissance. Je ne les en remercierai jamais assez.

Résumé

Cette thèse développe des méthodes de diagonalisation conjointe de matrices et de tenseurs d'ordre trois, et son application à la séparation MIMO de sources de télécommunications numériques. Après un état, les motivations et objectifs de la thèse sont présentés. Les problèmes de la diagonalisation conjointe et de la séparation de sources sont définis et un lien entre ces deux domaines est établi. Par la suite, plusieurs algorithmes itératifs de type Jacobi reposant sur une paramétrisation LU sont développés. Pour chacun des algorithmes, on propose de déterminer les matrices permettant de diagonaliser l'ensemble considéré par l'optimisation d'un critère inverse. On envisage la minimisation du critère selon deux approches : la première, de manière directe, et la seconde, en supposant que les éléments de l'ensemble considéré sont quasiment diagonaux. En ce qui concerne l'estimation des différents paramètres du problème, deux stratégies sont mises en œuvre : l'une consistant à estimer tous les paramètres indépendamment et l'autre reposant sur l'estimation indépendante de couples de paramètres spécifiquement choisis. Ainsi, pour la diagonalisation conjointe de matrices complexes symétriques ou hermitiennes, nous proposons trois algorithmes. Le premier repose sur la recherche des racines de la dérivée du critère, le second est basé sur une recherche de vecteur propre et le dernier utilise une méthode de gradient avec calcul du pas optimal d'adaptation. Dans le cadre de la diagonalisation conjointe d'ensembles de tenseurs admettant soit une décomposition symétrique, soit une décomposition INDSCAL, soit une décomposition non-symétrique, nous avons développé deux algorithmes. Pour chacun d'eux, le calcul des paramètres se fait par recherche des racines de la dérivée du critère considéré. Nous montrons aussi le lien existant entre la diagonalisation conjointe de tenseurs d'ordre trois et la décomposition canonique polyadique d'un tenseur d'ordre quatre, puis nous comparons les algorithmes développés à différentes méthodes de la littérature. Le bon comportement des algorithmes proposés est illustré au moyen de simulations numériques. Puis, ils sont validés dans le cadre de la séparation de sources de télécommunications numériques. Les ensembles à diagonaliser sont construits à l'aide de statistiques d'ordre supérieur calculées à partir des signaux d'observation.

Mots-clefs :

Analyse en composantes indépendantes ; Décomposition LU ; Diagonalisation conjointe ; Optimisation ; Séparation aveugle de sources ; Télécommunications numériques ; Tenseur d'ordre trois ; Traitement du signal.

Abstract

This thesis develops joint diagonalization of matrices and third-order tensors methods for MIMO source separation in the field of digital telecommunications. After a state of the art, the motivations and the objectives are presented. Then the joint diagonalisation and the blind source separation issues are defined and a link between both fields is established. Thereafter, five Jacobi-like iterative algorithms based on an LU parameterization are developed. For each of them, we propose to derive the diagonalization matrix by optimizing an inverse criterion. Two ways are investigated : minimizing the criterion in a direct way or assuming that the elements from the considered set are almost diagonal. Regarding the parameters derivation, two strategies are implemented : one consists in estimating each parameter independently, the other consists in the independent derivation of couple of well-chosen parameters. Hence, we propose three algorithms for the joint diagonalization of symmetric complex matrices or hermitian ones. The first one relies on searching for the roots of the criterion derivative, the second one relies on a minor eigenvector research and the last one relies on a gradient descent method enhanced by computation of the optimal adaptation step. In the framework of joint diagonalization of symmetric, INDSCAL or non symmetric third-order tensors, we have developed two algorithms. For each of them, the parameters derivation is done by computing the roots of the considered criterion derivative. We also show the link between the joint diagonalization of a third-order tensor set and the canonical polyadic decomposition of a fourth-order tensor. We confront both methods through numerical simulations. The good behavior of the proposed algorithms is illustrated by means of computing simulations. Finally, they are applied to the source separation of digital telecommunication signals. The sets to diagonalize are built using high-order statistics computed from observation signals.

Key words

Blind source separation; Independent component analysis; Joint diagonalization; LU decomposition; Digital telecommunications; Optimization; Signal processing; Third-order tensors.

Table des matières

Remerciements	i
Résumé	iii
Abstract	v
Liste des acronymes	xxi
1 Introduction générale	1
1.1 Cadre de la thèse	1
1.1.1 Diagonalisation conjointe de matrices	1
1.1.2 Décompositions tensorielles	4
1.2 Objectifs de la thèse et contenu du manuscrit	5
1.3 Notations et définitions	6
1.3.1 Notations classiques	6
1.3.2 Notations spécifiques	7
1.3.3 Opérations sur les matrices	8
1.3.4 Opérations sur les tenseurs	9
1.3.5 Définitions	9
1.4 Contributions	9
2 Formulation du problème et approches proposées	11
2.1 Diagonalisation conjointe de matrices	11
2.1.1 Ensemble de matrices complexes non-symétriques	11
2.1.2 Ensembles importants	13
2.2 Diagonalisation conjointe de tenseurs d'ordre trois	15
2.2.1 Ensemble de tenseurs complexes non-symétriques	15
2.2.2 Ensembles importants	17

2.3	Lien avec l'ICA et la séparation de sources	19
2.3.1	La séparation de sources	19
2.3.2	La diagonalisation conjointe comme méthode de séparation de sources .	21
2.4	Etat de l'art des méthodes de diagonalisation conjointe	24
2.4.1	Approches envisagées	26
2.4.2	Un problème d'optimisation	28
2.5	Méthodes proposées pour les algorithmes développés	32
2.5.1	Procédure de Jacobi	32
2.5.2	Critères choisis	33
2.5.3	Contrainte considérée	34
2.5.4	Structure générale des algorithmes proposés	36
3	Algorithmes de diagonalisation conjointe non-unitaire de matrices	39
3.1	Introduction	39
3.2	Stratégie d'estimation découplée	43
3.2.1	Approche classique	45
3.2.1.1	Modèle complexe symétrique	45
3.2.1.2	Modèle hermitien	46
3.2.2	Approche adaptée	47
3.2.2.1	Modèle complexe symétrique	48
3.2.2.2	Modèle hermitien	50
3.2.2.3	Description de l'algorithme	51
3.3	Stratégie d'estimation couplée	52
3.3.1	Approche classique	55
3.3.1.1	Modèle complexe symétrique	57
3.3.1.2	Modèle hermitien	61
3.3.2	Approche adaptée	62
3.3.2.1	Modèle complexe symétrique	63
3.3.2.2	Modèle hermitien	66
3.3.2.3	Etape de normalisation	68
3.3.2.4	Remarque	68
3.3.3	Description des algorithmes	69
3.4	Simulations	70
3.4.1	Modèle complexe symétrique	71
3.4.2	Modèle hermitien	79
3.4.3	Complexité algorithmique	87

4 Algorithmes de diagonalisation conjointe non-unitaire de tenseurs d'ordre trois	89
4.1 Introduction	89
4.2 Stratégie d'estimation découplée	92
4.2.1 Approche classique	95
4.2.2 Approche adaptée	98
4.2.3 Description des algorithmes	103
4.3 Simulations	104
4.3.1 Algorithmes de décomposition CP	105
4.3.2 Complexité algorithmique	106
4.3.3 Cadre de simulations	108
4.3.4 Influence des différents paramètres	110
4.3.5 Statistiques sur la vitesse de convergence	120
5 Application à la séparation de sources de télécommunications numériques	123
5.1 Séparation de sources	123
5.2 Cadre de simulations	124
5.3 Modulation QPSK et cas symétrique	127
5.3.1 Influence du rapport signal sur bruit	128
5.3.2 Influence du nombre de sources	131
5.3.3 Influence du nombre d'échantillons considérés	133
5.4 Modulation 4-ASK et cas hermitien/INDSCAL _h	136
5.4.1 Influence du rapport signal sur bruit	137
5.4.2 Influence du nombre de sources	140
5.4.3 Influence du nombre d'échantillons considérés	140
5.5 Conclusion	144
Conclusion et perspectives	145
Bibliographie	149
Article	161

Table des figures

2.1	Schéma classique d'un mélange linéaire instantané	20
2.2	Schéma de la séparation de source dans le cas non-bruité	20
2.3	Schéma de la séparation de sources dans le cadre général	21
2.4	Schéma de séparation de sources avec blanchiment des observations	21
2.5	Position des composantes non minimisées lors de la diagonalisation du tenseur \mathbf{T} pour $N = 10$, à gauche lors de la diagonalisation directe du tenseur, à droite en passant par la diagonalisation conjointe des matrices constituant les tranches de \mathbf{T} selon le troisième mode	25
3.1	Schéma de l'impact de la transformation (3.13) sur une matrice \mathbf{R}'_k	44
3.2	Schéma de l'impact de la transformation (3.20) sur une matrice $\mathbf{R}'_k^{(s)}$ lorsqu'on utilise l'approche adaptée.	49
3.3	Schéma de l'impact de la transformation (3.50) sur une matrice \mathbf{R}'_k	53
3.4	Schéma de l'impact de la transformation (3.50) sur une matrice \mathbf{R}'_k lorsqu'on utilise l'approche adaptée.	64
3.5	Indice de performance en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.	72
3.6	Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.	73
3.7	Indice de performance en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.	73
3.8	Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.	74

3.9	Indice de performance médian après convergence en fonction du niveau de bruit t pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ et pour 200 tirages indépendants.	75
3.10	Indice de performance médian après convergence en fonction de la dimension N_s des 15 matrices complexes symétriques 40×40 dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.	76
3.11	Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 40×40 avec $N_s = 30$ dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.	76
3.12	Indice de performance médian après convergence en fonction du nombre K de matrices complexes symétriques de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 200 tirages indépendants.	77
3.13	Indices de performance médians et moyens en fonction du nombre de balayages pour 2 matrices complexes symétriques de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 100 tirages indépendants.	77
3.14	Indices de performance médians et moyens en fonction du nombre de balayages pour 3 matrices complexes symétriques de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.	78
3.15	Indice de performance en fonction du nombre de balayages pour 3 matrices complexes symétriques de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.	79
3.16	Indice de performance en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.	80
3.17	Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.	81
3.18	Indice de performance en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.	81
3.19	Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.	82
3.20	Indice de performance médian après convergence en fonction du niveau de bruit t pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ et pour 200 tirages indépendants.	82

3.21	Indice de performance médian après convergence en fonction de la dimension N_s des 15 matrices hermitiennes 30×30 dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.	84
3.22	Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 30×30 avec $N_s = 30$ dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.	84
3.23	Indice de performance médian après convergence en fonction du nombre K de matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 200 tirages indépendants.	85
3.24	Indices de performance médians et moyens en fonction du nombre de balayages pour 2 matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 100 tirages indépendants.	86
3.25	Indices de performance médians et moyens en fonction du nombre de balayages pour 3 matrices hermitiennes de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.	86
3.26	Indice de performance en fonction du nombre de balayages pour 3 matrices hermitiennes de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.	87
4.1	Schéma de l'impact de la transformation (4.14) sur un tenseur $\mathbf{R}'(k)$	94
4.2	Schéma de l'impact de la transformation (4.22) sur un tenseur $\mathbf{R}'_1(k)$	96
4.3	Schéma de l'impact de la transformation (4.14) sur un tenseur $\mathbf{R}'(k)$ dans le cadre de l'approche adaptée.	100
4.4	Indice de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte non bruité.	111
4.5	Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte non bruité et pour 50 tirages indépendants.	111
4.6	Indice de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$	112
4.7	Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.	113

4.8	Indice de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-2}$. . .	113
4.9	Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.	114
4.10	Indice de performance médian après 500 itérations en fonction du niveau de bruit t pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ et pour 50 tirages indépendants.	114
4.11	Indice de performance moyen après 500 itérations en fonction du niveau de bruit t pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ et pour 50 tirages indépendants.	115
4.12	Indice de performance médian après 500 itérations en fonction du nombre K de tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.	116
4.13	Indice de performance moyen après 500 itérations en fonction du nombre K de tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.	116
4.14	Indices de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour 1 tenseur non-symétrique de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$	117
4.15	Indices de performance médians et moyens en fonction du nombre de balayages pour 1 tenseur non-symétrique de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.	117
4.16	Indice de performance médian après 500 itérations en fonction de la dimension réduite N_s des 15 tenseurs $32 \times 35 \times 40$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.	118
4.17	Indices de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $32 \times 35 \times 40$ avec $N_s = 30$ dans un contexte bruité $t = 10^{-4}$	119
4.18	Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $32 \times 35 \times 40$ avec $N_s = 30$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.	119

5.1	Indice de performance de chacun des algorithmes pour toutes les réalisations en fonction du nombre de balayages. Contexte : 7 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK et avec un RSB_{dB} de 20dB.	127
5.2	Indices de performance médians et moyens des algorithmes (50 réalisations) en fonction du nombre de balayages pour 7 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK et avec un RSB_{dB} de 20dB.	128
5.3	Indice de performance médian (50 réalisations) après 1000 balayages en fonction du RSB_{dB} pour 7 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK.	129
5.4	Vitesse de convergence médiane (50 réalisations) en fonction du RSB_{dB} pour 7 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK.	129
5.5	Exemple explicite de séparation de sources en fonction du RSB_{dB} pour 10 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 3 sources respectant une modulation QPSK. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, ALUJA et C-CLUJA	130
5.6	Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_s de sources de modulation QPSK. Contexte : 7 signaux d'observations de 4096 échantillons avec un RSB_{dB} de 20dB.	132
5.7	Vitesse de convergence médiane (50 réalisations) en fonction du nombre N_s de sources de modulation QPSK pour 18 signaux d'observations de 4096 échantillons avec un RSB_{dB} de 20dB.	132
5.8	Exemple explicite de séparation de sources en fonction du nombre N_s de sources respectant une modulation QPSK pour 10 signaux d'observations de 4096 échantillons avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, ALUJA et C-CLUJA	133
5.9	Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK avec un RSB_{dB} de 20dB.	134

5.10	Vitesse de convergence médiane (50 réalisations) en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK avec un RSB_{dB} de 20dB.	134
5.11	Exemple explicite de séparation de sources en fonction du nombre N_e d'échantillons pour 10 signaux d'observations constitués d'un mélange linéaire de 3 sources respectant une modulation QPSK avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, ALUJA et C-CLUJA	135
5.12	Indice de performance de chacun des algorithmes pour toutes les réalisations en fonction du nombre de balayages. Contexte : 8 signaux d'observations de 16384 échantillons, constitués d'un mélange linéaire de 7 sources respectant une modulation 4-ASK et avec un RSB_{dB} de 30dB.	136
5.13	Indices de performance médians et moyens des algorithmes (50 réalisations) en fonction du nombre de balayages pour 8 signaux d'observations de 16384 échantillons, constitués d'un mélange linéaire de 7 sources respectant une modulation 4-ASK et avec un RSB_{dB} de 30dB.	137
5.14	Indice de performance médian (50 réalisations) après 1000 balayages en fonction du RSB_{dB} pour 8 signaux d'observations de 16384 échantillons, constitués d'un mélange linéaire de 7 sources respectant une modulation 4-ASK.	138
5.15	Vitesse de convergence médiane (50 réalisations) en fonction du RSB_{dB} pour 8 signaux d'observations de 16384 échantillons, constitués d'un mélange linéaire de 7 sources respectant une modulation 4-ASK.	138
5.16	Exemple explicite de séparation de sources en fonction du RSB_{dB} pour 5 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 3 sources respectant une modulation 4-ASK. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, U-WEDGE et CPD-LM	139
5.17	Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_s de sources de modulation 4-ASK. Contexte : 15 signaux d'observations de 16384 échantillons avec un RSB_{dB} de 20dB.	141
5.18	Vitesse de convergence médiane (50 réalisations) en fonction du nombre N_s de sources de modulation 4-ASK pour 15 signaux d'observations de 16384 échantillons avec un RSB_{dB} de 20dB.	141

5.19 Exemple explicite de séparation de sources en fonction du nombre N_s de sources respectant une modulation 4-ASK pour 5 signaux d'observations de 4096 échantillons avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, U-WEDGE et CPD-LM 142

5.20 Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation 4-ASK avec un RSB_{dB} de 20dB. 143

5.21 Vitesse de convergence médiane (50 réalisations) en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation 4-ASK avec un RSB_{dB} de 20dB. . . . 143

5.22 Exemple explicite de séparation de sources en fonction du nombre N_e d'échantillons pour 5 signaux d'observations constitués d'un mélange linéaire de 3 sources respectant une modulation 4-ASK avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, U-WEDGE et CPD-LM 144

Liste des tableaux

3.1	Récapitulatifs des algorithmes matriciels de diagonalisation conjointe non-unitaire proposés selon l'approche liée au critère d'optimisation et la stratégie d'estimation des paramètres choisies	42
3.2	Paramètres de simulation pour les algorithmes de diagonalisation conjointe d'ensembles de matrices complexes symétriques	72
3.3	Quelques statistiques après convergence pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.	74
3.4	Paramètres de simulation pour les algorithmes de diagonalisation conjointe d'ensembles de matrices complexes hermitiennes	80
3.5	Quelques statistiques après convergence pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.	83
3.6	Récapitulatifs des complexités de chacun des algorithmes de diagonalisation conjointe de matrices proposés et utilisés dans ce chapitre	88
4.1	Récapitulatif des algorithmes tensoriels de diagonalisation conjointe non-unitaire proposés selon l'approche liée au critère d'optimisation et la stratégie d'estimation des paramètres choisis	92
4.2	Récapitulatif des complexités de chacun des algorithmes de diagonalisation conjointe de tenseurs proposés et de chacun des algorithmes de décomposition CP utilisés dans ce chapitre	107
4.3	Paramètres de simulation pour les algorithmes de diagonalisation conjointe d'ensembles de tenseurs complexes non-symétriques.	109
4.4	Quelques statistiques sur le nombre d'itérations nécessaires aux cinq algorithmes pour converger pour différents contextes de simulation (toutes les statistiques sont exprimées en itérations).	120

5.1	Numéro de figure, paramètres utilisés et courbes affichées pour l'ensemble des simulations liées à l'application à la séparation de sources de signaux de télécommunications numériques.	126
5.2	Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du RSB_{dB} pour 7 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les RSB_{dB}).	130
5.3	Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du nombre N_s de sources respectant une modulation QPSK. Contexte : 18 signaux d'observations de 4096 échantillons avec un RSB_{dB} de 20dB (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les N_s).	133
5.4	Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK avec un RSB_{dB} de 20dB (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les N_e).	135
5.5	Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du RSB_{dB} pour 8 signaux d'observations de 16384 échantillons, constitués d'un mélange linéaire de 7 sources respectant une modulation 4-ASK (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les RSB_{dB}).	139
5.6	Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation 4-ASK avec un RSB_{dB} de 20dB (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les N_e).	144

Liste des acronymes

4-ASK	4-Amplitude-Shift Keying
A-CLUJA	Adapted approach for Coupled LU Jacobi-like Algorithm
AC-DC	Aternating Columns - Diagonal Centers
ALS	Alternating Least Squares
ALUJA	Alternate LU Jacobi-like Algorithm
<i>c.à.d.</i>	<i>c'est-à-dire</i>
C-CLUJA	Combined method for Coupled LU Jacobi-like
Candecomp	Canonical decomposition
CJDi	Complex Joint Diagonalization
CP	Candecomp/Parafac ou Canonical Polyadic decomposition
DIEM	Diagonalization using Equivalent Matrices
DNJD	Distinctively Normalized Joint Diagonalization
EEG	Électro-EncéphaloGraphie
ELS	Enhanced Line Search
EJD	Exact Joint Diagonalization
EVD	EigenValue Value Decomposition
FAJD	Fast Approximate Joint Diagonalization
FFDiag	Fast Frobenius Diagonalization
G-CLUJA	Gradient descent method for Coupled LU Jacobi-like Algorithm
GEVD	Generalized EigenValue Decomposition
GSVD	Generalized Singular Value Decomposition
ICA	Independent Component Analysis
INDSCAL	INdividual Differences SCALing
JADE	Joint Approximate Diagonalization of Eigenmatrices
J-Di	Joint-Diagonalization
JTJD	Jacobi Tranformations for Joint Diagonalization
LM	Levenberg-Marquardt

LQCJD	LQ-based method for Complex non-orthogonal Joint Diagonalization
LS	Line Search
LUCJD	LU-based method for Complex non-orthogonal Joint Diagonalization
MIMO	Multiple Inputs Multiple Outputs
NCG	Non-linear Conjugate Gradient
NOODLES	NOn Orthogonal Decomposition of compLEx matrix Sets
PARAFAC	PARAllel FACtor analysis
PCA	Principal Component Analysis
QPSK	Quaternary Phase-Shift Keying
SOBI	Second Order Blind Identification
QDIAG	Quadratic DIAGonalization
SVD	Singular Value Decomposition
T-ALUJA	Tensorial ALUJA
T-ALUJA-A	T-ALUJA for an Adapted approach
U-WEDGE	Uniformly-Weighted Exhaustive Diagonalization with Gauss itErations

Introduction générale

1.1 Cadre de la thèse

1.1.1 Diagonalisation conjointe de matrices

La diagonalisation d'une matrice est un problème important en traitement du signal. En effet, on l'utilise notamment en analyse spectrale, en estimation de sous-espace signal ou bruit, en analyse en composantes principales, en réduction de dimension ou encore en blanchiment de données et en analyse en composantes indépendantes. De manière plus applicative, en traitement d'antennes, par exemple, à partir de l'analyse en composantes principales de la matrice de covariance des observations, on peut estimer les angles d'arrivée ainsi que le nombre de sources actives, autrement dit l'espace signal associé.

Cependant, il est parfois intéressant, voire nécessaire, de traiter un ensemble de matrices estimées à partir d'un même jeu de données. On parle alors de diagonalisation conjointe de matrices [24]. Elle se rencontre dans des problèmes pratiques comme la formation de faisceaux [14, 58, 60], le débruitage de signaux [5], l'égalisation aveugle de canaux pour des systèmes de télécommunications multi-entrées multi-sorties [32], l'extraction des différences de marche d'échos Doppler en radar [72], l'analyse en composantes indépendantes [3, 91] et la séparation aveugle de sources [31, 83]. Pour cette dernière, par exemple, on peut estimer une matrice de séparation en diagonalisant conjointement un ensemble de matrices cibles.

Les matrices cibles sont, de manière générale, construites à partir d'opérateurs d'analyse ou de statistiques. On trouve, notamment, des matrices liées à :

- des décompositions spectrales des données [43]
- des décompositions temps-fréquence [7, 8, 23, 47, 54, 53]
- des statistiques d'ordre deux (comme la covariance des signaux d'observation ou les covariances retardées) [6, 18, 49, 50, 77, 94, 116]
- des statistiques d'ordres supérieurs (comme les moments ou les cumulants d'ordre

supérieur) [4, 9, 10, 11, 12, 37, 48, 80, 81, 83, 88].

Comme ces matrices sont « seulement » construites à partir d'un nombre fini d'observations, la diagonalisation conjointe d'un ensemble de plus de deux matrices ne pourra pas être exacte, on l'appellera alors diagonalisation conjointe approchée. De par leurs différentes constructions, vues précédemment, les matrices cibles présentent souvent des propriétés de symétrie. Dans le cas réel, elles peuvent être symétriques tandis que dans le cas complexe, elles peuvent être soit hermitiennes, soit symétriques. Initialement, on considérait principalement les ensembles de matrices hermitiennes, mais récemment, un intérêt croissant s'est porté sur les ensembles de matrices complexes symétriques que l'on peut retrouver, notamment, en télécommunications numériques lorsque les signaux considérés sont non-circulaires [38, 69, 82, 91, 97]. Il existe aussi d'autres domaines d'applications où l'on peut considérer ce type de symétrie [100, 119]. Afin d'augmenter le nombre de matrices cibles, il est aussi possible de construire, à partir d'un jeu de données adéquat, un ensemble contenant des matrices complexes symétriques et hermitiennes [101, 118]. La diagonalisation conjointe de matrices possédant les types de symétrie énoncés précédemment est dite par congruence. Il est cependant important de noter que les matrices cibles ne présentent pas toujours ces types de symétries, c'est le cas notamment pour la diagonalisation conjointe par similitude [61, 73, 75] ou encore la diagonalisation conjointe de matrices cibles non-symétriques [21, 35, 34, 55].

Historiquement, les premières méthodes envisagées pour la diagonalisation d'une matrice ont été la décomposition en éléments propres et la décomposition en valeurs singulières. Par la suite, il est devenu important, voire nécessaire, de prendre en compte simultanément plusieurs matrices, notamment pour des problèmes de séparation de sources. La première méthode de diagonalisation conjointe considérée est la décomposition en éléments propres généralisée. Cette méthode ne prend en compte que deux matrices. Par la suite, il a été développé des méthodes de décomposition conjointe pour un nombre de matrices supérieur ou égal à deux. Dans un premier temps, le problème était posé de telle sorte que la matrice diagonalisante recherchée soit unitaire. Ainsi, la décomposition en éléments propres généralisée a été étendue pour un ensemble d'au moins deux matrices hermitiennes dans [14]. Cette méthode a ainsi donné lieu à l'un des tous premiers algorithmes de diagonalisation conjointe approchée, appliqué à des matrices hermitiennes de cumulants d'ordre quatre et connu sous le nom de *Joint Approximate Diagonalization of Eigenmatrices* (JADE) [14]. Un algorithme appliqué à des matrices hermitiennes de statistiques d'ordre deux, dénommé *Second Order Blind Identification* (SOBI), a aussi été développé dans [6]. Par la suite, JADE a été généralisé pour des statistiques d'ordre supérieur ou égal à deux dans [80]. Toujours dans le cadre des matrices unitaires, le cas des matrices complexes symétriques (obtenu à partir de cumulants d'ordre quatre) a été traité dans [39]. Tout problème de diagonalisation conjointe approchée peut être

ramené à un problème unitaire au moyen d'une étape de prétraitement, appelé blanchiment des données [29]. Toutefois, cette étape préliminaire limite en pratique les performances d'estimation [13, 67]. L'idée est donc de se défaire de ce prétraitement en développant des méthodes de diagonalisation conjointe approchée sans contraire unitaire. Récemment, de nombreux algorithmes de ce type ont été développés.

Pour diagonaliser conjointement un ensemble de matrices, il s'agit, la plupart du temps, d'optimiser un critère lié à la décomposition. Il existe différents types de critères dans la littérature. Parmi eux, le critère direct est une mesure de la différence entre les matrices cibles et leur modèle supposé au sens des moindres carrés. Ainsi, on pourra estimer directement la matrice de mélange [22, 118, 132]. Il existe aussi le critère indirect qui est une mesure de la diagonalité. On pourra ainsi estimer la matrice de séparation. Le critère indirect qui est très utilisé en séparation de sources, peut prendre de nombreuses formes comme celles d'un critère des moindres carrés classique [1, 51, 56, 73, 78, 80, 109, 126, 127] ou à invariance d'échelle [1]. On peut aussi considérer un critère mixte, combinant un critère direct et un critère inverse [114]. Enfin, l'approche par maximum de vraisemblance [70, 93] est réservée aux matrices définies positives. De nombreuses méthodes de diagonalisation conjointe approchée non-unitaire sont proposées dans la littérature. Parmi elles, certaines sont basées sur une estimation de la matrice diagonalisante complète [22, 114, 117, 118, 135] tandis que d'autres décomposent le problème de manière à pouvoir implémenter des méthodes itératives plus simples [70, 111, 132] comme, notamment, la procédure de type Jacobi [1, 15, 56, 78, 93, 109, 122, 126, 127]. De manière générale, les méthodes de type Jacobi possèdent l'avantage d'être relativement simples à implémenter. De plus, elles peuvent permettre une résolution analytique du problème ainsi qu'une parallélisation des calculs.

Concernant la séparation de sources, les applications sont très nombreuses. On retrouve, notamment, ce type de problème dans les domaines de l'astronomie [43], l'audio [20, 65, 123], le biomédical [33, 40, 59, 63], la sismique [92, 110, 125], la spectroscopie [87, 99] et des télécommunications numériques [18, 19, 25, 48]. Toujours dans le cadre de la séparation de sources, d'autres méthodes que la diagonalisation conjointe de matrices peuvent être mises en œuvre. Tout d'abord, une alternative à la diagonalisation conjointe est la déflation qui consiste à extraire les sources une à une du mélange [44, 45, 46]. On a aussi parfois recours à la bloc-diagonalisation conjointe. Cette dernière méthode s'applique lorsque, dans un problème, les sources ne sont pas toutes indépendantes, mais seulement indépendantes par groupe ou encore dans le cadre de modèles convolutifs. Récemment, de nombreux algorithmes de bloc-diagonalisation conjointe non-unitaire approchée ont été développés [47, 51, 52, 66, 90, 112]. Enfin, il est important de noter qu'un ensemble de matrices peut aussi être considéré comme un seul tenseur ou un ensemble de tenseurs.

1.1.2 Décompositions tensorielles

En traitement de signal, de nombreux problèmes peuvent être modélisés sous forme tensorielle [27, 30]. Ces tenseurs sont très souvent construits grâce aux différentes diversités des problèmes considérés. Tout d’abord, certaines données sources peuvent être directement générées sous forme de tenseurs. C’est le cas, par exemple, pour des images couleurs RGB ou hyperspectrales, pour des vidéos ou encore pour les affichages de champs lumineux [130]. De manière plus générale, tout signal multidimensionnel peut être stocké sous forme de tenseurs. Par exemple, en télécommunications sans-fil, les signaux ont des dépendances temporelles, spatiales et spectrales, ils pourront alors être stockés dans des tenseurs d’ordre trois [102]. Il en est de même en spectroscopie où le tenseur d’ordre trois contiendra d’une part les différents échantillons, d’autre part les longueurs d’onde avec lesquelles les échantillons sont excités et enfin les longueurs d’onde qu’ils émettent en réponse à cette excitation [103]. En électroencéphalographie, on peut considérer initialement un tenseur d’ordre trois contenant des informations temporelles et spectrales des signaux selon les différents canaux. Si cela s’avère insuffisant, on peut prendre en compte plus d’informations en considérant des diversités supplémentaires comme les différents sujets d’étude (patients), les différentes conditions d’examen ainsi que les différents essais. Ainsi on augmentera l’ordre du tenseur [86]. Finalement, le tenseur de stockage pourrait être ici d’ordre six. Enfin, les tenseurs peuvent être construits à partir d’opérateurs d’analyse. En effet, à partir des données on peut obtenir directement des tenseurs, à l’aide de statistiques d’ordre supérieur [89]. Dans le cadre de procédures d’analyse électroencéphalographique multicanale, des diversités supplémentaires pourront être considérées à l’aide des représentations temps-fréquence ou des transformées en ondelettes [28, 79].

En séparation de sources, par exemple, on cherche à décomposer ces tenseurs de manière à estimer une matrice de séparation ou la contribution des différentes sources, à faire de la réduction de dimension ou du débruitage. Pour traiter ce type de problème, certaines méthodes consistent à considérer le tenseur cible comme un ensemble de matrices cibles. On retombe ainsi sur un problème de diagonalisation conjointe de matrices. Cependant, même si cette méthode offre l’intérêt de pouvoir utiliser des algorithmes de diagonalisation conjointe de matrices, elle nécessite une phase de dépliement du tenseur parfois peu intuitive. Il a aussi été proposé différentes méthodes de décomposition tensorielle, comme, par exemple, la décomposition canonique polyadique (aussi connue sous le nom de factorisation parallèle) appliquée à la psychométrie dans [16] et la linguistique [57] ou encore la décomposition de Tucker introduite en psychométrie [120, 121]. Ces décompositions n’ont toutefois pas la même utilité. En traitement de signal, la décomposition canonique polyadique est, en général, utilisée pour factoriser des données qui seront ainsi facilement interprétables. C’est typiquement le

cas en séparation de sources où cette décomposition permettra de révéler la contribution des différentes sources. La décomposition de Tucker, quant à elle, est plutôt utilisée pour de la compression de données. Ces méthodes ont donné lieu au développement de nombreux algorithmes [41, 74, 95, 96, 98, 99].

Tout comme pour les matrices, il peut être intéressant, voire nécessaire, de considérer des ensembles de tenseurs. On parlera alors de diagonalisation conjointe de tenseurs ou encore de décomposition canonique polyadique conjointe. Pour obtenir de tels ensembles de tenseurs, on peut, par exemple, choisir astucieusement une des variables d'un problème d'ordre N et considérer pour chaque valeur que prend cette variable un tenseur d'ordre $N - 1$. Tout comme dans le cas matriciel, la diagonalisation conjointe de tenseurs est basée sur l'optimisation d'un critère. Les critères considérés sont similaires à ceux utilisés pour des ensembles de matrices mais adaptés pour des tenseurs. Les ensembles à diagonaliser conjointement peuvent être constitués de tenseurs présentant des symétries [17, 64], ou non. La diagonalisation conjointe approchée unitaire de tenseurs d'ordre trois a déjà été traitée dans [41, 80]. En ce qui concerne la diagonalisation conjointe approchée non-unitaire de tenseurs, même si celle-ci est à mettre en lien avec la décomposition canonique polyadique, elle ne semble pas encore avoir fait l'objet de travaux en tant que telle.

1.2 Objectifs de la thèse et contenu du manuscrit

L'objectif principal de cette thèse est de développer des algorithmes de diagonalisation conjointe non-unitaire pour des ensembles de matrices et de tenseurs d'ordre trois complexes. Ces algorithmes doivent être performants et se comparer favorablement à ceux de la littérature. Ils doivent fonctionner pour toutes les symétries envisagées. A cette fin, nous avons décidé, dans un premier temps, de développer des algorithmes de diagonalisation conjointe non-unitaire de matrices complexes, dans le but d'adapter les plus appropriés à la diagonalisation conjointe non-unitaire de tenseurs d'ordre trois. Tous ces algorithmes seront ensuite appliqués au domaine de la séparation de sources de télécommunications numériques.

Dans le chapitre 2, nous présenterons le problème de la diagonalisation conjointe approchée non-unitaire en lien avec l'analyse en composantes indépendantes. Dans un premier temps, nous montrerons comment se formule le problème dans le cas matriciel puis dans le cas tensoriel. Puis, nous établirons le lien entre l'analyse en composantes indépendantes, la séparation de sources et la diagonalisation conjointe. Par la suite, nous ferons un état de l'art des méthodes de diagonalisation conjointe sans contrainte unitaire. Enfin, nous présenterons les approches choisies pour résoudre ce problème, à savoir une procédure de type Jacobi, avec une paramétrisation LU des matrices de séparation, une estimation de ces dernières de type

couplée ou découplée et des critères inverses de type moindres carrés.

Dans le chapitre 3, nous présenterons les algorithmes de diagonalisation conjointe non-unitaire de matrices que nous avons développés au cours de cette thèse dans les cas d'ensembles de matrices complexes symétriques ou hermitiennes. Nous commencerons par exposer la méthode d'estimation découplée avec un critère adapté (un algorithme ayant déjà été développé pour le critère inverse classique [1]). Ensuite, nous mettrons en lumière nos algorithmes utilisant une méthode d'estimation couplée dans un premier temps avec le critère inverse classique puis avec celui adapté. Enfin, des simulations numériques nous permettront de valider les performances des différents algorithmes de diagonalisation conjointe non-unitaire de matrices développés, tout en les comparant à des algorithmes de l'état de l'art.

Le chapitre 4 traitera des algorithmes que nous avons développés pour la diagonalisation conjointe non-unitaire de tenseurs d'ordre trois avec une méthode de type découplé et pour les différentes symétries possibles des tenseurs cibles. Dans un premier temps, nous présenterons l'algorithme basé sur l'optimisation d'un critère inverse classique puis d'un critère inverse adapté. Nous décrirons ensuite quelques algorithmes de décomposition canonique polyadique (aussi connue sous les noms Candecomp et PARAFAC) puis nous les comparerons aux algorithmes de diagonalisation conjointe de tenseurs d'ordre trois proposés au moyen de simulations numériques.

Dans le chapitre 5, nous appliquerons les algorithmes développés dans le cadre de la séparation aveugle multi-entrées multi-sorties de sources télécommunications numériques. Nous rappellerons succinctement les modèles utilisés en séparation aveugle de sources, puis nous montrerons les performances des algorithmes développés pour différents types de signaux de télécommunications.

1.3 Notations et définitions

1.3.1 Notations classiques

Le corps des nombres réels sera noté \mathbb{R} et celui des nombres complexes sera noté \mathbb{C} .

Soit X une variable aléatoire discrète pouvant prendre un nombre fini K de valeurs x_1, \dots, x_K avec les probabilités respectives p_1, \dots, p_K , on définit l'espérance mathématique de X de la manière suivante

$$E(X) = \sum_{k=1}^K x_k p_k \quad \text{avec} \quad \sum_{k=1}^K p_k = 1.$$

Si X est centrée, alors on a $E(X) = 0$.

Soit A, B, C et D quatre variables aléatoires discrètes et centrées, un des cumulants d'ordre quatre de ces variables aléatoires peut être défini comme

$$\text{Cum}(A, B, C, D) = \text{E}(ABCD) - \text{E}(AB)\text{E}(CD) - \text{E}(AC)\text{E}(BD) - \text{E}(AD)\text{E}(BC).$$

Dans ce manuscrit, nous noterons les scalaires en minuscule x , les vecteurs en minuscule et en gras \mathbf{v} et les matrices et les tenseurs en majuscule et en gras \mathbf{M} .

L'estimation de la variable x sera notée \hat{x} .

Le i -ème élément d'un vecteur sera noté v_i , la composante (i, j) d'une matrice sera notée M_{ij} et la composante (i, j, k) d'un tenseur d'ordre trois sera notée T_{ijk} .

$(\cdot)^T$ est l'opérateur transposé.

$(\cdot)^*$ est l'opérateur conjugué.

$(\cdot)^H$ est l'opérateur transposé-conjugué.

$(\cdot)^\dagger$ est l'opérateur correspondant soit à $(\cdot)^T$, soit à $(\cdot)^H$.

$(\cdot)^{-1}$ est l'opérateur inverse.

$(\cdot)^\ddagger$ est l'opérateur pseudo-inverse de Moore-Penrose.

Pour un nombre complexe c , sa partie réelle sera notée $\Re(c)$, sa partie imaginaire sera notée $\Im(c)$ et son module sera noté $|c|$.

1.3.2 Notations spécifiques

Données des problèmes :

N_s est le nombre de signaux d'intérêt.

N_o est le nombre d'observations.

K est le nombre de matrices ou de tenseurs à diagonaliser.

N_e est le nombre d'échantillons.

n_{sweeps} est le nombre de balayages ou d'itérations.

n_{draws} est le nombre de tirages ou de réalisations.

Ensembles :

On définit les ensembles

$$\mathcal{N}_s = \{1, \dots, N_s\} \quad \text{et} \quad \mathcal{K} = \{1, \dots, K\}.$$

Pour tout $(i, j) \in \mathcal{N}_s^2$, $i < j$, on définit aussi les ensembles suivants :

$$\mathcal{P} = \mathcal{N}_s \setminus \{i\}, \quad \mathcal{P}' = \mathcal{N}_s \setminus \{j\}, \quad \mathcal{Q} = \mathcal{N}_s \setminus \{i, j\}, \quad \mathcal{S} = \mathcal{N}_s^2 \setminus \{(i, i)\} \quad \text{et} \quad \mathcal{S}' = \mathcal{N}_s^2 \setminus \{(j, j)\}.$$

Matrices et tenseurs d'ordre trois :

A est la matrice facteur ou la matrice de mélange.

B est la matrice diagonalisante ou la matrice de séparation.

D est une matrice diagonale.

P est une matrice de permutation.

L est une matrice triangulaire inférieure dont les composantes diagonales valent un.

U est une matrice triangulaire supérieure dont les composantes diagonales valent un.

I_N est la matrice identité de dimensions $N \times N$.

Pour tout $k \in \mathcal{K}$, on définit les matrices et tenseurs suivants :

M_k est la k -ième matrice de l'ensemble de matrices à diagonaliser.

R_k est la k -ième matrice réduite de l'ensemble de matrices à diagonaliser.

T(k) est le k -ième tenseur d'ordre trois de l'ensemble de tenseurs à diagonaliser.

R(k) est le k -ième tenseur réduit d'ordre trois de l'ensemble de tenseurs à diagonaliser.

1.3.3 Opérations sur les matrices

Soit une matrice $\mathbf{M} \in \mathbb{C}^{N \times N}$, l'opérateur $\text{Diag}\{\cdot\}$ appliqué à la matrice \mathbf{M} va annuler ses termes hors-diagonaux de telle sorte que

$$\text{Diag}\{\mathbf{M}\} = \begin{pmatrix} M_{11} & 0 & \cdots & \cdots & 0 \\ 0 & M_{22} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & M_{NN} \end{pmatrix}.$$

L'opérateur $\text{ZDiag}\{\cdot\}$ appliqué à la matrice \mathbf{M} va annuler ses termes diagonaux de telle sorte que

$$\text{ZDiag}\{\mathbf{M}\} = \mathbf{M} - \text{Diag}\{\mathbf{M}\}.$$

$\det(\mathbf{M})$ représente le déterminant de la matrice \mathbf{M} .

1.3.4 Opérations sur les tenseurs

Soit un tenseur d'ordre trois $\mathbf{T} \in \mathbb{C}^{N \times N \times N}$, l'opérateur $\text{TDiag}\{\cdot\}$ appliqué au tenseur \mathbf{T} va annuler ses termes hors-diagonaux de telle sorte que

$$\text{TDiag}\{\mathbf{T}\}_{ijk} = \mathbf{T}_{ijk} \delta_{ijk}, \quad (i, j, k) \in \{1, \dots, N\}^3$$

avec

$$\begin{aligned} \delta_{ijk} &= 1 \quad \text{si } i = j = k, \\ &= 0 \quad \text{sinon.} \end{aligned}$$

L'opérateur $\text{ZTDiag}\{\cdot\}$ appliqué au tenseur \mathbf{T} va annuler ses termes diagonaux de telle sorte que

$$\text{ZTDiag}\{\mathbf{T}\} = \mathbf{T} - \text{TDiag}\{\mathbf{T}\}.$$

1.3.5 Définitions

Norme de Frobenius :

- La norme de Frobenius $\|\mathbf{M}\|$ d'une matrice \mathbf{M} est définie par

$$\|\mathbf{M}\| = \sqrt{\sum_{i,j} |M_{ij}|^2}. \quad (1.1)$$

- La norme de Frobenius $\|\mathbf{T}\|$ d'un tenseur d'ordre trois \mathbf{T} est définie par

$$\|\mathbf{T}\| = \sqrt{\sum_{i,j,k} |T_{ijk}|^2}. \quad (1.2)$$

Gradient :

L'opérateur gradient est noté $\nabla(\cdot)$. Pour une fonction f de deux variables complexes x et y , le gradient est défini comme

$$\nabla f(x, y) = \begin{pmatrix} \frac{\partial f(x, y)}{\partial x^*} \\ \frac{\partial f(x, y)}{\partial y^*} \end{pmatrix}. \quad (1.3)$$

1.4 Contributions

Une partie des travaux effectués dans cette thèse a donné lieu aux publications suivantes :

- [A1] V. Maurandi and E. Moreau, “A Decoupled Jacobi-Like Algorithm for Non-Unitary Joint Diagonalization of Complex-Valued Matrices”, *IEEE Signal Processing Letters*, Vol. 21, No. 12, pp. 1453–1456, December 2014.
- [CI1] V. Maurandi, C. De Luigi and E. Moreau, “Fast Jacobi like algorithms for joint diagonalization of complex symmetric matrices”, in *Proc. EUSIPCO’2013*, Marrakech, Morocco, September 2013.
- [CI2] V. Maurandi, E. Moreau and C. De Luigi, “Jacobi like algorithm for non-orthogonal joint diagonalization of Hermitian matrices”, in *Proc. ICASSP’2014*, Florence, Italy, May 2014.
- [CI3] V. Maurandi and E. Moreau, “A new fast Jacobi-like algorithm for non-orthogonal joint diagonalization of real-valued matrices based on a QR parameterization”, in *Proc. MLSP’2014*, Reims, France, September 2014.
- [CI4] V. Maurandi and E. Moreau, “Fast Non-Unitary Simultaneous Diagonalization of Third-Order Tensors”, in *Proc. Asilomar’2014*, Pacific Grove, CA, USA, November 2014.
- [CI5] V. Maurandi and E. Moreau, “Fast Jacobi Algorithm for Non-Orthogonal Joint Diagonalization of Non-Symmetric Third-Order Tensors”, in *Proc. EUSIPCO’2015*, Nice, France, August 2015.
- [CN1] V. Maurandi, C. De Luigi et E. Moreau, “Diagonalisation conjointe non-orthogonale de matrices complexes basée sur une décomposition LU”, dans *les actes du Grets’2013*, Brest, France, Septembre 2013.
- [CN2] V. Maurandi and E. Moreau, “Algorithme de Jacobi rapide pour la diagonalisation conjointe non-unitaire de tenseurs d’ordre trois”, dans *les actes du Grets’2015*, Lyon, France, Septembre 2015.

Formulation du problème et approches proposées

Dans ce chapitre, nous décrivons le problème de la diagonalisation conjointe non-unitaire d'ensembles de matrices et de tenseurs d'ordre trois, puis nous faisons le lien avec l'analyse en composantes indépendantes et la séparation de sources. Par la suite, nous faisons un état de l'art des méthodes de diagonalisation sans contrainte unitaire et enfin nous présentons les méthodes mises en œuvre pour les algorithmes développés au cours de cette thèse.

2.1 Diagonalisation conjointe de matrices

Nous abordons ici la diagonalisation conjointe non-unitaire d'ensembles de matrices complexes, tout d'abord dans le cadre général non-symétrique, puis nous nous intéressons à deux cas particuliers qui s'avèrent importants, notamment en séparation de sources de télécommunications numériques : les ensembles de matrices complexes symétriques et hermitiennes.

2.1.1 Ensemble de matrices complexes non-symétriques

On considère un ensemble $\mathcal{M}^{(g)}$ de K ($K \in \mathbb{N}^* \setminus \{1\}$) matrices complexes non-symétriques de dimensions $N_{o_1} \times N_{o_2}$, avec $(N_{o_1}, N_{o_2}) \in \{\mathbb{N}^* \setminus \{1\}\}^2$, défini de la manière suivante

$$\mathcal{M}^{(g)} = \{\mathbf{M}_k^{(g)} \in \mathbb{C}^{N_{o_1} \times N_{o_2}}, k \in \mathcal{K}\}. \quad (2.1)$$

avec $\mathcal{K} = \{1, \dots, K\}$. On suppose que les matrices $\mathbf{M}_k^{(g)} \in \mathcal{M}^{(g)}$ respectent la décomposition suivante

$$\mathbf{M}_k^{(g)} = \mathbf{A}_1 \mathbf{D}_k^{(g)} \mathbf{A}_2^\dagger \quad (2.2)$$

où $(\cdot)^\dagger$ correspond soit à l'opérateur transposé, soit à l'opérateur transposé-conjugué. Les matrices $\mathbf{D}_k^{(g)} \in \mathbb{C}^{N_s \times N_s}$ ($N_s \in \mathbb{N}^* \setminus \{1\}$) sont diagonales et non nulles pour tout $k \in \mathcal{K}$. $\mathbf{A}_1 \in \mathbb{C}^{N_{o_1} \times N_s}$ et $\mathbf{A}_2 \in \mathbb{C}^{N_{o_2} \times N_s}$ sont des matrices de rang plein en colonne, on les nomme matrices facteurs. Définissons l'ensemble générique $\mathcal{A}_{N \times M}$ tel que

$$\forall (M, N) \in \{\mathbb{N}^* \setminus \{1\}\}^2, M \leq N, \quad \mathcal{A}_{N \times M} = \{\mathbf{A} \in \mathbb{C}^{N \times M} \mid \mathbf{A} \text{ de rang plein en colonnes}\}. \quad (2.3)$$

Ainsi $\mathbf{A}_1 \in \mathcal{A}_{N_{o_1} \times N_s}$ et $\mathbf{A}_2 \in \mathcal{A}_{N_{o_2} \times N_s}$. Dans toute la suite, on suppose que $N_s \leq \min(N_{o_1}, N_{o_2})$ où $\min(\cdot)$ donne le minimum parmi les éléments en argument.

Le but de la diagonalisation conjointe de matrices est, à partir du seul ensemble $\mathcal{M}^{(g)}$, d'estimer des matrices diagonalisantes $\mathbf{B}_1 \in \mathbb{C}^{N_s \times N_{o_1}}$ et $\mathbf{B}_2 \in \mathbb{C}^{N_s \times N_{o_2}}$ de rang plein en ligne telles que les matrices transformées

$$\mathbf{M}'_k^{(g)} = \mathbf{B}_1 \mathbf{M}_k^{(g)} \mathbf{B}_2^\dagger \quad (2.4)$$

soient toutes diagonales. Idéalement, $\mathbf{M}'_k^{(g)}$ devront être égales à $\mathbf{D}_k^{(g)}$. En effet d'après (2.4) et (2.2), on aura

$$\mathbf{D}_k^{(g)} = \mathbf{B}_1 \mathbf{A}_1 \mathbf{D}_k^{(g)} (\mathbf{B}_2 \mathbf{A}_2)^\dagger, \quad (2.5)$$

avec

$$\mathbf{B}_1 \mathbf{A}_1 = \mathbf{I}_{N_s} \quad \text{et} \quad \mathbf{B}_2 \mathbf{A}_2 = \mathbf{I}_{N_s}. \quad (2.6)$$

Les matrices \mathbf{B}_1 et \mathbf{B}_2 sont donc idéalement les pseudo-inverses de Moore-Penrose respectives de \mathbf{A}_1 et \mathbf{A}_2 ,

$$\mathbf{B}_1 = \mathbf{A}_1^\ddagger \quad \text{et} \quad \mathbf{B}_2 = \mathbf{A}_2^\ddagger. \quad (2.7)$$

Définissons l'ensemble générique $\mathcal{B}_{M \times N}$ comme il suit

$$\forall (M, N) \in \{\mathbb{N}^* \setminus \{1\}\}^2, M \leq N, \quad \mathcal{B}_{M \times N} = \{\mathbf{B} \in \mathbb{C}^{M \times N} \mid \exists \mathbf{A} \in \mathcal{A}_{N \times M} \text{ telle que } \mathbf{B} = \mathbf{A}^\ddagger\}. \quad (2.8)$$

Ainsi $\mathbf{B}_1 \in \mathcal{B}_{N_s \times N_{o_1}}$ et $\mathbf{B}_2 \in \mathcal{B}_{N_s \times N_{o_2}}$.

Remarque : Comme les matrices $\mathbf{D}_k^{(g)}$ sont diagonales pour tout k , il existe alors un lien direct entre la relation (2.2) et la décomposition CP d'un tenseur complexe d'ordre trois. En effet, (2.2) peut s'écrire comme la décomposition terme à terme suivante

$$M_{k, i_1 i_2}^{(g)} = \sum_{j=1}^{N_s} D_{k, j j}^{(g)} A_{1, i_1 j} A_{2, i_2 j}^{(*)} \quad (2.9)$$

où $A_{2,i_2j}^{(*)} = A_{2,i_2j}$ quand $\mathbf{A}_2^\dagger = \mathbf{A}_2^T$ dans (2.2) et $A_{2,i_2j}^{(*)} = A_{2,i_2j}^*$ quand $\mathbf{A}_2^\dagger = \mathbf{A}_2^H$. L'expression (2.9) peut directement être lue comme la décomposition CP suivante

$$M_{i_1 i_2 k}^{''(g)} = \sum_{j=1}^{N_s} A_{1,i_1j} A_{2,i_2j}^{(*)} A_{3,kj} \quad (2.10)$$

où $A_{3,kj} = D_{k,jj}^{(g)}$ et $M_{i_1 i_2 k}^{''(g)} = M_{k,i_1 i_2}^{(g)}$ pour tout $i_1 \in \{1, \dots, N_{o_1}\}$, $i_2 \in \{1, \dots, N_{o_2}\}$ et $k \in \mathcal{K}$. L'expression (2.10) est directement la décomposition CP du tenseur d'ordre trois $\mathbf{M}^{''(g)} = (M_{i_1 i_2 k}^{''(g)})$.

2.1.2 Ensembles importants

Dans le cadre des télécommunications numériques, par exemple, on considère généralement des ensembles particuliers de matrices : les ensembles complexes symétriques et hermitiens. Ces deux types d'ensembles sont constitués de matrices carrées de dimensions $N_o \times N_o$ ($N_o \in \mathbb{N}^* \setminus \{1\}$) présentant des symétries particulières. Ainsi pour les deux modèles, nous aurons $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}$ avec $\mathbf{A} \in \mathcal{A}_{N_o \times N_o}$.

Ensemble de matrices complexes symétriques : On considère un ensemble $\mathcal{M}^{(s)}$ de K matrices complexes symétriques

$$\mathcal{M}^{(s)} = \{\mathbf{M}_k^{(s)} \in \mathbb{C}^{N_o \times N_o}, k \in \mathcal{K}\}. \quad (2.11)$$

Les matrices de cet ensemble admettent chacune la décomposition

$$\mathbf{M}_k^{(s)} = \mathbf{A} \mathbf{D}_k^{(s)} \mathbf{A}^T \quad (2.12)$$

où $\mathbf{A} \in \mathcal{A}_{N_o \times N_o}$ et les matrices $\mathbf{D}_k^{(s)} \in \mathbb{C}^{N_o \times N_o}$ sont toutes diagonales et non nulles. De façon similaire au cas général (paragraphe 2.1.1), pour diagonaliser cet ensemble de matrices, le but est d'estimer une matrice $\mathbf{B} \in \mathcal{B}_{N_s \times N_o}$ (2.8) telle que les matrices transformées

$$\mathbf{M}_k^{'(s)} = \mathbf{B} \mathbf{M}_k^{(s)} \mathbf{B}^T \quad (2.13)$$

soient toutes diagonales. Et idéalement, on aura la relation

$$\mathbf{B} = \mathbf{A}^\dagger. \quad (2.14)$$

Ensemble de matrices hermitiennes : On considère un ensemble $\mathcal{M}^{(h)}$ de K matrices hermitiennes

$$\mathcal{M}^{(h)} = \{\mathbf{M}_k^{(h)} \in \mathbb{C}^{N_o \times N_o}, k \in \mathcal{K}\}. \quad (2.15)$$

Les matrices de cet ensemble admettent chacune la décomposition

$$\mathbf{M}_k^{(h)} = \mathbf{A} \mathbf{D}_k^{(h)} \mathbf{A}^H \quad (2.16)$$

où $\mathbf{A} \in \mathcal{A}_{N_o \times N_s}$ et $\mathbf{D}_k^{(h)} \in \mathbb{R}^{N_s \times N_s}$ sont toutes diagonales et non nulles. Pour que les matrices $\mathbf{M}_k^{(h)}$ présentent une symétrie hermitienne, *c.à.d.* $M_{k,ji}^{(h)} = M_{k,ij}^{(h)*}$, il est nécessaire que les matrices $\mathbf{D}_k^{(h)}$ soient à valeurs réelles. Toutefois, on peut étendre ce modèle pour des matrices $\mathbf{D}_k^{(h)} \in \mathbb{C}^{N_s \times N_s}$ en opérant au préalable sur l'ensemble $\mathcal{M}^{(h)}$. En effet, dans cette configuration, on ne s'intéressera qu'à la partie hermitienne des matrices $\mathbf{M}_k^{(h)}$ définie comme

$$\mathbf{M}_{k,herm}^{(h)} = \frac{\mathbf{M}_k^{(h)} + \mathbf{M}_k^{(h)H}}{2}. \quad (2.17)$$

En effet, d'après (2.16) si $\mathbf{D}_k^{(h)} \in \mathbb{C}^{N_s \times N_s}$ alors on peut écrire

$$\begin{aligned} \mathbf{M}_k^{(h)} &= \mathbf{A} (\Re(\mathbf{D}_k^{(h)}) + \imath \Im(\mathbf{D}_k^{(h)})) \mathbf{A}^H \\ &= \mathbf{A} \Re(\mathbf{D}_k^{(h)}) \mathbf{A}^H + \imath \mathbf{A} \Im(\mathbf{D}_k^{(h)}) \mathbf{A}^H \\ &= \mathbf{M}_{k,herm}^{(h)} + \imath \mathbf{M}_{k,aherm}^{(h)} \end{aligned} \quad (2.18)$$

où $\imath^2 = -1$ et $\mathbf{M}_{k,herm}^{(h)}$ et $\mathbf{M}_{k,aherm}^{(h)}$ sont respectivement les parties hermitiennes et anti-hermitiennes des $\mathbf{M}_k^{(h)}$, avec

$$\mathbf{M}_{k,aherm}^{(h)} = \frac{\mathbf{M}_k^{(h)} - \mathbf{M}_k^{(h)H}}{2\imath}. \quad (2.19)$$

De (2.18), on déduit simplement

$$\begin{aligned} \mathbf{M}_k^{(h)H} &= \mathbf{A} (\Re(\mathbf{D}_k^{(h)}) - \imath \Im(\mathbf{D}_k^{(h)})) \mathbf{A}^H \\ &= \mathbf{M}_{k,herm}^{(h)} - \imath \mathbf{M}_{k,aherm}^{(h)} \end{aligned} \quad (2.20)$$

Ainsi avec (2.18) et (2.20), on retrouve bien (2.17) et on obtient bien un ensemble de matrices hermitiennes sans pour autant avoir modifié \mathbf{A} .

Encore une fois, de façon similaire au cas général (paragraphe 2.1.1), pour diagonaliser l'ensemble de matrices $\mathcal{M}^{(h)}$, le but est d'estimer une matrice $\mathbf{B} \in \mathcal{B}_{N_s \times N_o}$ (2.8) telle que les

matrices transformées

$$\mathbf{M}'_k^{(h)} = \mathbf{B}\mathbf{M}_k^{(h)}\mathbf{B}^H \quad (2.21)$$

soient toutes diagonales. Et idéalement, on aura exactement (2.14).

Remarque : De manière générale, la diagonalisation conjointe s'applique aussi à des ensembles de matrices réelles. Pour cela, il suffit de remplacer toutes les matrices complexes du paragraphe 2.1 par des matrices réelles de mêmes dimensions et de même nature. Ainsi pour la diagonalisation conjointe d'un ensemble de matrices réelles non-symétriques, on se référera au paragraphe 2.1.1 et pour des matrices symétriques au paragraphe 2.1.2 (avec des matrices réelles, les cas complexes symétriques et hermitiens sont strictement équivalents).

2.2 Diagonalisation conjointe de tenseurs d'ordre trois

Nous abordons ici la diagonalisation conjointe non-unitaire d'ensembles de tenseurs d'ordre trois complexes, tout d'abord dans le cadre général non-symétrique (proche d'une décomposition CP classique), puis nous nous intéressons aux ensembles de tenseurs d'ordre trois symétriques et ceux admettant une décomposition INDSCAL (pour *Individual Differences SCALing*) [16]. Ces deux derniers ensembles sont très utilisés, notamment dans le domaine de la séparation de sources de télécommunications numériques.

2.2.1 Ensemble de tenseurs complexes non-symétriques

On considère un ensemble $\mathcal{T}^{(g)}$ de K ($K \in \mathbb{N}^*$) tenseurs d'ordre trois complexes non-symétriques de dimensions $N_{o_1} \times N_{o_2} \times N_{o_3}$, avec $(N_{o_1}, N_{o_2}, N_{o_3}) \in \{\mathbb{N}^* \setminus \{1\}\}^3$, défini de la manière suivante

$$\mathcal{T}^{(g)} = \{\mathbf{T}^{(g)}(k) \in \mathbb{C}^{N_{o_1} \times N_{o_2} \times N_{o_3}}, k \in \mathcal{K}\}. \quad (2.22)$$

On suppose que les tenseurs $\mathbf{T}^{(g)}(k) \in \mathcal{T}^{(g)}$ respectent la décomposition terme à terme suivante

$$T_{i_1 i_2 i_3}^{(g)}(k) = \sum_{j_1, j_2, j_3} D_{j_1 j_2 j_3}^{(g)}(k) A_{1, i_1 j_1} A_{2, i_2 j_2} A_{3, i_3 j_3} \quad (2.23)$$

où les matrices facteurs $\mathbf{A}_1 \in \mathcal{A}_{N_{o_1} \times N_s}$, $\mathbf{A}_2 \in \mathcal{A}_{N_{o_2} \times N_s}$ et $\mathbf{A}_3 \in \mathcal{A}_{N_{o_3} \times N_s}$ (2.3) sont de rang plein en colonne et pour tout $k \in \mathcal{K}$, les tenseurs $\mathbf{D}^{(g)}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ sont non nuls et diagonaux *c.à.d.*

$$D_{j_1 j_2 j_3}^{(g)}(k) = D_{j_1 j_2 j_3}^{(g)}(k) \delta_{j_1 j_2 j_3} \quad (2.24)$$

où

$$\begin{aligned} \delta_{j_1 j_2 j_3} &= 1 \quad \text{si } j_1 = j_2 = j_3 \\ &= 0 \quad \text{sinon.} \end{aligned} \quad (2.25)$$

Par la suite, par volonté de simplification, on notera la relation (2.23) de la façon suivante

$$\mathbf{T}^{(g)}(k) = \mathbf{D}^{(g)}(k) \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \mathbf{A}_3. \quad (2.26)$$

Dans toute la suite, on suppose que $N_s \leq \min(N_{o_1}, N_{o_2}, N_{o_3})$.

Le but de la diagonalisation conjointe de tenseurs d'ordre trois est, à partir du seul ensemble $\mathcal{T}^{(g)}(k)$, d'estimer des matrices diagonalisantes $\mathbf{B}_1 \in \mathcal{B}_{N_s \times N_{o_1}}$, $\mathbf{B}_2 \in \mathcal{B}_{N_s \times N_{o_2}}$ et $\mathbf{B}_3 \in \mathcal{B}_{N_s \times N_{o_3}}$ (2.8) de rang plein en ligne telles que les tenseurs transformés

$$\mathbf{T}'^{(g)}(k) = \mathbf{T}^{(g)}(k) \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \times_3 \mathbf{B}_3 \quad (2.27)$$

soient tous diagonaux. Idéalement, $\mathbf{T}'^{(g)}(k)$ seront égaux à $\mathbf{D}^{(g)}(k)$, ainsi d'après (2.27) et (2.23), on aura

$$\mathbf{D}^{(g)}(k) = \mathbf{D}^{(g)}(k) \times_1 (\mathbf{B}_1 \mathbf{A}_1) \times_2 (\mathbf{B}_2 \mathbf{A}_2) \times_3 (\mathbf{B}_3 \mathbf{A}_3), \quad (2.28)$$

avec

$$\mathbf{B}_1 \mathbf{A}_1 = \mathbf{I}_{N_s}, \quad \mathbf{B}_2 \mathbf{A}_2 = \mathbf{I}_{N_s} \quad \text{et} \quad \mathbf{B}_3 \mathbf{A}_3 = \mathbf{I}_{N_s}. \quad (2.29)$$

Ainsi les matrices \mathbf{B}_1 , \mathbf{B}_2 et \mathbf{B}_3 sont les pseudo-inverses respectives de \mathbf{A}_1 , \mathbf{A}_2 et \mathbf{A}_3 ,

$$\mathbf{B}_1 = \mathbf{A}_1^\dagger, \quad \mathbf{B}_2 = \mathbf{A}_2^\dagger \quad \text{et} \quad \mathbf{B}_3 = \mathbf{A}_3^\dagger. \quad (2.30)$$

Remarque : Ici encore, comme les tenseurs $\mathbf{D}^{(g)}(k)$ sont diagonaux pour tout k , il existe un lien direct entre la décomposition (2.23) et la décomposition CP d'un tenseur complexe d'ordre quatre. En effet, on a

$$T_{i_1 i_2 i_3}^{(g)}(k) = \sum_{j=1}^{N_s} D_{jjj}(k)^{(g)} A_{1,i_1 j} A_{2,i_2 j} A_{3,i_3 j} \quad (2.31)$$

qui peut directement être écrit comme

$$T_{i_1 i_2 i_3 k}^{(g)} = \sum_{j=1}^{N_s} A_{1, i_1 j} A_{2, i_2 j} A_{3, i_3 j} A_{4, k j} \quad (2.32)$$

où $A_{4, k j} = D_{j j j}^{(g)}(k)$ et $T_{i_1 i_2 i_3 k}^{(g)} = T_{i_1 i_2 i_3}^{(g)}(k)$ pour toutes les valeurs des indices. L'expression (2.32) est la décomposition CP du tenseur d'ordre quatre $\mathbf{T}^{(g)} = (T_{i_1 i_2 i_3 k}^{(g)})$.

2.2.2 Ensembles importants

En séparation de sources, la diagonalisation conjointe d'ensembles de tenseurs complexes d'ordre trois particuliers s'avère très importante. C'est notamment le cas des ensembles de tenseurs suivant le modèle INDSCAL ou encore symétrique.

Ensemble de tenseurs admettant une décomposition INDSCAL : La décomposition INDSCAL diffère du modèle général (2.26) par le fait que deux des matrices facteurs sont identiques. Ici, on considèrera

$$\mathbf{A}_2 = \mathbf{A}_1.$$

On considère un ensemble $\mathcal{T}^{(I)}$ de K tenseurs d'ordre trois complexes de dimensions $N_{o_1} \times N_{o_1} \times N_{o_3}$, avec $(N_{o_1}, N_{o_3}) \in \{\mathbb{N}^* \setminus \{1\}\}^2$, défini de la manière suivante

$$\mathcal{T}^{(I)} = \{\mathbf{T}^{(I)}(k) \in \mathbb{C}^{N_{o_1} \times N_{o_1} \times N_{o_3}}, k \in \mathcal{K}\}. \quad (2.33)$$

On suppose, cette fois, que les tenseurs $\mathbf{T}^{(I)}(k) \in \mathcal{T}^{(I)}$ respectent la décomposition simplifiée suivante

$$\mathbf{T}^{(I)}(k) = \mathbf{D}^{(I)}(k) \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_1 \times_3 \mathbf{A}_3 \quad (2.34)$$

avec $\mathbf{A}_1 \in \mathcal{A}_{N_{o_1} \times N_s}$ et $\mathbf{A}_3 \in \mathcal{A}_{N_{o_3} \times N_s}$ et où tous les tenseurs $\mathbf{D}^{(I)}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ sont diagonaux et non nuls.

Comme dans le cas général (paragraphe 2.2.1), la diagonalisation conjointe de l'ensemble $\mathcal{T}^{(I)}$ consiste en l'estimation de matrices $\mathbf{B}_1 \in \mathcal{B}_{N_s \times N_{o_1}}$, et $\mathbf{B}_3 \in \mathcal{B}^{N_s \times N_{o_3}}$ telles que les tenseurs transformés

$$\mathbf{T}^{(I)}(k) = \mathbf{T}^{(I)}(k) \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_1 \times_3 \mathbf{B}_3 \quad (2.35)$$

soient tous diagonaux. Ainsi, idéalement, on aura $\mathbf{B}_1 = \mathbf{A}_1^\dagger$ et $\mathbf{B}_3 = \mathbf{A}_3^\dagger$.

Remarque : Un cas particulier du modèle INDSCAL est à mettre en lumière. Il s'agit du cas où $\mathbf{A}_3 = \mathbf{A}_1^*$. Dans cette thèse, ce modèle est appelé INDSCAL hermitien, noté INDSCAL_h.

Un ensemble de K tenseurs $\mathbf{T}^{(h)}(k)$ admettant ce modèle est défini de la manière suivante

$$\mathbf{T}^{(h)}(k) = \mathbf{D}^{(h)}(k) \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_1 \times_3 \mathbf{A}_1^* \quad (2.36)$$

avec $\mathbf{A}_1 \in \mathcal{A}_{N_{o_1} \times N_s}$ et où tous les tenseurs $\mathbf{D}^{(h)}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ sont diagonaux et non nuls. Ce type d'ensembles de tenseurs peut être obtenu aisément par calcul de cumulants d'ordre quatre d'un ensemble de signaux complexes, comme nous le verrons par la suite.

Ensemble de tenseurs symétriques : Le modèle symétrique diffère du modèle général (2.26) par le fait que toutes les matrices facteurs sont identiques. Ici, on choisira

$$\mathbf{A}_3 = \mathbf{A}_2 = \mathbf{A}_1 = \mathbf{A}$$

avec $\mathbf{A} \in \mathcal{A}_{N_o \times N_s}$ et où $N_o \in \mathbb{N}^* \setminus \{1\}$.

On considère un ensemble $\mathcal{T}^{(s)}$ de K tenseurs d'ordre trois complexes de dimensions $N_o \times N_o \times N_o$ défini de la manière suivante

$$\mathcal{T}^{(s)} = \{\mathbf{T}^{(s)}(k) \in \mathbb{C}^{N_o \times N_o \times N_o}, k \in \mathcal{K}\}. \quad (2.37)$$

On suppose, cette fois, que les tenseurs $\mathbf{T}_k^{(s)} \in \mathcal{T}^{(s)}$ admettent la décomposition simplifiée suivante

$$\mathbf{T}^{(s)}(k) = \mathbf{D}^{(s)}(k) \times_1 \mathbf{A} \times_2 \mathbf{A} \times_3 \mathbf{A} \quad (2.38)$$

avec $\mathbf{A} \in \mathcal{A}_{N_o \times N_s}$ et où les tenseurs $\mathbf{D}^{(s)}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ sont diagonaux et non nuls. Comme les trois matrices facteurs sont égales, nous simplifierons l'expression (2.38) de la manière suivante

$$\mathbf{T}^{(s)}(k) = \mathbf{D}^{(s)}(k) \times_{(3)} \mathbf{A}. \quad (2.39)$$

Comme dans le cas général (paragraphe 2.2.1), la diagonalisation conjointe de l'ensemble $\mathcal{T}^{(s)}$ consiste en l'estimation d'une matrice $\mathbf{B} \in \mathcal{B}_{N_s \times N_o}$ telle que les tenseurs transformés

$$\mathbf{T}'^{(s)}(k) = \mathbf{T}^{(s)}(k) \times_{(3)} \mathbf{B} \quad (2.40)$$

soient tous diagonaux. Ainsi, idéalement, on aura $\mathbf{B} = \mathbf{A}^\dagger$.

Remarque : Bien évidemment, comme pour les matrices, on peut aussi diagonaliser conjointement des tenseurs réels d'ordre trois. Pour cela, il suffit de remplacer tous les tenseurs et les matrices complexes du paragraphe 2.2 par des tenseurs et des matrices réels de mêmes dimensions et de même nature. Ainsi pour la diagonalisation conjointe d'un ensemble

de tenseurs réels non-symétriques, on se référera au paragraphe 2.2.1 et pour des tenseurs symétriques ou admettant une décomposition INDSCAL, au paragraphe 2.2.2.

En séparation de sources, dans le domaine des télécommunications numériques, les ensembles de matrices ou de tenseurs sont généralement construits à l'aide de statistiques d'ordre supérieur. Ainsi, de tels ensembles présentent des propriétés de symétrie. Comme cette thèse a été réalisée dans le cadre applicatif de signaux de télécommunications numériques, nous nous sommes principalement concentrés, dans le cas matriciel, sur les ensembles de matrices complexes symétriques et hermitiennes (paragraphe 2.1.2) et dans le cas tensoriel, sur les ensembles de tenseurs complexes symétriques et INDSCAL_h (paragraphe 2.2.2). Toutefois, par soucis de comparaison, nous avons aussi confronté notre approche de diagonalisation conjointe non-unitaire de tenseurs non-symétriques d'ordre trois (paragraphe 2.2.1) avec la décomposition CP d'un tenseur d'ordre quatre.

2.3 Lien avec l'ICA et la séparation de sources

Dans ce paragraphe, nous rappelons, dans un premier temps, ce qu'est le problème de séparation de sources dans le cadre de mélanges linéaires et instantanés, en lien avec l'ICA. Puis, dans un second temps, nous montrons les relations avec la diagonalisation conjointe.

2.3.1 La séparation de sources

En séparation de sources, on dispose d'un nombre fini d'observations résultant du mélange instantané de signaux d'intérêt supposés indépendants (ICA) et appelés sources. Un tel mélange se modélise de la manière suivante [59]

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}(t) = \mathbf{x}(t) + \mathbf{b}(t) \quad (2.41)$$

où $\mathbf{s}(t)$ est le vecteur source de dimension $N_s \times 1$, \mathbf{A} est la matrice de mélange $N_o \times N_s$, $\mathbf{b}(t)$ est le vecteur de bruit additif de dimension $N_o \times 1$ (il peut représenter le bruit de mesure, l'erreur d'estimation ou l'erreur de modèle), $\mathbf{y}(t)$ est le vecteur d'observations $N_o \times 1$, tandis que le vecteur $\mathbf{x}(t)$ de dimension $N_o \times 1$ représente les observations non bruitées. La figure 2.1 schématise le modèle (2.41).

On suppose que ce modèle respecte les hypothèses classiques suivantes :

- les sources $s_i(t)$ sont des signaux aléatoires non gaussiens, statistiquement indépendants, de moyenne nulle et de puissance unitaire.
- la matrice de mélange \mathbf{A} est de rang plein en colonne

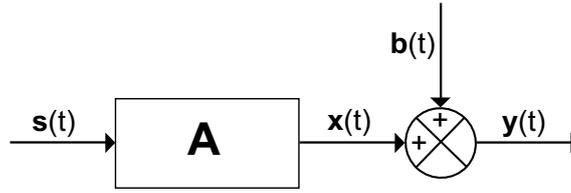


FIGURE 2.1 – Schéma classique d'un mélange linéaire instantané

- le bruit $\mathbf{b}(t)$ est gaussien, de moyenne nulle et indépendant des signaux sources.

Pour plus de précisions concernant le mélange, si $N_o < N_s$, il y a plus de sources que d'observations, on dit que le problème est sous-déterminé, la matrice de mélange \mathbf{A} sera alors supposée de rang plein en ligne, avec toutes ses colonnes distinctes deux à deux. Si, au contraire, $N_s \leq N_o$, le problème sera dit surdéterminé et \mathbf{A} sera supposée de rang plein en colonne. Par la suite, nous ne nous intéresserons qu'au cas surdéterminé.

Les sources étant non-observables et la matrice de mélange inconnue, le but de la séparation de sources est donc d'estimer une matrice \mathbf{B} , appelée matrice de séparation, telle que le vecteur

$$\mathbf{s}'(t) = \mathbf{B}\mathbf{x}(t) \quad (2.42)$$

corresponde au vecteur source recherché. Pour cela, on contraint l'indépendance des composantes de $\mathbf{s}'(t)$ afin d'estimer une matrice de séparation \mathbf{B} satisfaisante. On parle alors d'analyse en composantes indépendantes [29]. On cherche à retrouver l'indépendance des sources perdue lors du mélange. Idéalement, on voudrait donc avoir $\mathbf{s}'(t) = \mathbf{s}(t)$, autrement dit $\mathbf{B} = \mathbf{A}^\ddagger$. Cependant, on restaure la propriété d'indépendance dès lors que l'on a

$$\mathbf{B}\mathbf{A} = \mathbf{D}\mathbf{P} \quad (2.43)$$

où \mathbf{D} est une matrice diagonale inversible et \mathbf{P} est une matrice de permutation. En séparation de sources, ces matrices \mathbf{D} et \mathbf{P} sont considérées comme les indéterminations classiques lorsqu'on estime une matrice de séparation. La figure 2.2 schématise le cas idéal, *c.à.d.* non-bruité.

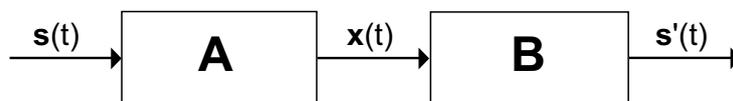


FIGURE 2.2 – Schéma de la séparation de source dans le cas non-bruité

Bien sûr, il est peu probable d'avoir accès à des observations non-bruitées. Le but est donc, en général, d'estimer une matrice de séparation \mathbf{B} à partir des observations bruitées, telle que le vecteur $\mathbf{s}'(t)$ corresponde le plus possible aux signaux sources. La figure 2.3 schématise la séparation de sources dans le cadre général.

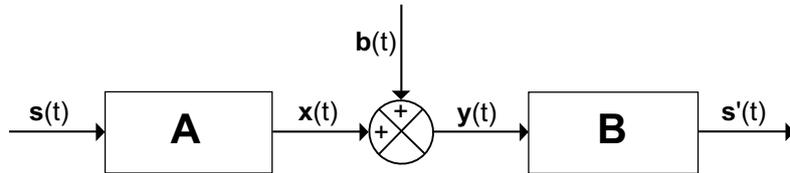


FIGURE 2.3 – Schéma de la séparation de sources dans le cadre général

Deux types de méthodes sont envisagées pour traiter le problème de séparation de sources. La première consiste à se placer avec le cadre de mélanges unitaires [29], *c.à.d.* $\mathbf{A}^H \mathbf{A} = \mathbf{I}$ où \mathbf{I} est la matrice identité. Auquel cas, on peut traiter le cas des mélanges non-unitaires en passant par une étape de prétraitement appelée blanchiment. Le blanchiment est en fait une décorrélation des observations associée à une normalisation en puissance. Après une telle étape de prétraitement, la matrice de séparation à estimer est unitaire. Ainsi, comme le montre le schéma de la figure 2.4, la matrice \mathbf{B} de la figure 2.3 se décomposera en une matrice de blanchiment \mathbf{W} et une matrice de séparation unitaire \mathbf{Q} . La seconde méthode consiste à traiter directement le problème de séparation de sources dans le cadre de mélanges non-unitaires [63], ainsi on estime une matrice de séparation \mathbf{B} qui sera elle aussi non-unitaire (voir figure 2.3).

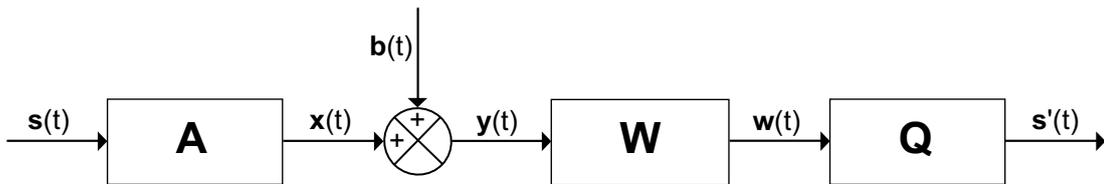


FIGURE 2.4 – Schéma de séparation de sources avec blanchiment des observations

2.3.2 La diagonalisation conjointe comme méthode de séparation de sources

Afin de traiter le problème de séparation de sources, de nombreuses méthodes ont été développées. Parmi elles, on trouve les méthodes de diagonalisation conjointe qui tirent, quasiment toutes, leur origine des travaux de Cardoso et Souloumiac [14]. Ce sont ces méthodes de diagonalisation conjointe qui nous intéressent particulièrement dans le cadre de cette thèse.

Afin d'utiliser la diagonalisation conjointe, on peut construire des ensembles de matrices ou de tenseurs que l'on qualifiera de cibles, à partir du modèle de mélange linéaire et instantané (2.41). Pour cela, on estime, en général, des statistiques issues des observations que l'on range, ensuite, dans des matrices ou des tenseurs. Enfin, des décompositions algébriques sont mises en place afin de diagonaliser conjointement les matrices ou tenseurs de ces ensembles cibles. Les statistiques qui peuvent être considérées sont nombreuses. Ici, nous prendrons seulement l'exemple des cumulants d'ordre quatre (statistiques d'ordre supérieur) qui peuvent aussi bien être utilisés pour la construction d'ensembles de matrices cibles ou de tenseurs cibles.

Pour un mélange non-bruité, considérons les cumulants d'ordre quatre suivants

$$C_{\mathbf{x},1}^{a,b,c,d} = \text{Cum}(x_a, x_b, x_c, x_d) \quad (2.44)$$

$$C_{\mathbf{x},2}^{a,b,c,d} = \text{Cum}(x_a, x_b^*, x_c, x_d) \quad (2.45)$$

$$C_{\mathbf{x},3}^{a,b,c,d} = \text{Cum}(x_a, x_b, x_c^*, x_d^*) \quad (2.46)$$

avec $(a, b, c, d) \in \{1, \dots, N_o\}$.

Remarque : Certains de ces cumulants peuvent présenter des propriétés particulières en fonction de la circularité des signaux considérés [26]. Un signal \mathbf{x} est dit circulaire, au sens strict, si sa loi de probabilité est invariante par rotation. Pour un tel signal, on a la propriété suivante

$$\mathbb{E}(x_{i_1} \dots x_{i_p} x_{j_1}^* \dots x_{j_q}^*) = 0 \quad (2.47)$$

pour tout $(i_k, j_k) \in \{1, \dots, N_o\}^{p+q}$ et pour $p \neq q$. Le signal \mathbf{x} est dit circulaire à l'ordre n si (2.47) est vérifié pour tout $p+q = n$ avec $p \neq q$. Ainsi, notons que, pour tout type de signaux circulaires, le cumulante (2.45) sera nul. Le cumulante (2.44) sera nul, quant à lui, pour des signaux circulaires à l'ordre au moins quatre. Finalement, (2.45) ne concerne que les signaux non circulaires, (2.44) sera utilisé pour des signaux non-circulaires à l'ordre quatre et (2.46) sera utilisable pour tout type de signaux.

A partir de ces statistiques d'ordre quatre, on va pouvoir construire des matrices et des tenseurs d'ordre trois directement en lien avec le modèle (2.41).

Pour les matrices : On peut considérer

$$(\mathbf{M}_{\mathbf{x},1}^{c,d})_{a,b} = C_{\mathbf{x},1}^{a,b,c,d} \quad (2.48)$$

$$(\mathbf{M}_{\mathbf{x},2}^{c,d})_{a,b} = C_{\mathbf{x},2}^{a,b,c,d} \quad (2.49)$$

$$(\mathbf{M}_{\mathbf{x},3}^{c,d})_{a,b} = C_{\mathbf{x},3}^{a,b,c,d}. \quad (2.50)$$

En remplaçant $\mathbf{x}(t)$ par son expression dans (2.41), on obtient

$$\mathbf{M}_{\mathbf{x},1}^{c,d} = \mathbf{A}\mathbf{M}_{\mathbf{s},\mathbf{x},1}^{c,d}\mathbf{A}^T \quad (2.51)$$

$$\mathbf{M}_{\mathbf{x},2}^{c,d} = \mathbf{A}\mathbf{M}_{\mathbf{s},\mathbf{x},2}^{c,d}\mathbf{A}^H \quad (2.52)$$

$$\mathbf{M}_{\mathbf{x},3}^{c,d} = \mathbf{A}\mathbf{M}_{\mathbf{s},\mathbf{x},3}^{c,d}\mathbf{A}^T \quad (2.53)$$

où les matrices

$$\mathbf{M}_{\mathbf{s},\mathbf{x},1}^{c,d} = \text{Cum}(s_a, s_b, x_c, x_d) \quad (2.54)$$

$$\mathbf{M}_{\mathbf{s},\mathbf{x},2}^{c,d} = \text{Cum}(s_a, s_b^*, x_c, x_d) \quad (2.55)$$

$$\mathbf{M}_{\mathbf{s},\mathbf{x},3}^{c,d} = \text{Cum}(s_a, s_b, x_c^*, x_d^*) \quad (2.56)$$

sont diagonales car les sources sont supposées statistiquement indépendantes.

Ainsi, à partir de cumulants d'ordre quatre, il est possible de construire des ensembles de matrices symétriques ou hermitiennes conjointement diagonalisables. Les décompositions considérées en (2.54) et (2.56) se modélisent génériquement par des matrices symétriques (2.12)

$$\mathbf{M}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^T \quad \text{avec } k \in \mathcal{K} \quad (2.57)$$

où \mathbf{D}_k sont des matrices diagonales. Quant à la décomposition considérée en (2.55), elle se modélise de façon générique par des matrices hermitiennes (2.16)

$$\mathbf{M}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^H \quad \text{avec } k \in \mathcal{K}. \quad (2.58)$$

Pour les tenseurs d'ordre trois : On peut considérer

$$(\mathbf{T}_{\mathbf{x},1}^d)_{a,b,c} = C_{\mathbf{x},1}^{a,b,c,d} \quad (2.59)$$

$$(\mathbf{T}_{\mathbf{x},2}^d)_{a,b,c} = C_{\mathbf{x},2}^{a,b,c,d} \quad (2.60)$$

$$(\mathbf{T}_{\mathbf{x},3}^d)_{a,b,c} = C_{\mathbf{x},3}^{a,b,c,d}. \quad (2.61)$$

En remplaçant $\mathbf{x}(t)$ par son expression dans (2.41), on obtient

$$\mathbf{T}_{\mathbf{x},1}^d = \mathbf{T}_{\mathbf{s},\mathbf{x},1}^d \times_{(3)} \mathbf{A} \quad (2.62)$$

$$\mathbf{T}_{\mathbf{x},2}^d = \mathbf{T}_{\mathbf{s},\mathbf{x},2}^d \times_1 \mathbf{A} \times_2 \mathbf{A}^* \times_3 \mathbf{A} \quad (2.63)$$

$$\mathbf{T}_{\mathbf{x},3}^d = \mathbf{T}_{\mathbf{s},\mathbf{x},3}^d \times_1 \mathbf{A} \times_2 \mathbf{A} \times_3 \mathbf{A}^* \quad (2.64)$$

où les tenseurs

$$\mathbf{T}_{\mathbf{s},\mathbf{x},1}^d = \text{Cum}(s_a, s_b, s_c, x_d) \quad (2.65)$$

$$\mathbf{T}_{\mathbf{s},\mathbf{x},2}^d = \text{Cum}(s_a, s_b^*, s_c, x_d) \quad (2.66)$$

$$\mathbf{T}_{\mathbf{s},\mathbf{x},3}^d = \text{Cum}(s_a, s_b, s_c^*, x_d^*) \quad (2.67)$$

sont diagonaux car les sources sont supposées statistiquement indépendantes.

Ainsi, à partir de cumulants d'ordre quatre, il est aussi possible de construire des ensembles de tenseurs d'ordre trois symétriques ou INDSCAL_h , conjointement diagonalisables. La décomposition considérée en (2.65) se modélise génériquement par des tenseurs symétriques (2.39)

$$\mathbf{T}(k) = \mathbf{D}(k) \times_{(3)} \mathbf{A} \quad \text{avec } k \in \mathcal{K} \quad (2.68)$$

où $\mathbf{D}(k)$ sont des tenseurs diagonaux. Quant aux décompositions considérées en (2.66) et (2.67), elles se modélisent de façon générique, respectivement, par les tenseurs $\mathbf{T}_2(k)$ et $\mathbf{T}_3(k)$ admettant, pour tout $k \in \mathcal{K}$, les décompositions INDSCAL_h suivantes

$$\mathbf{T}_2(k) = \mathbf{D}_2(k) \times_1 \mathbf{A} \times_2 \mathbf{A}^* \times_3 \mathbf{A} \quad (2.69)$$

$$\mathbf{T}_3(k) = \mathbf{D}_3(k) \times_1 \mathbf{A} \times_2 \mathbf{A} \times_3 \mathbf{A}^* \quad (2.70)$$

où $\mathbf{D}_2(k)$ et $\mathbf{D}_3(k)$ sont des tenseurs diagonaux.

Remarque : Il est important de noter que les matrices cibles et tenseurs cibles sont estimés à partir de statistiques calculées sur un nombre fini d'observations. Cela induit une erreur d'estimation et de ce fait, la diagonalisation conjointe de ces ensembles ne pourra pas être exacte mais seulement approchée.

2.4 Etat de l'art des méthodes de diagonalisation conjointe

Actuellement, de nombreux algorithmes de diagonalisation conjointe d'ensembles de tenseurs sont basés sur une étape de dépliement des tenseurs de façon à pouvoir ensuite utiliser des algorithmes de diagonalisation conjointe de matrices. Si on se contente d'un dépliement basique, deux problèmes peuvent se poser. En effet, on considère un tenseur $\mathbf{T} \in \mathbb{C}^{N \times N \times N}$ et les N matrices $\mathbf{M}_z \in \mathbb{C}^{N \times N}$, $z \in \{1, \dots, N\}$ correspondant au dépliement basique de \mathbf{T} selon le troisième mode, *c.à.d.* tels que, pour tout $(x, y, z) \in \{1, \dots, N\}^3$,

$$M_{z,xy} = T_{xyz}. \quad (2.71)$$

La diagonalisation de \mathbf{T} consiste à minimiser ses $N(N^2 - 1)$ termes hors-diagonaux tandis que la diagonalisation conjointe des \mathbf{M}_z consiste à minimiser, pour chacune des matrices, $N(N - 1)$ termes hors-diagonaux, soit un total de $N^2(N - 1)$ termes. Ces deux types de diagonalisations ne sont donc clairement pas équivalents. La figure 2.5 met en avant les différences entre ces deux approches. Notons, aussi, que si on traite un problème de diagonalisation conjointe

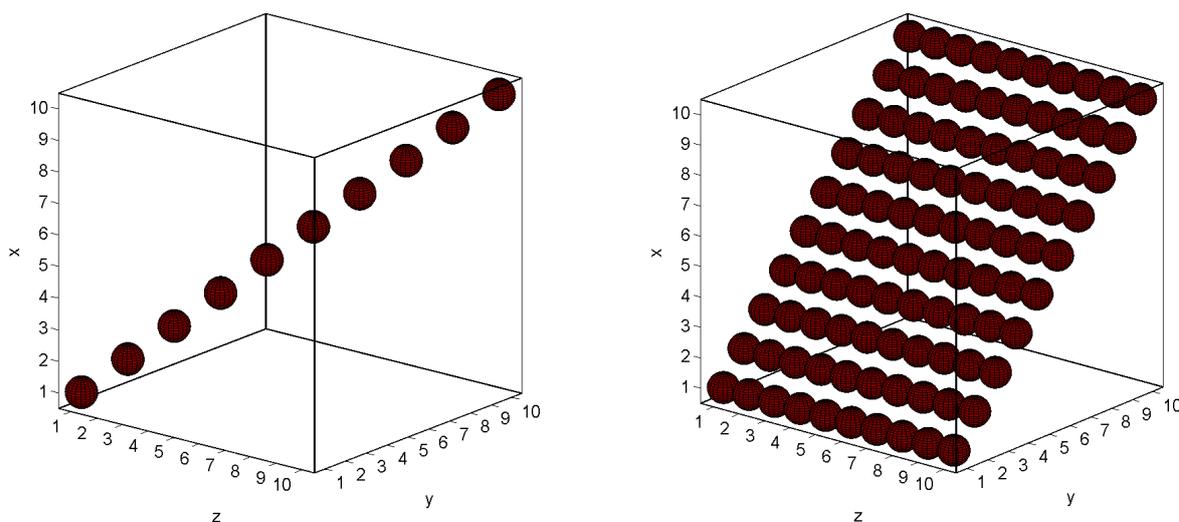


FIGURE 2.5 – Position des composantes non minimisées lors de la diagonalisation du tenseur \mathbf{T} pour $N = 10$, à gauche lors de la diagonalisation directe du tenseur, à droite en passant par la diagonalisation conjointe des matrices constituant les tranches de \mathbf{T} selon le troisième mode

de tenseurs d'ordre trois à l'aide d'un algorithme de diagonalisation conjointe de matrices en passant par un dépliement basique, seules deux matrices diagonalisantes seront estimées. Les informations de la troisième matrice diagonalisante restent contenues dans l'ensemble de matrices diagonalisées. S'il est nécessaire d'estimer cette matrice, il s'agira alors de mettre en place un protocole de dépliement spécifique des tenseurs cibles afin de ne pas perdre d'information.

Dans ce paragraphe nous faisons un état de l'art des méthodes de diagonalisation conjointe de matrices. Dans tout ce paragraphe, afin de simplifier les explications, nous nous placerons dans le cas où toutes les matrices du problème sont carrées de dimensions $N_s \times N_s$ et où les matrices facteurs sont identiques $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}$ et donc les matrices diagonalisantes aussi $\mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B}$ (voir paragraphe 2.1.2).

2.4.1 Approches envisagées

Dans la littérature, différentes approches sont envisagées pour la diagonalisation conjointe d'un ensemble de matrices. Nous rappelons que le but d'une telle opération est d'estimer la matrice diagonalisante permettant de diagonaliser conjointement de façon exacte ou approchée l'ensemble des matrices cibles. Pour cela, deux grandes catégories d'approches se distinguent.

Approches globales : Le premier type d'approches consiste à estimer la matrice diagonalisante de manière globale, *c.à.d.* que l'on va estimer toutes les composantes de la matrice diagonalisante simultanément, comme c'est le cas par exemple pour la décomposition en éléments propres généralisée (GEVD) et la décomposition en valeurs singulières généralisée (GSVD). C'est, notamment, ce type d'approches qui a été envisagé pour l'algorithme EJD (pour *Exact Joint Diagonalization*) développé par Yeredor [133]. EJD fournit une solution sous-optimale et non-itérative en calculant deux matrices représentatives de l'ensemble de matrices cibles puis en les diagonalisant conjointement à l'aide d'une GEVD. L'algorithme DIEM (pour *Diagonalization using Equivalent Matrices*) développé par Chabriel et Barrère et présenté dans [22] est, lui aussi, construit sur une approche globale. DIEM s'appuie sur EJD. Toutefois, la méthode proposée est plus robuste. Tichavský et Yeredor ont aussi développé un algorithme selon cette approche, dénommé U-WEDGE (pour *Uniformly-Weighted Exhaustive Diagonalization with Gauss iterations*) et présenté dans [114]. Dans cet algorithme la matrice diagonalisante, judicieusement initialisée, est ensuite estimée itérativement à l'aide d'une méthode de Gauss. Parmi les approches globales, on peut aussi citer, FFdiag (pour *Fast Frobenius Diagonalization*) développé par Ziehe *et al.* et proposé dans [135]. Cet algorithme itératif repose sur des étapes de mise à jour quasi-Newton. L'algorithme NOODLES (pour *NON Orthogonal Decomposition of complex matrix Sets*), développé par Trainini et Moreau dans [118], a pour particularité de pouvoir diagonaliser conjointement des ensembles contenant des matrices complexes symétriques et des matrices hermitiennes. C'est un algorithme itératif de gradient conjugué qui permet d'estimer la matrice diagonalisante. NOODLES et FFdiag sont assez proches dans leurs structures.

Approches locales : Le second type d'approches, que l'on qualifiera de locales, consiste à décomposer le problème initial de grandes dimensions parfois compliqué à résoudre, en plusieurs problèmes bien plus abordables. On va donc mettre en place des algorithmes itératifs permettant d'estimer une partie des paramètres de la matrice diagonalisante, les autres paramètres étant fixés. Ainsi, par un système de balayage, toutes les composantes de cette matrice seront estimées.

Parmi les algorithmes utilisant ce type d'approches, on retrouve FAJD (pour *Fast Ap-*

proximate Joint Diagonalization) présenté dans [70] par Li et Zhang. Cet algorithme, à chaque itération, va mettre à jour une seule colonne de la matrice diagonalisante, les autres colonnes étant considérées fixées. Vollgraff et Obermayer [124] propose l'algorithme QDIAG (pour *Quadratic DIAGonalization*) qui estime chaque vecteur ligne de la matrice diagonalisante alternativement par un calcul de vecteurs propres. Dans [132], Yeredor présente l'algorithme AC-DC (pour *Aternating Columns - Diagonal Centers*) qui consiste en l'estimation de la matrice facteur et des matrices diagonales contenant les sources en deux étapes itératives. Durant la première étape (AC), on estime une colonne de la matrice facteur, en gardant les paramètres des autres colonnes ainsi que des matrices sources fixés. La seconde étape (DC) permet l'estimation des composantes des matrices sources, les matrices de mélanges étant, bien entendu, fixées. Notons toutefois que pour assurer la convergence de l'algorithme, une seule étape DC est suivie de plusieurs étapes AC.

Toujours parmi ces approches locales, les procédures de Jacobi (dans le cas de mélanges unitaires) et de type Jacobi (dans le cas de mélanges non-unitaires) sont à mettre en lumière. Ces procédures ont pour but de décomposer la matrice diagonalisante en un produit de matrices élémentaires possédant chacune au maximum quatre composantes utiles. Pour illustrer une telle décomposition, on considère la matrice diagonalisante $\mathbf{B} \in \mathcal{B}_{N_s \times N_s}$ (2.8). En utilisant une procédure de type Jacobi, cette matrice \mathbf{B} sera décomposée en un produit de $\frac{N_s(N_s-1)}{2}$ matrices élémentaires \mathbf{B}^{ij} telles que

$$\mathbf{B} = \prod_{i=1}^{N_s-1} \prod_{j=i+1}^{N_s} \mathbf{B}^{ij} \quad (2.72)$$

avec pour tout $(i, j) \in \mathcal{N}_s^2 = \{1, \dots, N\}$, $i < j$

$$\mathbf{B}^{ij} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & B_{ii}^{ij} & \cdots & B_{ij}^{ij} & \vdots \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & B_{ji}^{ij} & \cdots & B_{jj}^{ij} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}. \quad (2.73)$$

Ainsi avec un balayage itératif de toutes les paires (i, j) possibles avec $i < j$ et une mise à jour régulière des matrices cibles, on obtient une estimation de la matrice diagonalisante. Les algorithmes basés sur une procédure de type Jacobi ont l'avantage d'être très simples à implémenter, d'autant plus si l'estimation des paramètres se fait de manière analytique. Ils sont aussi potentiellement parallélisables.

Ces vingt dernières années, de nombreux algorithmes de Jacobi ou de type Jacobi ont

été développés dans le cadre de la diagonalisation conjointe. JADE, proposé par Cardoso et Souloumiac dans [15], est l'un des tous premiers algorithmes de diagonalisation conjointe. Il fonctionne dans le cadre de mélanges unitaires avec un ensemble de matrices hermitiennes. Les matrices élémentaires sont des matrices de Givens (matrices de rotation) et celles-ci sont estimées à l'aide d'un calcul de vecteurs propres. Dans le cadre de mélange non-unitaire, de nombreux algorithmes utilisent aussi une paramétrisation particulière des matrices élémentaires. C'est le cas d'Afsari dans [1] qui utilise les paramétrisations LU (des matrices triangulaires respectivement inférieures et supérieures) et QR (une matrice orthogonale et une matrice triangulaire) pour les différents algorithmes qu'il propose dans le cas réel. Ces derniers ont ensuite été adaptés au cas complexe hermitien par Wang *et al.* dans [127]. Souloumiac dans [109] puis Mesloub *et al.* dans [78] ont développé respectivement les algorithmes J-Di (pour *Joint-Diagonalization*) et CJDi (pour *Complex J-Di*) en décomposant chaque matrice élémentaire comme le produit d'une matrice de Givens et d'une matrice de rotation hyperbolique. Dans [126], Wang *et al.* proposent l'algorithme DNJD (pour *Distinctively Normalized Joint Diagonalization*). Dans ce dernier, les matrices élémentaires sont paramétrées de façon similaire à des matrices de rotation mais dépendant de deux angles distincts. D'autres algorithmes de type Jacobi n'utilisent pas de paramétrisation, c'est le cas, notamment, de JTJD (pour *Jacobi Transformations for Joint Diagonalization*) développé par Guo *et al.* dans [56]. Dans cet algorithme, pour chaque couple (i, j) avec $i < j$, des matrices 2×2 sont extraites aux intersections des lignes et colonnes i et j des matrices cibles. Par recherche de valeurs propres, on trouve deux matrices 2×2 "représentatives" des matrices cibles extraites. Enfin ces deux matrices "représentatives" sont diagonalisées conjointement par recherche de vecteurs propres afin d'estimer les paramètres de la matrice élémentaire correspondante.

2.4.2 Un problème d'optimisation

La diagonalisation conjointe approchée est traitée, la plupart du temps, comme un problème d'optimisation. Pour cela, chaque algorithme s'appuie sur un critère qu'il s'agira d'optimiser. Un tour d'horizon des critères utilisés en diagonalisation conjointe est fait dans [24]. Il en existe de quatre types : le critère direct, le critère inverse, le critère mixte et la log-vraisemblance. Dans ce paragraphe, nous détaillons ces différents critères.

Critère direct : Le critère direct est une mesure de l'erreur, au sens des moindres carrés, entre les matrices cibles et leur modèle supposé. Il est défini de la manière suivante

$$\mathcal{J}_{\text{direct}}(\mathbf{A}, \mathbf{D}_k) = \sum_{k=1}^K \|\mathbf{M}_k - \mathbf{A}\mathbf{D}_k\mathbf{A}^\dagger\|^2. \quad (2.74)$$

En minimisant ce critère, on va pouvoir estimer la matrice facteur \mathbf{A} et les K matrices diagonales \mathbf{D}_k avec $k \in \mathcal{K}$. Le critère a été proposé initialement par Wax et Sheinvald [129] dans le cas de mélanges unitaires. Par la suite, il a été utilisé pour des mélanges non-unitaires, notamment par Van der Veen qui propose de minimiser ce critère à l'aide d'une technique faisant correspondre des sous-espaces [122]. Ensuite, Yeredor a proposé un schéma d'optimisation efficace dans son algorithme AC-DC [133]. Plus récemment, Zheng et Feng ont proposé un algorithme basé sur une méthode de région de confiance hybride [134]. Notons aussi l'algorithme DIEM [22], dans lequel l'optimisation d'un critère direct sous-optimal est proposée afin d'obtenir une solution approchée et non-itérative. DIEM permet aussi de traiter un problème non-symétrique, *c.à.d.* avec deux matrices facteurs différentes (2.2). Le critère direct présente l'avantage de pouvoir traiter les cas où \mathbf{A} n'est pas carré. Par exemple, De Lathauwer et Castaing s'attaquent à un problème sous-déterminé, *c.à.d.* où \mathbf{A} possède plus de colonnes que de lignes [36].

Critère inverse : Le critère inverse est défini de la manière suivante

$$\mathcal{J}_{\text{inverse}}(\mathbf{B}) = \sum_{k=1}^K \|\text{ZDiag}\{\mathbf{B}\mathbf{M}_k\mathbf{B}^\dagger\}\|^2 \quad (2.75)$$

Ce critère est une mesure de la diagonalité. La minimisation de $\mathcal{J}_{\text{inverse}}$ va donc permettre d'estimer une matrice diagonalisante. Toutefois, la minimisation de (2.75) admet une solution triviale et non désirée $\mathbf{B} = \mathbf{0}$ où $\mathbf{0}$ est la matrice nulle. Or la matrice diagonalisante \mathbf{B} à estimer doit appartenir à l'ensemble $\mathcal{B}_{N_s \times N_s}$ et donc être inversible. Ainsi, pour éviter la solution $\mathbf{B} = \mathbf{0}$ ou toute autre solution singulière et assurer l'inversibilité de \mathbf{B} plusieurs solutions ont été envisagées dans l'état de l'art.

- \mathbf{B} est imposé unitaire. On utilise cette contrainte dans le cas de mélanges unitaires ou encore dans le cas de mélanges non-unitaires mais avec une étape de blanchiment préalable. Cardoso et Soudoumiac ont utilisé cette contrainte pour JADE qui a pour but de diagonaliser des matrices hermitiennes de cumulants d'ordre quatre [15]. Moreau a ensuite généralisé JADE pour toutes les statistiques d'ordre supérieur ou égal à deux [80]. Dans l'algorithme SOBI qui s'applique à des matrices hermitiennes de statistiques d'ordre deux, Belouchrani *et al.* contraignent aussi la matrice diagonalisante à être unitaire.
- Les lignes de \mathbf{B} sont de norme unitaire. Cette contrainte, plus faible que la précédente, a été adoptée, par exemple, dans [42] et [47].
- Les composantes diagonales de la matrice $\mathbf{M}'_0 = \mathbf{B}\mathbf{M}_0\mathbf{B}^\dagger$ doivent être égales à un, avec \mathbf{M}_0 qui peut être retenue ou non dans l'ensemble des matrices cibles. En séparation

de sources, si \mathbf{M}_0 est la matrice de covariance non retardée des observations, alors cette contrainte impose que les sources séparées soient de puissance égale et unitaire. C'est ce type de contrainte qui est considéré dans l'algorithme QDIAG [124].

- La matrice \mathbf{B} est à diagonale dominante, ce qui signifie que pour chaque ligne de \mathbf{B} le module de la composante diagonale est supérieur à la somme des modules des composantes hors-diagonales. C'est le cas pour l'algorithme FFdiag [135], où à chaque itération, la matrice estimée à l'aide du critère (2.75) est contrainte à être à diagonale dominante. Par la suite, Xu *et al.* ont généralisé FFdiag au cas complexe [131]. Trainini et Moreau proposent l'utilisation de cette contrainte pour l'algorithme NOODLES [118] dont la résolution repose sur une méthode de gradient conjugué. La décomposition suivante est considérée $\mathbf{B} = \mathbf{I} + \mathbf{Z}$ où \mathbf{I} est la matrice identité et \mathbf{Z} est une matrice zéro-diagonale. Pour assurer que \mathbf{B} soit à diagonale dominante, la somme des modules de chaque ligne de \mathbf{Z} doit donc être inférieure à un. Si ce n'est pas le cas, il est possible de choisir un pas μ pour le gradient conjugué de façon à ce que $\mu\mathbf{Z}$ respecte la contrainte précédente.
- Un terme de pénalité est ajouté à (2.75). Li et Wang utilisent comme pénalisation $\log |\det(\mathbf{B})|$ puis minimisent alternativement le critère par rapport à chacun des vecteurs colonnes de \mathbf{B} [70]. Si la matrice diagonalisante \mathbf{B} n'est pas carrée, Li *et al.* optent pour la pénalisation $\log |\det(\mathbf{B}\mathbf{B}^H)|$ et proposent d'optimiser le critère par un algorithme du gradient [71]. Wang *et al.*, quant à eux, considèrent deux termes de pénalité, à savoir $\|\mathbf{B}\mathbf{B}^H - \mathbf{I}\|^2$ et $\|\text{Diag}\{\mathbf{B} - \mathbf{I}\}\|^2$ et implémentent, eux aussi, un algorithme du gradient [128].
- Le déterminant de la matrice \mathbf{B} est égal à un. Cette contrainte est, en général, obtenue en décomposant la matrice \mathbf{B} en produit de deux matrices de déterminant un. Par exemple, on trouve dans la littérature, la décomposition LU qui est la combinaison de deux matrices triangulaires, l'une inférieure et l'autre supérieure, et dont les composantes diagonales sont toutes égales à un. Cette décomposition (ou paramétrisation ou encore factorisation) a été utilisée pour l'algorithme LUJ1D [1], puis pour sa généralisation au cas complexe hermitien LUCJD (pour *LU-based method for Complex non-orthogonal Joint Diagonalization*) dans [127]. La combinaison QR (aussi dénotée LQ) est aussi fréquemment utilisée pour la factorisation de \mathbf{B} , la matrice \mathbf{Q} étant une matrice orthogonale (généralement, il est choisi une matrice de rotation de Givens) et \mathbf{R} (ou \mathbf{L}) étant une matrice triangulaire avec ses composantes diagonales toutes égales à un. Cette décomposition a été envisagée pour les algorithmes QRJ1D [1], LQCJD (pour *LQ-based method for Complex non-orthogonal Joint Diagonalization*) qui est la généralisation de QRJ1D au cas complexe hermitien [127] ou encore pour

l'algorithme SL, développé par Sørensen *et al.* [108]. \mathbf{B} peut aussi être paramétrée par une matrice de Givens et une matrice de rotation hyperbolique, comme c'est le cas pour les algorithmes J-Di [109] ou bien sa version complexe hermitienne CJDi [78]. On peut enfin citer, par exemple, la paramétrisation trigonométrique mise en œuvre par Wang *et al.* [126]. Lorsque ces décompositions sont couplées à une procédure de type Jacobi, alors chaque matrice élémentaire ne dépend plus, au maximum, que de deux paramètres. Ainsi l'estimation de ces matrices élémentaires peut souvent être traitée algébriquement.

- Un critère inverse à invariance d'échelle est considéré. Ce critère a été proposé par Afsari dans [1] et [2] dans le cas de mélanges réels. Il est défini de la manière suivante

$$\mathcal{J}_{\text{inverse}_2}(\mathbf{B}) = \sum_{k=1}^K \|\mathbf{M}_k - \mathbf{B}^{-1} \text{Diag}\{\mathbf{B}\mathbf{M}_k\mathbf{B}^T\}\mathbf{B}^{-T}\|^2. \quad (2.76)$$

Tout changement d'échelle au niveau des lignes de \mathbf{B} n'aura aucune influence sur (2.76). Dans [1], Afsari propose deux algorithmes basés sur l'optimisation de ce critère : LUJ2D et QRJ2D. La matrice diagonalisante \mathbf{B} suit une décomposition LU dans le premier algorithme et une décomposition QR dans le second.

Critère mixte : Il est possible de combiner les critères directs et indirects de manière efficace. Tichavský et Yeredor définissent le critère mixte, pour un ensemble de matrices cibles réelles symétriques, dans [113] comme

$$\mathcal{J}_{\text{mixte}}(\mathbf{A}^{[i]}) = \sum_{k=1}^K \|\mathbf{B}^{[i]}\mathbf{M}_k\mathbf{B}^{[i]T} - \mathbf{A}^{[i]}\text{Diag}\{\mathbf{B}^{[i]}\mathbf{M}_k\mathbf{B}^{[i]T}\}\mathbf{A}^{[i]T}\|^2 \quad (2.77)$$

où $\mathbf{B}^{[i]}\mathbf{M}_k\mathbf{B}^{[i]T}$ est un ensemble de matrices partiellement diagonalisées avec $k \in \mathcal{K}$ et i est l'indice d'itération. Dans ce même article, Tichavský et Yeredor proposent l'algorithme U-WEDGE basé sur l'optimisation du critère (2.77). La méthode employée décrite dans [24] est la suivante. Il faut dans un premier temps initialiser la matrice de diagonalisation, que l'on pourra prendre comme l'identité par exemple, ainsi $\mathbf{B}^{[0]} = \mathbf{I}$. Chaque itération va ensuite consister à estimer une matrice $\mathbf{A}^{[i]}$ à l'aide de la procédure directe. Cette estimation est réalisée à l'aide d'une méthode de Gauss-Newton en prenant pour point initial $\mathbf{A}^{[i]} = \mathbf{I}$. Il s'avère qu'avec cette initialisation, on peut se contenter d'une seule itération de la méthode de Gauss-Newton, car la matrice Hessienne est telle qu'il est possible de l'inverser en résolvant des ensembles distincts d'équations linéaires 2×2 . Lorsqu'on a trouvé une bonne estimation de $\mathbf{A}^{[i]}$, la matrice diagonalisante est mise à jour pour l'itération suivante. Ainsi $\mathbf{B}^{[i+1]} =$

$\mathbf{A}^{[i]-1}\mathbf{B}^{[i]}$. Notons qu'une version pondérée du critère (2.77) est aussi proposée dans ce même article [113] et l'algorithme résultant est dénoté WEDGE.

Log-vraisemblance : Ce dernier critère a été développé par Pham, afin de mesurer la diagonalité d'un ensemble de matrices définies positives [93]. Plus précisément, Pham avait pour but de diagonaliser conjointement un ensemble de matrices de covariance estimé à partir de blocs d'observations distincts. Au niveau du modèle, les sources sont supposées indépendantes et de variance constante dans chaque bloc, mais variant d'un bloc à l'autre. La log-vraisemblance est définie dans le cas réel de la manière suivante

$$\mathcal{J}_{\log V}(B) = \sum_{k=1}^K \log \frac{\det(\text{Diag}\{\mathbf{B}\mathbf{M}_k\mathbf{B}^T\})}{\det(\mathbf{B}\mathbf{M}_k\mathbf{B}^T)}. \quad (2.78)$$

Pour optimiser ce critère, Pham propose de maximiser la borne inférieure de la décroissance de (2.78) au moyen de transformations 2×2 dont il calcule les solutions analytiquement. Notons que ce critère est à invariance d'échelle (tout comme (2.76)) et de plus, il ne nécessite aucune contrainte sur \mathbf{B} . Cependant, comme il est logarithmique, il ne peut être utilisé que sous contrainte de positivité des matrices cibles. Une variante pondérée de ce critère est utilisée par Joho, qui propose de l'optimiser à l'aide d'une méthode de Newton [62]. Todros et Tabrikian proposent, de leur côté, d'optimiser une approximation de la log-vraisemblance [115].

2.5 Méthodes proposées pour les algorithmes développés

Dans ce paragraphe, nous présentons les approches proposées, communes aux algorithmes de diagonalisation conjointe non-unitaire de matrices (chapitre 3) et de tenseurs d'ordre trois (chapitre 4) développés au cours de cette thèse.

2.5.1 Procédure de Jacobi

Nous proposons des algorithmes basés sur une procédure itérative de type Jacobi. Comme il a été expliqué au paragraphe 2.4.1, on rappelle que cette procédure permet de décomposer la matrice diagonalisante \mathbf{B} de dimensions $N_s \times N_s$ en un produit de $\frac{N_s(N_s-1)}{2}$ matrices élémentaires (2.72) notées \mathbf{B}^{ij} et définies par (2.73). Ces matrices élémentaires sont de dimensions $N_s \times N_s$ et ne diffèrent de la matrice identité que par leurs quatre composantes situées aux intersections des i -ème et j -ème lignes et colonnes. Ces quatre composantes utiles peuvent

être écrites sous la forme de la matrice 2×2

$$\tilde{\mathbf{B}}^{ij} = \begin{pmatrix} B_{ii}^{ij} & B_{ij}^{ij} \\ B_{ji}^{ij} & B_{jj}^{ij} \end{pmatrix}. \quad (2.79)$$

Ainsi chaque matrice élémentaire \mathbf{B}^{ij} est entièrement définie par sa matrice extraite $\tilde{\mathbf{B}}^{ij}$ que nous appellerons partie utile de la matrice. Par la suite, lorsque nous évoquerons des matrices élémentaires, nous les définirons simplement par leurs matrices extraites aux lignes et colonnes i et j et nous les représenterons à l'aide de l'opérateur $\widetilde{(\cdot)}^{ij}$. Chacune des matrices élémentaires possède donc au plus quatre paramètres qu'il faudra estimer, contre N_s^2 paramètres pour matrice \mathbf{B} . La procédure de type Jacobi, décompose donc un problème de dimensions $N_s \times N_s$ en $\frac{N_s(N_s-1)}{2}$ problèmes de dimensions 2×2 . En simplifiant le problème initial de la sorte, cette procédure permet de mettre en place des méthodes de résolution algébrique. Toutefois, il est important de noter que, comme cette procédure est itérative, si l'estimation des paramètres n'est pas faite de manière algébrique, les erreurs numériques risqueront de se multiplier entraînant une mauvaise estimation de la matrice diagonalisante. Ainsi, pour éviter toute erreur numérique, dans tous les algorithmes proposés dans cette thèse, nous chercherons systématiquement une solution analytique aux problèmes élémentaires. Un autre très grand avantage de cette procédure est la possibilité de paralléliser les calculs. En effet, dans le cas où la mise à jour des matrices cibles ou tenseurs cibles n'est faite qu'après l'estimation de toutes les matrices élémentaires, la parallélisation est évidente. Dans le cas où une mise à jour est faite après chaque estimation d'une matrice \mathbf{B}^{ij} , la gestion de l'ordre dans lequel sont estimées les matrices élémentaires est alors fondamentale.

2.5.2 Critères choisis

Ici, nous considérons la détermination des matrices élémentaires comme un problème d'optimisation. Il faut donc choisir un critère. En séparation de sources, on va, généralement, chercher à mesurer la diagonalité des matrices transformées

$$\mathbf{M}_k^{ij} = \mathbf{B}^{ij} \mathbf{M}_k \mathbf{B}^{ij \dagger}. \quad (2.80)$$

Un critère inverse est donc envisagé. Gardons aussi à l'idée que nous cherchons à mettre en place une méthode de résolution algébrique, aussi bien dans le cas matriciel que dans le cas tensoriel. Dans ce dernier cas, que nous traitons sans dépliement, nous avons affaire à des formes trilineaires dont la résolution analytique s'avère parfois peu évidente. Il est donc intéressant de choisir un critère qui ne complexifie pas le problème. Nous choisissons donc de

considérer le critère inverse classique (2.75). A partir de maintenant et dans toute la suite du manuscrit, nous le noterons simplement \mathcal{J} . Lorsque nous considérerons spécifiquement les ensembles de matrices complexes symétriques ou hermitiennes, nous dénoterons ce critère respectivement $\mathcal{J}^{(T)}$ et $\mathcal{J}^{(H)}$ et pour les ensembles de tenseurs complexes admettant une décomposition INDSCAL_h ou un modèle symétrique, nous le dénoterons respectivement $\mathcal{J}^{(h)}$ et $\mathcal{J}^{(s)}$. Appliqué à une matrice élémentaire \mathbf{B}^{ij} , ce critère s'écrit

$$\mathcal{J}(\mathbf{B}^{ij}) = \sum_{k=1}^K \|\text{ZDiag}\{\mathbf{M}_k^{ij}\}\|^2. \quad (2.81)$$

Deux approches différentes seront utilisées afin d'optimiser le critère (2.81). La première, que l'on qualifie de classique, consiste à minimiser directement (2.81). Pour la seconde, que l'on appelle approche adaptée, la même minimisation est effectuée, mais cette fois en faisant l'hypothèse que l'on se situe proche d'une solution diagonalisante. On se trouve dans cette situation lorsque toutes les composantes hors-diagonales des \mathbf{M}_k (2.80) sont très petites devant un. Nous verrons dans les chapitres 3 et 4 que, dans ce cas précis, le critère (2.81) s'écrit de façon très intéressante, permettant une estimation particulièrement simple et efficace des paramètres.

Maintenant que notre choix s'est arrêté sur un critère, il s'agit d'estimer les différentes matrices élémentaires de telle sorte que

$$\widehat{\mathbf{B}}^{ij} = \arg \min_{\mathbf{B}^{ij}} \mathcal{J}(\mathbf{B}^{ij}) \quad (2.82)$$

Toutefois, comme il est mentionné au paragraphe 2.4.2, avec l'utilisation du critère (2.81), il est nécessaire d'envisager une contrainte afin d'éviter les solutions singulières pour les matrices \mathbf{B}^{ij} et assurer l'inversibilité de la matrice de séparation \mathbf{B} .

2.5.3 Contrainte considérée

Nous considérons la contrainte, pour tout $(i, j) \in \mathcal{N}_s^2$ avec $i < j$

$$\det(\mathbf{B}^{ij}) = 1. \quad (2.83)$$

Ainsi, nous évitons les solutions singulières pour toutes les matrices élémentaires et, comme la matrice diagonalisante \mathbf{B} est égale au produit des \mathbf{B}^{ij} (2.72), alors $\det(\mathbf{B}) = 1$ et donc \mathbf{B} est inversible. Afin d'assurer (2.83), nous choisissons d'utiliser une décomposition LU pour chacune des matrices \mathbf{B}^{ij} . On peut utiliser une telle décomposition, car il est bien connu que toute matrice carrée peut s'écrire sous la forme \mathbf{DPLU} , où \mathbf{D} est une matrice diagonale, \mathbf{P} une

matrice de permutation et \mathbf{L} et \mathbf{U} sont des matrices triangulaires, respectivement inférieure et supérieure avec toutes leurs composantes diagonales égales à un. Or, en diagonalisation conjointe, la matrice diagonalisante est estimée à des facteurs d'échelle et des permutations près. \mathbf{D} et \mathbf{P} sont des indéterminations classiques, on n'en tient donc pas compte et il ne reste plus que les matrices \mathbf{L} et \mathbf{U} . Ainsi, les matrices élémentaires peuvent être décomposées de la manière suivante

$$\mathbf{B}^{ij} = \mathbf{L}^{ij}\mathbf{U}^{ij} \quad (2.84)$$

avec les parties utiles de \mathbf{L}^{ij} et \mathbf{U}^{ij} définies par

$$\tilde{\mathbf{L}}^{ij} = \begin{pmatrix} 1 & 0 \\ \ell^{ij} & 1 \end{pmatrix} \quad \text{et} \quad \tilde{\mathbf{U}}^{ij} = \begin{pmatrix} 1 & u^{ij} \\ 0 & 1 \end{pmatrix}. \quad (2.85)$$

Toutes les matrices élémentaires \mathbf{L}^{ij} et \mathbf{U}^{ij} ne dépendent donc que d'un seul paramètre. Notons que cette paramétrisation a l'avantage d'être simple et de ne pas subir de modifications, que l'on travaille dans l'ensemble des réels ou des complexes (car on peut avoir $(\ell^{ij}, u^{ij}) \in \mathbb{R}^2$ ou $(\ell^{ij}, u^{ij}) \in \mathbb{C}^2$).

A partir de (2.84) plusieurs stratégies d'estimation des paramètres peuvent être envisagées. Nous en proposons deux.

Stratégie couplée : Pour un couple (i, j) fixé, cette stratégie consiste à considérer la matrice \mathbf{B}^{ij} directement comme le résultat du produit de \mathbf{L}^{ij} par \mathbf{U}^{ij} . Sa partie utile s'écrit donc

$$\tilde{\mathbf{B}}^{ij} = \tilde{\mathbf{L}}^{ij}\tilde{\mathbf{U}}^{ij} = \begin{pmatrix} 1 & u^{ij} \\ \ell^{ij} & 1 + \ell^{ij}u^{ij} \end{pmatrix}. \quad (2.86)$$

De ce fait, chaque matrice élémentaire \mathbf{B}^{ij} dépendra des deux paramètres ℓ^{ij} et u^{ij} . Pour tout $(i, j) \in \mathcal{N}_s^2$, $i < j$, il faut donc implémenter une méthode d'optimisation algébrique telle que

$$(\hat{\ell}^{ij}, \hat{u}^{ij}) = \arg \min_{\mathbf{B}^{ij}} \mathcal{J}(\mathbf{B}^{ij}). \quad (2.87)$$

Avec cette stratégie couplée, la matrice de séparation \mathbf{B} admettra exactement la décomposition (2.72) avec chaque matrice élémentaire \mathbf{B}^{ij} caractérisée par sa partie utile (2.86).

Stratégie découplée : Avec cette stratégie, on propose d'estimer dans un premier temps les matrices \mathbf{U}^{ij} , pour chacun des couples (i, j) possibles, en imposant $\mathbf{B}^{ij} = \mathbf{U}^{ij}$ à \mathbf{L}^{ij} fixées. Puis, dans un second temps, nous faisons l'inverse pour l'estimation des \mathbf{L}^{ij} . Le but de cette stratégie est donc de mettre en place une méthode algébrique permettant d'estimer un seul

et unique paramètre (soit u^{ij} , soit ℓ^{ij}) de telle sorte que pour tout $(i, j) \in \mathcal{N}_s^2$ avec $i < j$

$$\hat{u}^{ij} = \arg \min_{\mathbf{U}^{ij}} \mathcal{J}(\mathbf{U}^{ij}) \quad \text{ou} \quad \hat{\ell}^{ij} = \arg \min_{\mathbf{L}^{ij}} \mathcal{J}(\mathbf{L}^{ij}) \quad (2.88)$$

Avec cette stratégie, la matrice de séparation \mathbf{B} admettra la décomposition suivante

$$\mathbf{B} = \left(\prod_{i=1}^{N_s-1} \prod_{j=i+1}^{N_s} \mathbf{L}^{ij} \right) \left(\prod_{i=1}^{N_s-1} \prod_{j=i+1}^{N_s} \mathbf{U}^{ij} \right). \quad (2.89)$$

Dans le cas de la stratégie découplée, on note que la décomposition LU possède un autre avantage. En effet, comme $\mathbf{L}^{ij\dagger}$ est une matrice triangulaire supérieure avec toutes ses composantes diagonales égales à un, alors les calculs pour l'estimation de \mathbf{L}^{ij} se déduiront aisément de ceux effectués pour l'estimation de \mathbf{U}^{ij} .

2.5.4 Structure générale des algorithmes proposés

Notons que, dans le but d'accélérer la convergence des algorithmes proposés, dans les chapitres suivants, nous avons décidé de mettre à jour l'ensemble de matrices cibles ou de tenseurs cibles systématiquement après l'estimation de chacune des matrices élémentaires, et cela quelle que soit la stratégie choisie. Dans l'Algorithme 1, nous présentons sous la forme d'un pseudo-code la structure générale des algorithmes que nous avons développés au cours de cette thèse.

Algorithme 1 Structure générale des algorithmes proposés

- 1: **input** : Ensemble de cibles (matrices carrées ou tenseurs d'ordre trois cubiques)
 - 2: Initialiser la matrice diagonalisante
 - 3: Mettre à jour des matrices cibles
 - 4: **repeat** le balayage suivant
 - 5: **for** $1 \leq i < j \leq N_s$ **do**
 - 6: Estimer les paramètres u^{ij} et ℓ^{ij} selon la stratégie choisie (2.87) ou (2.88)
 - 7: Mettre à jour les cibles
 - 8: Mettre à jour la matrice diagonalisante
 - 9: **end for**
 - 10: **until** convergence
 - 11: **output** : Matrice diagonalisante estimée
-

Conclusion

Au cours de ce chapitre, nous avons défini le cadre des problèmes de diagonalisation conjointe, d'ICA et de séparation de sources, ainsi que les liens entre ces différents domaines. Puis, nous avons présenté les méthodes mises en œuvre, au cours de cette thèse, pour le développement des algorithmes de diagonalisation conjointe non-unitaire proposés pour des ensembles de matrices (chapitre 3) et de tenseurs d'ordre trois (chapitre 4).

Algorithmes de diagonalisation conjointe non-unitaire de matrices

Dans ce chapitre, nous développons des méthodes pour l'estimation algébrique de paramètres dans le cadre de la diagonalisation conjointe non-unitaire de matrices. On ne s'intéressera qu'aux ensembles de matrices complexes symétriques (2.11) et aux ensembles de matrices hermitiennes (2.15). En effet, dans le cadre de la séparation de sources (notamment pour les signaux de télécommunications numériques), les modèles complexe symétrique (2.12) et hermitien (2.16) sont particulièrement utiles et utilisés pour tout type de signaux.

3.1 Introduction

Rappelons succinctement, les données du problème détaillées au paragraphe 2.1.2. On considère un ensemble de K ($K \geq 2$) matrices \mathbf{M}_k respectant le modèle générique suivant pour tout $k \in \mathcal{K}$

$$\mathbf{M}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^\dagger \quad (3.1)$$

où $\mathbf{A} \in \mathcal{A}_{N_o \times N_s}$ (2.3) avec $N_s \leq N_o$. Dans le cas complexe symétrique, on aura $(\cdot)^\dagger = (\cdot)^T$ et $\mathbf{D}_k \in \mathbb{C}^{N_s \times N_s}$ et dans le cas hermitien, on aura $(\cdot)^\dagger = (\cdot)^H$ et $\mathbf{D}_k \in \mathbb{R}^{N_s \times N_s}$. On cherche une matrice diagonalisante $\mathbf{B} \in \mathcal{B}_{N_s \times N_o}$ (2.8) telle que les matrices transformées

$$\mathbf{M}'_k = \mathbf{B}\mathbf{M}_k\mathbf{B}^\dagger \quad (3.2)$$

soient les plus diagonales possibles. Au paragraphe 2.5, nous avons présenté les méthodes mises en œuvre pour le développement des algorithmes de cette thèse. Pour chacun des deux ensembles de matrices considérées, nous proposons un algorithme de type Jacobi basé sur une paramétrisation LU de la matrice de diagonalisation. Cependant, la décomposition (2.72)

obtenue à l'aide de la procédure de Jacobi ainsi que la factorisation LU (2.84) se justifient pour des matrices carrées. Or, on considère, ici, une matrice diagonalisante de dimensions $N_s \times N_o$. En supposant que nous connaissons les dimensions du problème, une solution est d'initialement décomposer la matrice \mathbf{B} comme

$$\mathbf{B} = \mathbf{B}_r \mathbf{B}_o \quad (3.3)$$

où $\mathbf{B}_o \in \mathbb{C}^{N_s \times N_o}$ est une matrice de rang plein en ligne et $\mathbf{B}_r \in \mathbb{C}^{N_s \times N_s}$ est une matrice inversible. La matrice \mathbf{B}_o correspond à une initialisation de la matrice diagonalisante \mathbf{B} et \mathbf{B}_r est appelée matrice diagonalisante réduite. Une fois l'initialisation $\mathbf{B} \leftarrow \mathbf{B}_o$ faite, les matrices \mathbf{M}_k (3.1) de dimensions $N_o \times N_o$ sont mises à jour par la transformation suivante

$$\mathbf{R}_k = \mathbf{B}_o \mathbf{M}_k \mathbf{B}_o^\dagger \quad (3.4)$$

où $\mathbf{R}_k \in \mathbb{C}^{N_s \times N_s}$.

Remarques :

- En séparation aveugle de sources, on connaît le nombre d'observations N_o , mais le nombre de sources N_s peut être inconnu. Toutefois, il existe des méthodes d'estimation de N_s . L'une d'elles, relativement simple, consiste à prendre deux matrices de l'ensemble des \mathbf{M}_k (3.1) et calculer leur GEVD. Idéalement, en l'absence de bruit, le nombre de sources correspond au nombre de valeurs propres non-nulles. En présence de bruit, les plus petites valeurs propres correspondent à la contribution du bruit. Si les M_k considérées sont des matrices de covariance, les modules de ces valeurs propres sont de l'ordre de la puissance du bruit. Il est possible d'établir un seuil au-dessus duquel le module des valeurs propres correspond à la contribution des différentes sources. Ainsi on détermine N_s .
- En séparation de sources, la transformation (3.4) permettant une réduction de dimensions est appelée projection sur l'espace signal.

Finalement, le problème de diagonalisation conjointe est le suivant. A partir de l'ensemble des nouvelles matrices cibles \mathbf{R}_k , nous allons estimer la matrice diagonalisante réduite \mathbf{B}_r à l'aide d'une procédure itérative de type Jacobi. Nous décomposons \mathbf{B}_r en produit de matrices élémentaires

$$\mathbf{B}_r = \prod_{i=1}^{N_s-1} \prod_{j=i+1}^{N_s} \mathbf{B}_r^{ij}. \quad (3.5)$$

Chacune des matrices élémentaires est ensuite estimée en optimisant le critère inverse

$$\mathcal{J}(\mathbf{B}_r^{ij}) = \sum_{k=1}^K \left\| \text{ZDiag}\{\mathbf{B}_r^{ij} \mathbf{R}_k \mathbf{B}_r^{ij \dagger}\} \right\|^2. \quad (3.6)$$

Deux approches sont considérées :

- Une approche que l'on qualifera de classique, où l'on minimise simplement le critère précédent appliqué aux matrices transformées

$$\mathbf{R}'_k = \mathbf{B}_r^{ij} \mathbf{R}_k \mathbf{B}_r^{ij \dagger} \quad (3.7)$$

sans hypothèse particulière.

- Une approche adaptée, décrite au paragraphe 2.5.2, qui consiste à minimiser le même critère (3.6) mais en supposant, cette fois, que l'on est déjà proche d'une solution diagonalisante.

Afin d'assurer la contrainte $\det(\mathbf{B}_r^{ij}) = 1$ et la pseudo-inversibilité de \mathbf{B} , nous avons décidé d'utiliser une décomposition LU sur les matrices \mathbf{B}_r^{ij} . La partie utile de \mathbf{B}_r^{ij} s'écrit donc

$$\tilde{\mathbf{B}}_r^{ij} = \tilde{\mathbf{L}}^{ij} \tilde{\mathbf{U}}^{ij} = \begin{pmatrix} 1 & 0 \\ \ell^{ij} & 1 \end{pmatrix} \begin{pmatrix} 1 & u^{ij} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & u^{ij} \\ \ell^{ij} & 1 + \ell^{ij} u^{ij} \end{pmatrix}. \quad (3.8)$$

Deux stratégies d'estimation des paramètres des matrices \mathbf{L}^{ij} et \mathbf{U}^{ij} sont également envisagées :

- La stratégie découplée, qui consiste à estimer $\mathbf{B}_r^{ij} = \mathbf{U}^{ij}$ à \mathbf{L}^{ij} fixée, pour tous les couples (i, j) avec $1 \leq i < j \leq N_s$, puis à faire de même pour l'estimation des \mathbf{L}^{ij} . Ainsi l'estimation de chacun des paramètres ℓ^{ij} et u^{ij} est faite de façon indépendante.
- La stratégie couplée, où pour chaque couple (i, j) tel que $1 \leq i < j \leq N_s$, une matrice $\mathbf{B}_r^{ij} = \mathbf{L}^{ij} \mathbf{U}^{ij}$ est estimée. Ainsi, on cherche la valeur optimale du couple de paramètres (ℓ^{ij}, u^{ij}) .

Finalement, quel que soit l'ensemble de matrices considéré (complexe symétrique ou hermitien), il y a quatre méthodes d'estimation possibles, correspondant aux combinaisons des deux approches envisagées liées au critère et des deux stratégies d'estimation des paramètres.

1. Une méthode basée sur l'approche classique et la stratégie découplée a été proposée par Afsari dans [1] dans le cas réel, puis a été étendue pour des ensembles de matrices hermitiennes par Wang *et al.* dans [127]. Cette dernière est résumée dans le paragraphe 3.2.1.2. Nous avons généralisé l'algorithme LUJ1D [1] pour des ensembles de matrices complexes symétriques dans le paragraphe 3.2.1.1.

2. Dans le paragraphe 3.2.2, nous proposons l'algorithme ALUJA (pour *Alternate LU Jacobi-like Algorithm*) basé sur l'optimisation du critère inverse selon l'approche adaptée et reposant sur la stratégie découplée. Pour cet algorithme, la minimisation du critère se fait par la recherche de racines de sa dérivée.
3. Dans le paragraphe 3.3.1, nous proposons un algorithme basé sur la stratégie d'estimation couplée, nommé G-CLUJA (pour *Gradient descent method for Coupled LU Jacobi-like Algorithm*) et qui consiste à optimiser le critère inverse selon l'approche classique. L'estimation des paramètres se fait par une méthode du gradient avec le calcul d'un pas optimal.
4. Une méthode d'estimation concernant l'approche adaptée, combinée à une stratégie d'estimation couplée des paramètres est proposée dans le paragraphe 3.3.2 et l'algorithme en résultant est noté A-CLUJA (pour *Adapted approach for Coupled LU Jacobi-like Algorithm*). Pour cet algorithme, l'estimation des paramètres se fait par une méthode de recherche de vecteur propre.

Le tableau 3.1 permet d'associer à chaque approche et à chaque stratégie son algorithme correspondant, ainsi que le paragraphe ou la référence dans lequel le cœur de l'algorithme est développé. Le symbole § signifie paragraphe, (\mathbb{R}) et (\mathbb{C}) précisent que l'algorithme est développé dans le cas réel ou dans le cas complexe respectivement.

Stratégie Approche	Découplée	Couplée
Classique	Modèle symétrique (\mathbb{R}) : LUJ1D [1] Modèle symétrique (\mathbb{C}) : §3.2.1.1 Modèle hermitien (\mathbb{C}) : LUJCD [127] décrit §3.2.1.2	G-CLUJA : Modèle symétrique (\mathbb{C}) : §3.3.1.1 Modèle hermitien (\mathbb{C}) : §3.3.1.2
Adaptée	ALUJA : Modèle symétrique (\mathbb{C}) : §3.2.2.1 Modèle hermitien (\mathbb{C}) : §3.2.2.2	A-CLUJA : Modèle symétrique (\mathbb{C}) : §3.3.2.1 Modèle hermitien (\mathbb{C}) : §3.3.2.2

TABLE 3.1 – Récapitulatifs des algorithmes matriciels de diagonalisation conjointe non-unitaire proposés selon l'approche liée au critère d'optimisation et la stratégie d'estimation des paramètres choisies

Dans les paragraphes suivants de ce chapitre, nous décrivons les méthodes d'estimation des matrices élémentaires réduites \mathbf{B}_r^{ij} pour chacun des algorithmes développés. Pour une meilleure lisibilité, nous n'écrirons plus l'indice r des matrices \mathbf{B}_r^{ij} . Pour un couple (i, j) , $i < j$ donné, définissons les ensembles suivants qui nous seront utiles par la suite :

$$\mathcal{P} = \mathcal{N}_s \setminus \{i\} \quad \text{et} \quad \mathcal{Q} = \mathcal{N}_s \setminus \{i, j\}. \quad (3.9)$$

3.2 Stratégie d'estimation découplée

Commençons par rappeler que les matrices \mathbf{U}^{ij} et \mathbf{L}^{ij} ne diffèrent de la matrice identité \mathbf{I}_{N_s} que par un seul et unique paramètre hors-diagonal situé à la position (i, j) pour \mathbf{U}^{ij} et (j, i) pour \mathbf{L}^{ij} . Les matrices 2×2 extraites aux intersections de leurs lignes et de leurs colonnes i et j sont appelées parties utiles de \mathbf{U}^{ij} et \mathbf{L}^{ij} et sont respectivement définies par

$$\tilde{\mathbf{U}}^{ij} = \begin{pmatrix} 1 & u^{ij} \\ 0 & 1 \end{pmatrix} \quad \text{et} \quad \tilde{\mathbf{L}}^{ij} = \begin{pmatrix} 1 & 0 \\ \ell^{ij} & 1 \end{pmatrix}. \quad (3.10)$$

Dans ce paragraphe, pour un couple (i, j) , $i < j$ donné, le but est d'estimer le paramètre u^{ij} (caractérisant complètement la matrice \mathbf{U}^{ij}) de telle sorte que

$$\hat{u}^{ij} = \arg \min_{\mathbf{U}^{ij}} \mathcal{J}(\mathbf{U}^{ij}). \quad (3.11)$$

La matrice \mathbf{L}^{ij} étant la transposée d'une matrice triangulaire supérieure similaire à \mathbf{U}^{ij} , on déduira donc directement la solution optimale pour le paramètre ℓ^{ij} , en interchangeant simplement les indices i et j dans les développements suivants.

Pour un ensemble de K matrices génériques $\mathbf{R}_k \in \mathbb{C}^{N_s \times N_s}$ respectant le modèle complexe symétrique (2.12) ou hermitien (2.16), la stratégie d'estimation découplée, repose donc sur la minimisation du critère

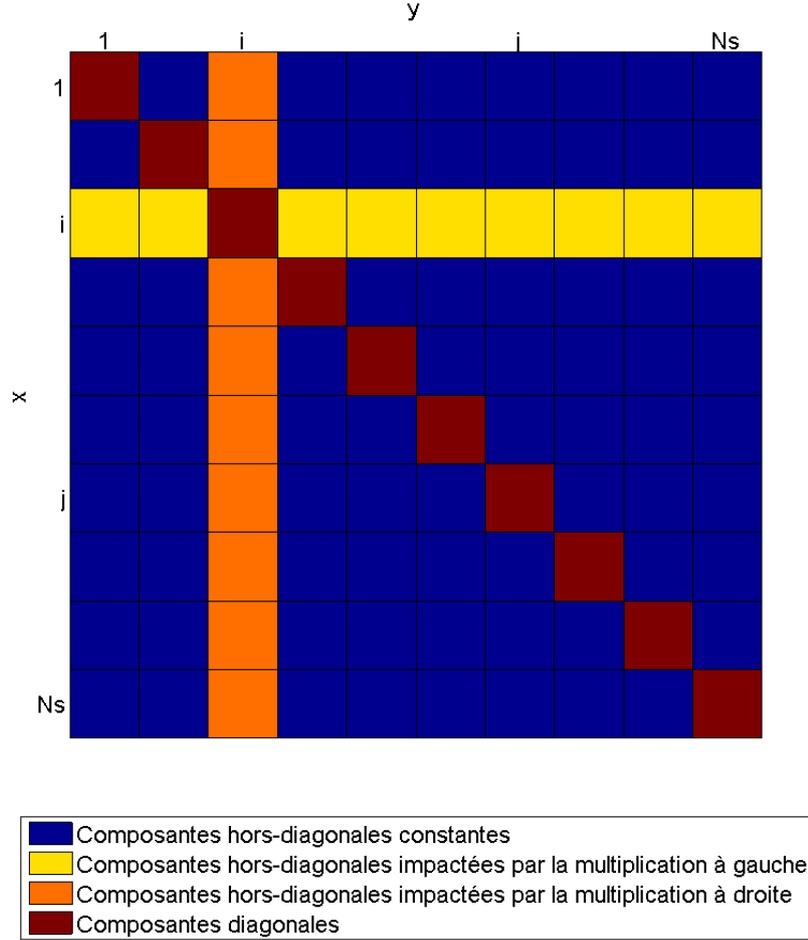
$$\mathcal{J}(\mathbf{U}^{ij}) = \sum_{k=1}^K \|\text{ZDiag}\{\mathbf{R}'_k\}\|^2, \quad (3.12)$$

avec

$$\mathbf{R}'_k = \mathbf{U}^{ij} \mathbf{R}_k \mathbf{U}^{ij \dagger}. \quad (3.13)$$

Dans la transformation (3.13), la multiplication à gauche par \mathbf{U}^{ij} ne modifie que la ligne i de chaque \mathbf{R}_k , tandis que la multiplication à droite par $\mathbf{U}^{ij \dagger}$ ne modifie que leur colonne i . Afin d'illustrer l'impact de cette transformation, la figure 3.1 schématise une matrice transformée \mathbf{R}'_k et met en évidence ses composantes hors-diagonales ayant été modifiées linéairement par le paramètre u^{ij} par rapport à la matrice-cible \mathbf{R}_k correspondante. De ce fait, on peut décomposer (3.12) de la manière suivante

$$\mathcal{J}(\mathbf{U}^{ij}) = \mathcal{J}'(\mathbf{U}^{ij}) + C_{dc}, \quad (3.14)$$

FIGURE 3.1 – Schéma de l'impact de la transformation (3.13) sur une matrice \mathbf{R}'_k .

avec

$$\mathcal{J}'(\mathbf{U}^{ij}) = \sum_{k=1}^K \sum_{l \in \mathcal{P}} \left(|R'_{k,il}|^2 + |R'_{k,li}|^2 \right) \quad (3.15)$$

$$C_{dc} = \sum_{k=1}^K \sum_{\substack{(m,n) \in \mathcal{P}^2 \\ m \neq n}} |R'_{k,mn}|^2. \quad (3.16)$$

$\mathcal{J}'(\mathbf{U}^{ij})$ regroupe tous les termes hors-diagonaux des \mathbf{R}'_k linéaires par rapport à u^{ij} (en jaune et orange sur la figure 3.1) et C_{dc} regroupe tous leurs termes hors-diagonaux constants (en bleu sur la figure 3.1). Finalement, minimiser $\mathcal{J}(\mathbf{U}^{ij})$ revient donc à minimiser $\mathcal{J}'(\mathbf{U}^{ij})$. On

cherche donc à estimer le paramètre u^{ij} optimal de telle manière que

$$\hat{u}^{ij} = \arg \min_{\mathbf{U}^{ij}} \mathcal{J}'(\mathbf{U}^{ij}). \quad (3.17)$$

Maintenant que l'objectif est défini pour la stratégie d'estimation découplée, nous allons calculer la valeur optimale du paramètre u^{ij} , d'abord dans le cadre de l'approche classique, puis pour l'approche adaptée.

3.2.1 Approche classique

Cette approche consiste, basiquement, à minimiser le critère (3.15). Dans ce paragraphe, on détaille une méthode algébrique utilisant l'approche classique, dans un premier temps, pour un ensemble de matrices cibles suivant un modèle complexe symétrique (2.12), puis dans un second temps, pour un ensemble de matrices respectant le modèle hermitien (2.16).

3.2.1.1 Modèle complexe symétrique

Bien que cette méthode n'ait pas été développée pour le modèle complexe symétrique dans la littérature, elle est similaire aux travaux menés par Afsari dans [1].

On s'intéresse ici à la diagonalisation conjointe non-unitaire d'un ensemble de matrices complexes symétriques $\mathbf{R}_k^{(s)}$ de dimensions $N_s \times N_s$ suivant le modèle (2.12). Ainsi, pour tout $k \in \mathcal{K}$ et pour tout $(a, b) \in \mathcal{N}_s^2$, on a

$$R_{k,ab}^{(s)} = R_{k,ba}^{(s)}. \quad (3.18)$$

Pour un tel ensemble de matrices cibles, le critère (3.15), noté ici $\mathcal{J}'^{(T)}$, s'écrit

$$\mathcal{J}'^{(T)}(\mathbf{U}^{ij}) = 2 \sum_{k=1}^K \sum_{l \in \mathcal{P}} |R_{k,il}^{(s)}|^2 \quad (3.19)$$

avec

$$\mathbf{R}_k'^{(s)} = \mathbf{U}^{ij} \mathbf{R}_k^{(s)} \mathbf{U}^{ijT}. \quad (3.20)$$

Notons que par la transformation (3.20), les matrices $\mathbf{R}_k'^{(s)}$ sont, bien entendu, elles aussi symétriques.

Pour tout $k \in \mathcal{K}$ et pour tout $l \in \mathcal{P}$, les éléments hors-diagonaux des $\mathbf{R}_k'^{(s)}$ à la ligne i s'écrivent

$$R_{k,il}'^{(s)} = R_{k,il}^{(s)} + R_{k,jl}^{(s)} u^{ij}. \quad (3.21)$$

On peut maintenant développer le critère (3.19) de la manière suivante

$$\mathcal{J}^{(T)}(\mathbf{U}^{ij}) = 2 \sum_{k=1}^K \sum_{l \in \mathcal{P}} |R_{k,il}^{(s)} + R_{k,jl}^{(s)} u^{ij}|^2. \quad (3.22)$$

On obtient donc pour (3.22), un polynôme d'ordre deux en u^{ij} . Pour déterminer le paramètre u^{ij} minimisant $\mathcal{J}^{(T)}(\mathbf{U}^{ij})$, on dérive l'expression (3.22) par rapport au conjugué de u^{ij} comme il suit

$$\frac{\partial \mathcal{J}^{(T)}(\mathbf{U}^{ij})}{\partial u^{ij*}} = 2 \sum_{k=1}^K \sum_{l \in \mathcal{P}} R_{k,jl}^{(s)*} (R_{k,il}^{(s)} + R_{k,jl}^{(s)} u^{ij}). \quad (3.23)$$

Finalement, en égalisant (3.23) à zéro, on obtient la valeur optimale

$$\hat{u}^{ij} = - \frac{\sum_{k=1}^K \sum_{l \in \mathcal{P}} R_{k,il}^{(s)} R_{k,jl}^{(s)*}}{\sum_{k=1}^K \sum_{l \in \mathcal{P}} |R_{k,jl}^{(s)}|^2}. \quad (3.24)$$

Comme précisé dans l'introduction du paragraphe 3.2, la valeur optimale de ℓ^{ij} est calculée de manière similaire à celle de u^{ij} . On peut la déduire directement de (3.24) en interchangeant tous les indices i et j dans l'expression ainsi que dans la définition de l'ensemble \mathcal{P} (3.9). Soit l'ensemble $\mathcal{P}' = \mathcal{N}_s \setminus \{j\}$, la valeur optimale de ℓ^{ij} s'écrit

$$\hat{\ell}^{ij} = - \frac{\sum_{k=1}^K \sum_{l \in \mathcal{P}'} R_{k,jl}^{(s)} R_{k,il}^{(s)*}}{\sum_{k=1}^K \sum_{l \in \mathcal{P}'} |R_{k,il}^{(s)}|^2}. \quad (3.25)$$

Maintenant que nous avons illustré comment déduire $\hat{\ell}^{ij}$ à partir de \hat{u}^{ij} , nous ne donnerons plus l'expression de $\hat{\ell}^{ij}$ au cours des prochains développements.

3.2.1.2 Modèle hermitien

On s'intéresse, maintenant, à la diagonalisation conjointe non-unitaire d'un ensemble de matrices $\mathbf{R}_k^{(h)}$ de dimensions $N_s \times N_s$ respectant le modèle (2.16) avec pour tout $k \in \mathcal{K}$, $\mathbf{D}_k^{(h)} \in \mathbb{C}^{N_s \times N_s}$. Les matrices $\mathbf{R}_k^{(h)}$ ne sont donc pas hermitiennes à strictement parler.

Ici, nous nous contentons de résumer les résultats importants des travaux menés par

Wang *et al.* [127]. On considère exactement le critère (3.15), que l'on notera ici

$$\mathcal{J}'^{(H)}(\mathbf{U}^{ij}) = \sum_{k=1}^K \sum_{l \in \mathcal{P}} \left(|R_{k,il}^{(h)}|^2 + |R_{k,li}^{(h)}|^2 \right). \quad (3.26)$$

avec

$$\mathbf{R}_k^{(h)} = \mathbf{U}^{ij} \mathbf{R}_k^{(h)} \mathbf{U}^{ijH}. \quad (3.27)$$

Pour tout $k \in \mathcal{K}$ et pour tout $l \in \mathcal{P}$, les composantes des $\mathbf{R}_k^{(h)}$ concernées par (3.26) s'écrivent

$$\begin{aligned} R_{k,il}^{(h)} &= R_{k,il}^{(h)} + R_{k,jl}^{(h)} u^{ij} \\ R_{k,li}^{(h)} &= R_{k,li}^{(h)} + R_{k,lj}^{(h)} u^{ij*} \end{aligned} \quad (3.28)$$

De ce fait, (3.26) devient

$$\mathcal{J}'^{(H)}(\mathbf{U}^{ij}) = \sum_{k=1}^K \sum_{l \in \mathcal{P}} \left(|R_{k,il}^{(h)} + R_{k,jl}^{(h)} u^{ij}|^2 + |R_{k,li}^{(h)} + R_{k,lj}^{(h)} u^{ij*}|^2 \right). \quad (3.29)$$

L'expression (3.29) est dérivée par rapport à u^{ij*}

$$\frac{\partial \mathcal{J}'^{(H)}(\mathbf{U}^{ij})}{\partial u^{ij*}} = \sum_{k=1}^K \sum_{l \in \mathcal{P}} \left(R_{k,jl}^{(h)*} \left(R_{k,il}^{(h)} + u^{ij} R_{k,jl}^{(h)} \right) + R_{k,lj}^{(h)} \left(R_{k,li}^{(h)*} + u^{ij} R_{k,lj}^{(h)*} \right) \right). \quad (3.30)$$

Egalisant (3.30) à zéro, on obtient, finalement, la valeur optimale

$$\hat{u}^{ij} = - \frac{\sum_{k=1}^K \sum_{l \in \mathcal{P}} \left(R_{k,il}^{(h)} R_{k,jl}^{(h)*} + R_{k,li}^{(h)*} R_{k,lj}^{(h)} \right)}{\sum_{k=1}^K \sum_{l \in \mathcal{P}} \left(|R_{k,jl}^{(h)}|^2 + |R_{k,lj}^{(h)}|^2 \right)}. \quad (3.31)$$

3.2.2 Approche adaptée

Nous allons voir maintenant, qu'avec une approche différente, nous pouvons estimer les paramètres u^{ij} et ℓ^{ij} de manière encore plus simple mais non moins efficace, comme le montreront les simulations numériques.

Considérons, dans un premier temps, K matrices cibles génériques \mathbf{R}_k de dimensions $N_s \times N_s$ et respectant soit le modèle complexe symétrique (2.12), soit le modèle hermitien (2.16). L'estimation de la matrice \mathbf{U}^{ij} à partir des \mathbf{R}_k selon l'approche adaptée consiste à minimiser le même critère que pour l'approche classique (3.15) en supposant toutefois que l'on se trouve proche d'une solution diagonalisante. De telles solutions sont caractérisées par

le fait que pour tout $k \in \mathcal{K}$, toutes les composantes hors-diagonales des \mathbf{R}_k valent zéro. Ceci implique que $u^{ij} = 0$ est la solution minimisant (3.22) et (3.29) quand $\sum_{k=1}^K |R_{k,jj}|^2 \neq 0$, ce qui peut être supposé raisonnablement. Ainsi, pour tout $k \in \mathcal{K}$ et pour tout $(a, b) \in \mathcal{N}_s^2$ avec $a \neq b$, la proximité d'une solution diagonalisante peut se traduire mathématiquement par

$$|R_{k,ab}| \ll 1 \quad \text{et} \quad |u^{ij}| \ll 1. \quad (3.32)$$

Calculons, maintenant, les valeurs optimales des paramètres, d'abord pour le modèle complexe symétrique, puis pour le modèle hermitien.

3.2.2.1 Modèle complexe symétrique

On considère ici l'ensemble $\{\mathbf{R}_k^{(s)}\}$ de K matrices admettant comme décomposition (2.12) et vérifiant la propriété (3.18). Comme dans l'approche classique, pour calculer la valeur optimale du paramètre u^{ij} , on minimise, au sens des moindres carrés, les composantes hors-diagonales situées sur la ligne i des matrices transformées $\mathbf{R}'_k^{(s)}$ (3.21). Cependant, lorsqu'on fait l'hypothèse d'être proche d'une solution diagonalisante, on obtient les propriétés très intéressantes (3.32). Ainsi, pour tout $l \in \mathcal{Q}$ (3.9), on a

$$|R_{k,jl}^{(s)} u^{ij}| \ll |R_{k,il}^{(s)}|. \quad (3.33)$$

De ce fait, (3.21) peut être approximée pour tout $k \in \mathcal{K}$ et $l \in \mathcal{Q}$ par

$$\begin{aligned} R'_{k,ij}{}^{(s)} &= R_{k,ij}{}^{(s)} + R_{k,jj}{}^{(s)} u^{ij} \\ R'_{k,il}{}^{(s)} &\approx R_{k,il}{}^{(s)} \end{aligned} \quad (3.34)$$

Par l'utilisation de l'approche adaptée, la transformation (3.20) ne modifie donc approximativement que les termes situés aux positions (i, j) et (j, i) pour chacune des matrices cibles $\mathbf{R}_k^{(s)}$. L'impact du paramètre u^{ij} sur une matrice $\mathbf{R}_k^{(s)}$, lorsque l'on utilise l'approche adaptée, est schématisé par la figure 3.2.

Développons, maintenant, le critère (3.22) à l'aide de (3.34). Nous dénotons le critère utilisé avec l'approche adaptée $\mathcal{J}_a^{(T)}$ et celui-ci vaut

$$\mathcal{J}_a^{(T)}(\mathbf{U}^{ij}) = 2 \sum_{k=1}^K \left(|R_{k,ij}^{(s)} + R_{k,jj}^{(s)} u^{ij}|^2 \right) + C_{das} \approx \mathcal{J}'^{(T)}(\mathbf{U}^{ij}) \quad (3.35)$$

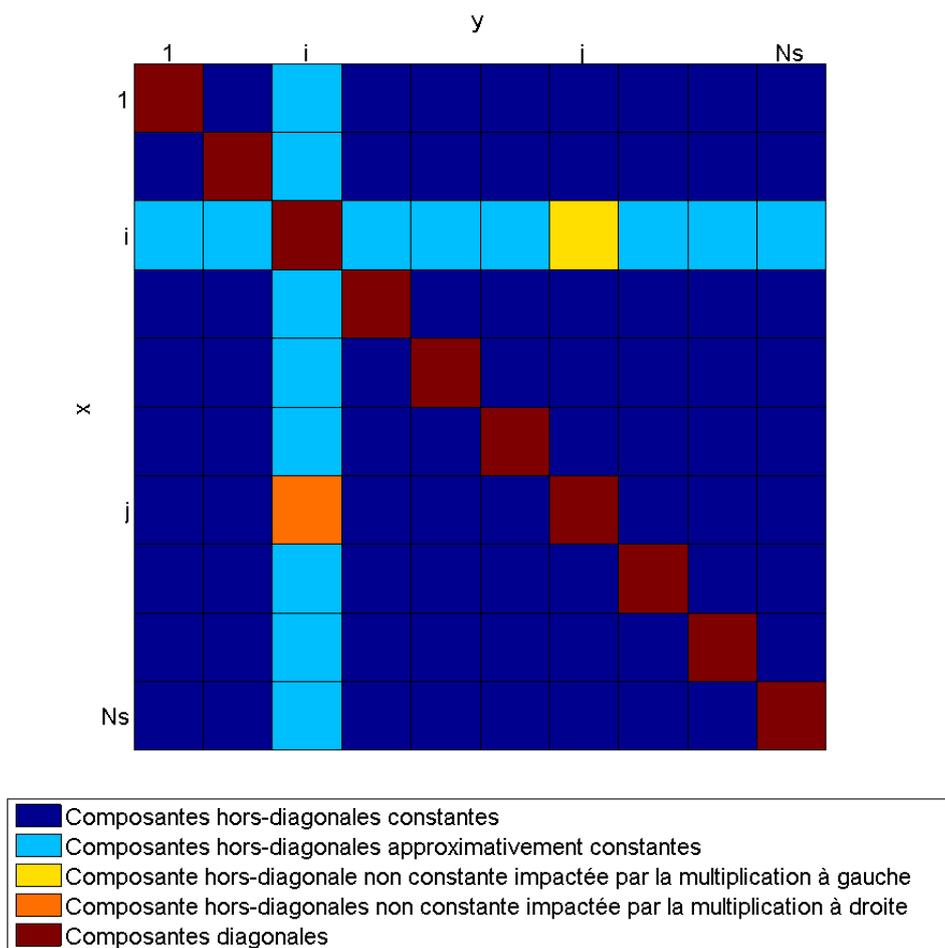


FIGURE 3.2 – Schéma de l'impact de la transformation (3.20) sur une matrice $\mathbf{R}_k^{(s)}$ lorsqu'on utilise l'approche adaptée.

où

$$C_{das} = 2 \sum_{k=1}^K \sum_{l \in \mathcal{Q}} |R_{k,il}^{(s)}|^2. \quad (3.36)$$

Le critère (3.35) s'écrit donc sous la forme d'un polynôme d'ordre deux en u^{ij} dont l'expression est bien plus simple que (3.22). Calculons u^{ij} minimisant $\mathcal{J}_a'^{(T)}(\mathbf{U}^{ij})$. Pour cela, on dérive (3.35) par rapport à u^{ij*} , ainsi on a

$$\frac{\partial \mathcal{J}_a'^{(T)}(\mathbf{U}^{ij})}{\partial u^{ij*}} = 2 \sum_{k=1}^K R_{k,jj}^{(s)*} \left(R_{k,ij}^{(s)} + R_{k,jj}^{(s)} u^{ij} \right). \quad (3.37)$$

Enfin, on égalise (3.37) à zéro pour obtenir la valeur optimale

$$\hat{u}^{ij} = -\frac{\sum_{k=1}^K R_{k,ij}^{(s)} R_{k,jj}^{(s)*}}{\sum_{k=1}^K |R_{k,jj}^{(s)}|^2}. \quad (3.38)$$

3.2.2.2 Modèle hermitien

On considère, ici, le même ensemble $\{\mathbf{R}_k^{(h)}\}$ de K matrices complexes qu'au paragraphe 3.2.1.2. Comme on suppose que l'on se trouve proche d'une solution diagonalisante, alors les relations (3.32) sont vérifiées. Pour tout $k \in \mathcal{K}$ et tout $l \in \mathcal{Q}$, on a alors

$$\begin{aligned} |R_{k,jl}^{(h)} u^{ij}| &\ll |R_{k,il}^{(h)}| \\ |R_{k,lj}^{(h)} u^{ij*}| &\ll |R_{k,li}^{(h)}|. \end{aligned} \quad (3.39)$$

Ainsi, les éléments hors diagonaux des $\mathbf{R}_k'^{(h)}$ (3.28) peuvent être approximés pour tout $k \in \mathcal{K}$ et tout $l \in \mathcal{Q}$ par

$$\begin{aligned} R_{k,ij}'^{(h)} &= R_{k,ij}^{(h)} + R_{k,jj}^{(h)} u^{ij} \\ R_{k,il}'^{(h)} &\approx R_{k,il}^{(h)} \\ R_{k,ji}'^{(h)} &= R_{k,ji}^{(h)} + R_{k,jj}^{(h)} u^{ij*} \\ R_{k,li}'^{(h)} &= R_{k,li}^{(h)} \end{aligned} \quad (3.40)$$

Comme pour le modèle symétrique, dans le cadre de l'approche adaptée, seules les composantes situées aux positions (i, j) et (j, i) de chacune des matrices transformées $\mathbf{R}_k'^{(h)}$ sont intéressantes pour le calcul du paramètre u^{ij} (voir figure 3.2).

A partir des équations (3.40), on développe le critère (3.26), dénoté ici $\mathcal{J}_a'^{(H)}$ pour l'approche adaptée. On obtient

$$\mathcal{J}_a'^{(H)}(\mathbf{U}^{ij}) = \sum_{k=1}^K \left(|R_{k,ij}^{(h)} + R_{k,jj}^{(h)} u^{ij}|^2 + |R_{k,ji}^{(h)} + R_{k,jj}^{(h)} u^{ij*}|^2 \right) + C_{dah} \approx \mathcal{J}'^{(H)}(\mathbf{U}^{ij}). \quad (3.41)$$

où

$$C_{dah} = \sum_{k=1}^K \sum_{l \in \mathcal{Q}} \left(|R_{k,il}^{(h)}|^2 + |R_{k,li}^{(h)}|^2 \right). \quad (3.42)$$

Le critère (3.41) s'écrit donc sous la forme d'un polynôme d'ordre deux en u^{ij} très simple. Afin de calculer la valeur optimale de u^{ij} , dérivons $\mathcal{J}_a'^{(H)}(\mathbf{U}^{ij})$ par rapport à u^{ij*} . Cette dérivée

s'écrit

$$\frac{\partial \mathcal{J}_a^{(H)}(\mathbf{U}^{ij})}{\partial u^{ij*}} = \sum_{k=1}^K \left(R_{k,jj}^{(h)*} \left(R_{k,ij}^{(h)} + R_{k,jj}^{(h)} u^{ij} \right) + R_{k,jj}^{(h)} \left(R_{k,ji}^{(h)*} + R_{k,jj}^{(h)*} u^{ij} \right) \right). \quad (3.43)$$

Finalement, la valeur optimale de u^{ij} est obtenue en égalisant (3.43) à zéro et elle vaut

$$\hat{u}^{ij} = - \frac{\sum_{k=1}^K \left(R_{k,ij}^{(h)} R_{k,jj}^{(h)*} + R_{k,ji}^{(h)*} R_{k,jj}^{(h)} \right)}{2 \sum_{k=1}^K |R_{k,jj}^{(h)}|^2}. \quad (3.44)$$

Remarque : Notons qu'avec le modèle hermitien (2.16), si pour tout $k \in \mathcal{K}$, $\mathbf{D}_k^{(h)} \in \mathbb{R}^{N_s \times N_s}$ alors les matrices $\mathbf{R}_k^{(h)}$ sont hermitiennes et, par la transformation (3.27), $\mathbf{R}_k^{\prime(h)}$ le sont aussi. Ainsi, pour tout $k \in \mathcal{K}$ et pour tout $(a, b) \in \mathcal{N}_s^2$, on a la propriété suivante

$$R_{k,ab}^{(h)} = R_{k,ba}^{(h)*}. \quad (3.45)$$

Par cette symétrie hermitienne, pour (3.44), on a

$$R_{k,ji}^{(h)*} = R_{k,ij}^{(h)} \quad \text{et} \quad R_{k,jj}^{(h)} = R_{k,jj}^{(h)*}. \quad (3.46)$$

Finalement, lorsque les matrices cibles sont hermitiennes, la valeur optimale du paramètre u^{ij} (3.44) s'écrit simplement

$$\hat{u}^{ij} = - \frac{\sum_{k=1}^K R_{k,ij}^{(h)} R_{k,jj}^{(h)*}}{\sum_{k=1}^K |R_{k,jj}^{(h)}|^2}. \quad (3.47)$$

Il est aussi possible de retomber dans ce cas de figure, si pour tout $k \in \mathcal{K}$, $\mathbf{D}_k^{(h)} \in \mathbb{C}^{N_s \times N_s}$, en ne travaillant que sur la partie hermitienne des matrices cibles (2.17). Il est intéressant de constater que, dans ce cas précis, la solution optimale du problème s'écrit de la même façon que pour le modèle complexe symétrique (3.38).

3.2.2.3 Description de l'algorithme

Le pseudo-code ci-dessous (algorithme 2) décrit l'algorithme générique résultant des précédents calculs. Il a pour dénomination ALUJA (pour *Alternate LU Jacobi-like Algorithm*).

Algorithme 2 ALUJA

```

1: input :  $\{\mathbf{M}_k\} \subset \mathcal{M}^{(s)}$  (2.11) ou  $\{\mathbf{M}_k\} \subset \mathcal{M}^{(h)}$  (2.15)

2: Initialiser  $\mathbf{B} \leftarrow \mathbf{B}_o$  de dimensions  $N_s \times N_o$ 
3: Mettre à jour les matrices  $\mathbf{M}_k$  suivant (3.4)
4: repeat
5:   for  $1 \leq i < j \leq N_s$  do ▷ Etape U
6:     Calculer  $u^{ij}$  en utilisant (3.38) ou (3.44)
7:     Construire  $\mathbf{U}^{ij}$ 
8:     Mettre à jour  $\mathbf{R}_k \leftarrow \mathbf{U}^{ij} \mathbf{R}_k \mathbf{U}^{ij \dagger}$ 
9:     Mettre à jour  $\mathbf{B} \leftarrow \mathbf{U}^{ij} \mathbf{B}$ 
10:  end for
11:  for  $1 \leq i < j \leq N_s$  do ▷ Etape L
12:    Calculer  $\ell^{ij}$  en interchangeant les  $i$  et les  $j$  dans (3.38) ou (3.44)
13:    Construire  $\mathbf{L}^{ij}$ 
14:    Mettre à jour  $\mathbf{R}_k \leftarrow \mathbf{L}^{ij} \mathbf{R}_k \mathbf{L}^{ij \dagger}$ 
15:    Mettre à jour  $\mathbf{B} \leftarrow \mathbf{L}^{ij} \mathbf{B}$ 
16:  end for
17: until convergence

18: output :  $\mathbf{B}$ 

```

3.3 Stratégie d'estimation couplée

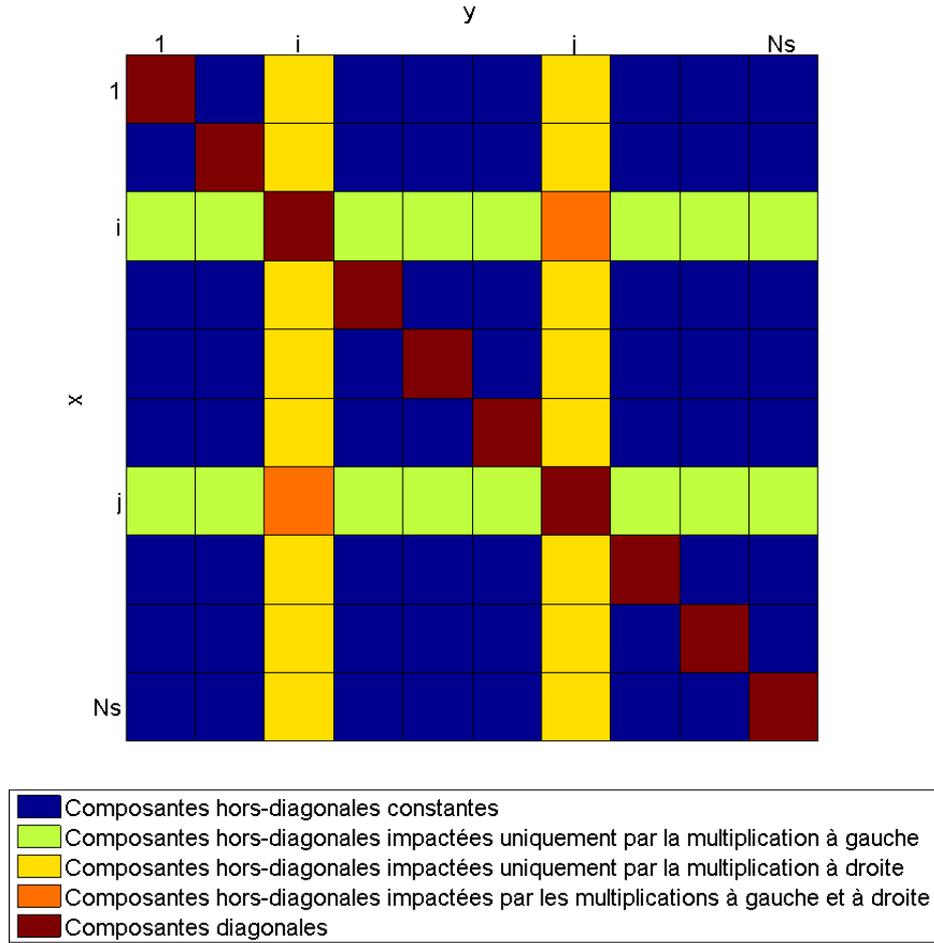
Rappelons, tout d'abord, la structure des matrices élémentaires \mathbf{B}^{ij} que nous considérons. Celles-ci ne diffèrent de la matrice identité \mathbf{I}_{N_s} que par leur partie utile définie comme

$$\tilde{\mathbf{B}}^{ij} = \begin{pmatrix} 1 & u^{ij} \\ \ell^{ij} & 1 + \ell^{ij} u^{ij} \end{pmatrix}. \quad (3.48)$$

Dans ce paragraphe, pour un couple (i, j) , $i < j$ donné, nous mettons en place des méthodes d'optimisation algébriques pour estimer simultanément les deux paramètres ℓ^{ij} et u^{ij} , caractérisant complètement la matrice élémentaire \mathbf{B}^{ij} .

Nous considérons, ici, un ensemble de K matrices génériques $\mathbf{R}_k \in \mathbb{C}^{N_s \times N_s}$. Ces matrices admettent, soit une décomposition complexe symétrique (2.12), vérifiant la propriété (3.18), soit une décomposition hermitienne (2.16), dont on ne traitera que leur partie hermitienne (2.17) et qui vérifient donc la propriété (3.45). La stratégie d'estimation couplée, repose sur la minimisation du critère

$$\mathcal{J}(\mathbf{B}^{ij}) = \sum_{k=1}^K \|\mathbf{Z} \text{Diag}\{\mathbf{R}'_k\}\|^2, \quad (3.49)$$

FIGURE 3.3 – Schéma de l'impact de la transformation (3.50) sur une matrice \mathbf{R}'_k .

avec

$$\mathbf{R}'_k = \mathbf{B}^{ij} \mathbf{R}_k \mathbf{B}^{ij \dagger}. \quad (3.50)$$

On note que les matrices \mathbf{R}'_k par la transformation (3.50) sont soit complexes symétriques, soit hermitiennes. L'estimation des paramètres u^{ij} et ℓ^{ij} est faite par l'optimisation (2.87). Dans la transformation (3.50), la multiplication à gauche par \mathbf{B}^{ij} modifie uniquement les lignes i et j des \mathbf{R}_k tandis que la multiplication à droite par $\mathbf{B}^{ij \dagger}$ modifie uniquement leurs colonnes i et j . La figure 3.3 illustre l'impact de cette transformation sur une matrice transformée \mathbf{R}'_k et met en évidence ses composantes hors-diagonales ayant été modifiées par les paramètres u^{ij} et ℓ^{ij} par rapport à la matrice-cible \mathbf{R}_k correspondante.

A l'aide de ces remarques concernant l'impact de la transformation (3.50) sur les \mathbf{R}'_k et

de leurs propriétés de symétrie, nous proposons de décomposer (3.49) comme

$$\mathcal{J}(\mathbf{B}^{ij}) = \mathcal{J}_2(\mathbf{B}^{ij}) + \mathcal{J}'(\mathbf{B}^{ij}) + C_{cc}, \quad (3.51)$$

avec

$$\mathcal{J}_2(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K |R'_{k,ij}|^2 \quad (3.52)$$

$$\mathcal{J}'(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K \sum_{q \in \mathcal{Q}} \left(|R'_{k,iq}|^2 + |R'_{k,jq}|^2 \right) \quad (3.53)$$

$$C_{cc} = 2 \sum_{k=1}^K \sum_{\substack{(m,n) \in \mathcal{Q}^2 \\ m < n}} |R'_{k,mn}|^2. \quad (3.54)$$

Avec $\mathcal{J}_2(\mathbf{B}^{ij})$, on ne considère que les composantes hors-diagonales des \mathbf{R}'_k ayant été impactées par une transformation quadratique (multiplications à gauche et à droite par la matrice élémentaire). Avec $\mathcal{J}'(\mathbf{U}^{ij})$, on considère tous les termes hors-diagonaux des \mathbf{R}'_k ayant été impactées par une transformation linéaire (soit multiplication à gauche, soit multiplication à droite par la matrice élémentaire). C_{cc} regroupe tous leurs termes hors-diagonaux constants. Finalement, minimiser $\mathcal{J}(\mathbf{U}^{ij})$ revient donc à minimiser la somme de $\mathcal{J}_2(\mathbf{B}^{ij})$ et $\mathcal{J}'(\mathbf{U}^{ij})$. On cherche donc à estimer les paramètres optimaux u^{ij} et ℓ^{ij} tels que

$$(\hat{\ell}^{ij}, \hat{u}^{ij}) = \arg \min_{\mathbf{B}^{ij}} \left(\mathcal{J}_2(\mathbf{B}^{ij}) + \mathcal{J}'(\mathbf{B}^{ij}) \right). \quad (3.55)$$

Remarque : Il est intéressant de noter que

$$\mathcal{J}_2(\mathbf{B}^{ij}) = \mathcal{J}_2(\tilde{\mathbf{B}}^{ij}) = \sum_{k=1}^K \left\| \text{ZDiag}\{\tilde{\mathbf{B}}^{ij} \tilde{\mathbf{R}}_k^{ij} \tilde{\mathbf{B}}^{ij\dagger}\} \right\|^2, \quad (3.56)$$

où $\tilde{\mathbf{R}}_k^{ij}$ sont les matrices 2×2 extraites des \mathbf{R}_k aux intersections des lignes et colonnes i et j telles que

$$\tilde{\mathbf{R}}_k^{ij} = \begin{pmatrix} R_{k,ii} & R_{k,ij} \\ R_{k,ji} & R_{k,jj} \end{pmatrix}. \quad (3.57)$$

Le critère $\mathcal{J}_2(\mathbf{B}^{ij})$ est donc un critère mesurant la diagonalité du sous problème 2×2

$$\tilde{\mathbf{R}}_k^{ij} = \tilde{\mathbf{B}}^{ij} \tilde{\mathbf{R}}_k^{ij} \tilde{\mathbf{B}}^{ij\dagger} \quad (3.58)$$

où

$$\tilde{\mathbf{R}}_k^{nj} = \begin{pmatrix} R'_{k,ii} & R'_{k,ij} \\ R'_{k,ji} & R'_{k,jj} \end{pmatrix}. \quad (3.59)$$

Afin d'estimer les paramètres des matrices élémentaires \mathbf{B}^{ij} , nous proposons deux méthodes différentes. La première, dans le cadre de l'approche classique, est une méthode itérative de type gradient, la seconde, dans le cadre de l'approche adaptée, est une méthode entièrement analytique.

3.3.1 Approche classique

Ici, nous proposons d'optimiser le critère (3.51) à l'aide d'une méthode du gradient.

Calcul générique du gradient : Pour une paire d'indices $(i, j) \in \mathcal{N}_s^2$, $i < j$ donnée, le but est de calculer itérativement

$$\mathbf{r}_{ij} = \begin{pmatrix} u^{ij} \\ \ell^{ij} \end{pmatrix} \quad (3.60)$$

tel que

$$\mathbf{r}_{ij}^{(n+1)} = \mathbf{r}_{ij}^{(n)} - \mu \nabla_{\mathbf{r}_{ij}}^{(\dagger)} \quad (3.61)$$

où $\mu \in \mathbb{R}$ est le pas du gradient et $(\cdot)^{(n)}$ représente la variable passée en paramètre à la n -ième itération. $\nabla_{\mathbf{r}_{ij}}^{(\dagger)}$ est le gradient de $\mathcal{J}(\mathbf{B}^{ij})$ défini de la manière suivante

$$\nabla_{\mathbf{r}_{ij}}^{(\dagger)} = \begin{pmatrix} \nabla_{u^{ij}}^{(\dagger)} \\ \nabla_{\ell^{ij}}^{(\dagger)} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathcal{J}^{(\dagger)}(\mathbf{B}^{ij})}{\partial u^{ij*}} \\ \frac{\partial \mathcal{J}^{(\dagger)}(\mathbf{B}^{ij})}{\partial \ell^{ij*}} \end{pmatrix} = \frac{\partial \mathcal{J}^{(\dagger)}(\mathbf{B}^{ij})}{\partial \mathbf{r}_{ij}^*}. \quad (3.62)$$

La notation $(\cdot)^{(\dagger)}$ signifie que l'expression de la variable passée en paramètre est générique. Elle prendra une forme différente selon que l'on traite le modèle complexe symétrique (pour lequel on utilisera la notation $(\cdot)^{(T)}$) ou le modèle hermitien (pour lequel on utilisera la notation $(\cdot)^{(H)}$). A partir de la relation (3.51), le gradient se lit

$$\nabla_{\mathbf{r}_{ij}}^{(\dagger)} = \nabla_{2, \mathbf{r}_{ij}}^{(\dagger)} + \nabla'_{\mathbf{r}_{ij}}^{(\dagger)} = \frac{\partial \mathcal{J}_2^{(\dagger)}(\mathbf{B}^{ij})}{\partial \mathbf{r}_{ij}^*} + \frac{\partial \mathcal{J}'^{(\dagger)}(\mathbf{B}^{ij})}{\partial \mathbf{r}_{ij}^*}. \quad (3.63)$$

Pour tout $k \in \mathcal{K}$ et pour tout $q \in \mathcal{Q}$, il existe trois fonctions de \mathbf{r}_{ij} , notées $f_k^{(\dagger)}(\mathbf{r}_{ij})$, $g_{kq}^{(\dagger)}(\mathbf{r}_{ij})$ et $h_{kq}^{(\dagger)}(\mathbf{r}_{ij})$ telles qu'on puisse réécrire (3.52) et (3.53) comme

$$\mathcal{J}_2^{(\dagger)}(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K |f_k^{(\dagger)}(\mathbf{r}_{ij})|^2 \quad (3.64)$$

$$\mathcal{J}'^{(\dagger)}(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K \sum_{q \in \mathcal{Q}} \left(|g_{kq}^{(\dagger)}(\mathbf{r}_{ij})|^2 + |h_{kq}^{(\dagger)}(\mathbf{r}_{ij})|^2 \right). \quad (3.65)$$

Finalement les dérivées de (3.64) et (3.65) s'écrivent

$$\nabla_{2, \mathbf{r}_{ij}}^{(\dagger)} = 2 \sum_{k=1}^K \left(\frac{\partial f_k^{(\dagger)}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} f_k^{(\dagger)*}(\mathbf{r}_{ij}) + f_k^{(\dagger)}(\mathbf{r}_{ij}) \frac{\partial f_k^{(\dagger)*}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} \right) \quad (3.66)$$

$$\begin{aligned} \nabla_{\mathbf{r}_{ij}}^{(\dagger)} = 2 \sum_{k=1}^K \sum_{q \in \mathcal{Q}} & \left(\frac{\partial g_{kq}^{(\dagger)}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} g_{kq}^{(\dagger)*}(\mathbf{r}_{ij}) + g_{kq}^{(\dagger)}(\mathbf{r}_{ij}) \frac{\partial g_{kq}^{(\dagger)*}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} \right. \\ & \left. + \frac{\partial h_{kq}^{(\dagger)}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} h_{kq}^{(\dagger)*}(\mathbf{r}_{ij}) + h_{kq}^{(\dagger)}(\mathbf{r}_{ij}) \frac{\partial h_{kq}^{(\dagger)*}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} \right). \end{aligned} \quad (3.67)$$

Les calculs spécifiques liés aux modèles complexe symétrique et hermitien seront développés dans les paragraphes suivants.

Calcul générique du pas optimal : Afin d'améliorer la vitesse de convergence de l'algorithme, nous proposons de calculer la valeur optimale du pas μ . A une étape n fixée, après calcul du gradient (3.63), on a la solution $\mathbf{r}'_{ij} = \mathbf{r}_{ij}^{(n+1)}$ (3.61), que l'on peut aussi écrire sous la forme vectorielle

$$\mathbf{r}'_{ij} = \begin{pmatrix} u^{ij} \\ \ell^{ij} \end{pmatrix} = \begin{pmatrix} u^{ij} - \mu \nabla_{u^{ij}}^{(\dagger)} \\ \ell^{ij} - \mu \nabla_{\ell^{ij}}^{(\dagger)} \end{pmatrix}. \quad (3.68)$$

On connaît u^{ij} et ℓ^{ij} à l'étape courante n , on vient de calculer le gradient $\nabla_{\mathbf{r}_{ij}}^{(\dagger)}$, il ne nous reste qu'à estimer la valeur optimale du pas μ . Soit \mathbf{B}'^{ij} la matrice élémentaire associée \mathbf{r}'_{ij} caractérisée par sa partie utile

$$\tilde{\mathbf{B}}'^{ij} = \begin{pmatrix} 1 & u'^{ij} \\ \ell'^{ij} & 1 + \ell'^{ij} u'^{ij} \end{pmatrix}, \quad (3.69)$$

l'estimation de μ se fait par l'optimisation du critère $\mathcal{J}^{(\dagger)}(\mathbf{B}'^{ij})$ de telle sorte que

$$\hat{\mu} = \arg \min_{\mathbf{B}'^{ij}} \mathcal{J}^{(\dagger)}(\mathbf{B}'^{ij}), \quad (3.70)$$

où selon (3.51)

$$\mathcal{J}^{(\dagger)}(\mathbf{B}'^{ij}) = \mathcal{J}_2^{(\dagger)}(\mathbf{B}'^{ij}) + \mathcal{J}'^{(\dagger)}(\mathbf{B}'^{ij}) + C_{cc}, \quad (3.71)$$

avec $\mathcal{J}_2^{(\dagger)}(\mathbf{B}'^{ij})$ et $\mathcal{J}'^{(\dagger)}(\mathbf{B}'^{ij})$ égaux, respectivement, à (3.64) et (3.65) en remplaçant \mathbf{r}_{ij} par \mathbf{r}'_{ij} . On va donc calculer la racine de

$$\frac{\partial \mathcal{J}^{(\dagger)}(\mathbf{B}'^{ij})}{\partial \mu} = \frac{\partial \mathcal{J}_2^{(\dagger)}(\mathbf{B}'^{ij})}{\partial \mu} + \frac{\partial \mathcal{J}'^{(\dagger)}(\mathbf{B}'^{ij})}{\partial \mu} \quad (3.72)$$

minimisant (3.71). Comme $\mu \in \mathbb{R}$, alors la propriété $\left(\frac{\partial(\cdot)}{\partial \mu}\right)^* = \frac{\partial(\cdot)^*}{\partial \mu}$ est vérifiée. Finalement les dérivées de $\mathcal{J}_2^{(\dagger)}(\mathbf{B}'^{ij})$ et $\mathcal{J}'^{(\dagger)}(\mathbf{B}'^{ij})$ par rapport à μ s'écrivent sous la forme générique

$$\frac{\partial \mathcal{J}_2^{(\dagger)}(\mathbf{B}'^{ij})}{\partial \mu} = 4\Re \left[\sum_{k=1}^K \left(\frac{\partial f_k^{(\dagger)}(\mathbf{r}'_{ij})}{\partial \mu} f_k^{(\dagger)*}(\mathbf{r}'_{ij}) \right) \right] \quad (3.73)$$

$$\frac{\partial \mathcal{J}'^{(\dagger)}(\mathbf{B}'^{ij})}{\partial \mu} = 4\Re \left[\sum_{k=1}^K \sum_{q \in \mathcal{Q}} \left(g_{kq}^{(\dagger)}(\mathbf{r}'_{ij}) \frac{\partial g_{kq}^{(\dagger)*}(\mathbf{r}'_{ij})}{\partial \mu} + h_{kq}^{(\dagger)}(\mathbf{r}'_{ij}) \frac{\partial h_{kq}^{(\dagger)*}(\mathbf{r}'_{ij})}{\partial \mu} \right) \right]. \quad (3.74)$$

Nous allons maintenant développer les calculs du gradient et du pas optimal, tout d'abord en considérant le modèle complexe symétrique, puis le modèle hermitien.

3.3.1.1 Modèle complexe symétrique

On considère K matrices cibles complexes $\mathbf{R}_k^{(s)}$ de dimensions $N_s \times N_s$ suivant le modèle (2.12) que l'on cherche à diagonaliser conjointement. Etant donné que ces matrices sont symétriques, la propriété (3.18) s'applique à elles.

Calcul du gradient : Pour le calcul du gradient, noté ici $\nabla_{\mathbf{r}'_{ij}}^{(T)}$ (3.63), qui se lit comme la somme de $\nabla_{2, \mathbf{r}'_{ij}}^{(T)}$ (3.66) et de $\nabla_{\mathbf{r}'_{ij}}'^{(T)}$ (3.67), il suffit de trouver les expressions des fonctions $f_k^{(T)}$, $g_{kq}^{(T)}$ et $h_{kq}^{(T)}$. En égalant (3.52) et (3.64), on obtient $f_k^{(T)}$, on en déduit aisément l'expression

de son conjugué $f_k^{(T)*}$ et enfin, on calcule les gradients de ces deux fonctions

$$\begin{aligned}
f_k^{(T)}(\mathbf{r}_{ij}) &= \ell^{ij} R_{k,ii}^{(s)} + (1 + 2\ell^{ij} u^{ij}) R_{k,ij}^{(s)} + u^{ij} (1 + \ell^{ij} u^{ij}) R_{k,jj}^{(s)} \\
\frac{\partial f_k^{(T)}(\mathbf{r})}{\partial \mathbf{r}_{ij}^*} &= \begin{pmatrix} \frac{\partial f_k^{(T)}(\mathbf{r}_{ij})}{\partial u^{ij*}} \\ \frac{\partial f_k^{(T)}(\mathbf{r}_{ij})}{\partial \ell^{ij*}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
f_k^{(T)*}(\mathbf{r}_{ij}) &= \ell^{ij*} R_{k,ii}^{(s)*} + (1 + 2\ell^{ij*} u^{ij*}) R_{k,ij}^{(s)*} + u^{ij*} (1 + \ell^{ij*} u^{ij*}) R_{k,jj}^{(s)*} \\
\frac{\partial f_k^{(T)*}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} &= \begin{pmatrix} 2\ell^{ij*} R_{k,ij}^{(s)*} + (1 + 2\ell^{ij*} u^{ij*}) R_{k,jj}^{(s)*} \\ R_{k,ii}^{(s)*} + 2u^{ij*} R_{k,ij}^{(s)*} + u^{ij*2} R_{k,jj}^{(s)*} \end{pmatrix}. \tag{3.75}
\end{aligned}$$

On fait de même pour $g_{kq}^{(T)}$ et $h_{kq}^{(T)}$ en égalant (3.53) et (3.65)

$$\begin{aligned}
g_{kq}^{(T)}(\mathbf{r}_{ij}) &= \ell^{ij} R_{k,iq}^{(s)} + (1 + \ell^{ij} u^{ij}) R_{k,jq}^{(s)} \\
\frac{\partial g_{kq}^{(T)}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} &= \mathbf{0} \\
g_{kq}^{(T)*}(\mathbf{r}_{ij}) &= \ell^{ij*} R_{k,iq}^{(s)*} + (1 + \ell^{ij*} u^{ij*}) R_{k,jq}^{(s)*} \\
\frac{\partial g_{kq}^{(T)*}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} &= \begin{pmatrix} \ell^{ij*} R_{k,jq}^{(s)*} \\ R_{k,iq}^{(s)*} + u^{ij*} R_{k,jq}^{(s)*} \end{pmatrix} \tag{3.76}
\end{aligned}$$

$$\begin{aligned}
h_{kq}^{(T)}(\mathbf{r}_{ij}) &= R_{k,iq}^{(s)} + u^{ij} R_{k,jq}^{(s)} \\
\frac{\partial h_{kq}^{(T)}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} &= \mathbf{0} \\
h_{kq}^{(T)*}(\mathbf{r}_{ij}) &= R_{k,iq}^{(s)*} + u^{ij*} R_{k,jq}^{(s)*} \\
\frac{\partial h_{kq}^{(T)*}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} &= \begin{pmatrix} R_{k,jq}^{(s)*} \\ 0 \end{pmatrix} \tag{3.77}
\end{aligned}$$

où $\mathbf{0}$ est le vecteur nul, ici de dimension 2×1 . Finalement, en injectant (3.75) dans (3.66) d'une part et (3.76) et (3.77) dans (3.67) d'autre part, on est en mesure de calculer le gradient $\nabla_{\mathbf{r}_{ij}}^{(T)}$.

Estimation du pas optimal : Pour déterminer le pas optimal μ , on doit minimiser le critère $\mathcal{J}^{(T)}(\mathbf{B}'^{ij})$ (3.71). Pour cela, on propose de chercher les racines de sa dérivée (3.72). Procédons par étape, calculons d'abord $\frac{\partial \mathcal{J}_2^{(T)}(\mathbf{B}'^{ij})}{\partial \mu}$ (3.73). Pour cela, il suffit de réécrire $f_k^{(T)}(\mathbf{r}_{ij})$ (3.75), mais en remplaçant \mathbf{r}_{ij} par $\mathbf{r}'_{ij} = \mathbf{r}_{ij} - \mu \nabla_{\mathbf{r}_{ij}}^{(\dagger)}$. Après quelques calculs évidents, on obtient $f_k^{(T)}(\mathbf{r}'_{ij})$ et la dérivée de son conjugué par rapport à μ

$$\begin{aligned} f_k^{(T)}(\mathbf{r}'_{ij}) &= \delta_f^{(T)} \mu^3 + \gamma_f^{(T)} \mu^2 + \beta_f^{(T)} \mu + \alpha_f^{(T)} \\ \frac{\partial f_k^{(T)*}(\mathbf{r}'_{ij})}{\partial \mu} &= 3\delta_f^{(T)*} \mu^2 + 2\gamma_f^{(T)*} \mu + \beta_f^{(T)*}, \end{aligned} \quad (3.78)$$

avec

$$\begin{aligned} \alpha_f^{(T)} &= \ell^{ij} R_{k,ii}^{(s)} + (1 + 2\ell^{ij} u^{ij}) R_{k,ij}^{(s)} + u^{ij} (1 + \ell^{ij} u^{ij}) R_{k,jj}^{(s)} \\ \beta_f^{(T)} &= - \left(\nabla_{\ell^{ij}}^{(T)} R_{k,ii}^{(s)} + 2 \left(\ell^{ij} \nabla_{u^{ij}}^{(T)} + u \nabla_{\ell^{ij}}^{(T)} \right) R_{k,ij}^{(s)} + \left((1 + 2\ell^{ij} u^{ij}) \nabla_{\ell^{ij}}^{(T)} + u^{ij} 2 \nabla_{\ell^{ij}}^{(T)} \right) R_{k,jj}^{(s)} \right) \\ \gamma_f^{(T)} &= 2 \nabla_{\ell^{ij}}^{(T)} \nabla_{u^{ij}}^{(T)} R_{k,ij}^{(s)} + \left(2u^{ij} \nabla_{\ell^{ij}}^{(T)} \nabla_{u^{ij}}^{(T)} + \ell^{ij} \nabla_{u^{ij}}^{(T)2} \right) R_{k,jj}^{(s)} \\ \delta_f^{(T)} &= - \nabla_{\ell^{ij}}^{(T)} \nabla_{u^{ij}}^{(T)2} R_{k,jj}^{(s)} \end{aligned} \quad (3.79)$$

On fait de même pour $g_{kq}^{(T)}(\mathbf{r}'_{ij})$ à partir de (3.76)

$$\begin{aligned} g_{kq}^{(T)}(\mathbf{r}'_{ij}) &= \gamma_g^{(T)} \mu^2 + \beta_g^{(T)} \mu + \alpha_g^{(T)} \\ \frac{\partial g_{kq}^{(T)*}(\mathbf{r}'_{ij})}{\partial \mu} &= 2\gamma_g^{(T)*} \mu + \beta_g^{(T)*}, \end{aligned} \quad (3.80)$$

où

$$\begin{aligned} \alpha_g^{(T)} &= \ell^{ij} R_{k,iq}^{(s)} + (1 + \ell^{ij} u^{ij}) R_{k,jq}^{(s)} \\ \beta_g^{(T)} &= - \left(\nabla_{\ell^{ij}}^{(T)} R_{k,iq}^{(s)} + \left(\ell^{ij} \nabla_{u^{ij}}^{(T)} + u^{ij} \nabla_{\ell^{ij}}^{(T)} \right) R_{k,jq}^{(s)} \right) \\ \gamma_g^{(T)} &= \nabla_{\ell^{ij}}^{(T)} \nabla_{u^{ij}}^{(T)} R_{k,jq}^{(s)} \end{aligned} \quad (3.81)$$

et $h_{kq}^{(T)}(\mathbf{r}'_{ij})$ à partir de (3.77)

$$\begin{aligned} h_{kq}^{(T)}(\mathbf{r}'_{ij}) &= \beta_h^{(T)} \mu + \alpha_h^{(T)} \\ \frac{\partial h_{kq}^{(T)*}(\mathbf{r}'_{ij})}{\partial \mu} &= \beta_h^{(T)*}, \end{aligned} \quad (3.82)$$

où

$$\begin{aligned} \alpha_h^{(T)} &= R_{k,iq}^{(s)} + u^{ij} R_{k,jq}^{(s)} \\ \beta_h^{(T)} &= -\nabla_{u^{ij}}^{(T)} R_{k,jq}^{(s)}. \end{aligned} \quad (3.83)$$

Avec (3.78), (3.80) et (3.82), on peut remonter à $\frac{\partial \mathcal{J}^{(T)}(\mathbf{B}'^{ij})}{\partial \mu}$ (3.72) qui s'écrit sous la forme polynomiale suivante

$$\frac{\partial \mathcal{J}^{(T)}(\mathbf{B}'^{ij})}{\partial \mu} = 4\Re \left[\sum_{k=1}^K F^{(T)} \mu^5 + E^{(T)} \mu^4 + D^{(T)} \mu^3 + C^{(T)} \mu^2 + B^{(T)} \mu + A^{(T)} \right] \quad (3.84)$$

avec

$$\begin{aligned} A^{(T)} &= \alpha_f^{(T)} \beta_f^{(T)*} + \sum_{q \in \mathcal{Q}} \left(\alpha_g^{(T)} \beta_g^{(T)*} + \alpha_h^{(T)} \beta_h^{(T)*} \right) \\ B^{(T)} &= 2\alpha_f^{(T)} \gamma_f^{(T)*} + |\beta_f^{(T)}|^2 + \sum_{q \in \mathcal{Q}} \left(2\alpha_g^{(T)} \gamma_g^{(T)*} + |\beta_g^{(T)}|^2 + |\beta_h^{(T)}|^2 \right) \\ C^{(T)} &= 3\alpha_f^{(T)} \delta_f^{(T)*} + 2\beta_f^{(T)} \gamma_f^{(T)*} + \gamma_f^{(T)} \beta_f^{(T)*} + \sum_{q \in \mathcal{Q}} \left(2\beta_g^{(T)} \gamma_g^{(T)*} + \gamma_g^{(T)} \beta_g^{(T)*} \right) \\ D^{(T)} &= 3\beta_f^{(T)} \delta_f^{(T)*} + 2|\gamma_f^{(T)}|^2 + \delta_f^{(T)} \beta_f^{(T)*} + 2 \sum_{q \in \mathcal{Q}} |\gamma_g^{(T)}|^2 \\ E^{(T)} &= 3\gamma_f^{(T)} \delta_f^{(T)*} + 2\delta_f^{(T)} \gamma_f^{(T)*} \\ F^{(T)} &= 3|\delta_f^{(T)}|^2 \end{aligned} \quad (3.85)$$

Il ne reste plus qu'à calculer les racines de (3.84) et à choisir celle minimisant (3.71). Comme (3.84) est un polynôme de degré cinq, il ne permet pas une résolution analytique. Cependant, en supposant que l'on se trouve proche d'une solution diagonalisante, nous avons les propriétés suivantes, pour tout $a \neq b$, $|R_{k,ab}^{(s)}| \ll 1$, $|\mathbf{r}_{ij}| \ll \mathbf{1}$ (où $\mathbf{1}$ est le vecteur unitaire, ici de dimension 2×1), $|\nabla_{\mathbf{r}_{ij}}^{(T)}| \ll \mathbf{1}$ et $\mu \ll 1$. Sous cette hypothèse, $F^{(T)}$ et $E^{(T)}$ peuvent être

négligés devant tous les autres coefficients et (3.84) peut être approximée par

$$\frac{\partial J^{(T)}(\mathbf{B}')}{\partial \mu} = 4\Re \left[\sum_{k=1}^K D^{(T)} \mu^3 + C^{(T)} \mu^2 + B^{(T)} \mu + A^{(T)} \right]. \quad (3.86)$$

Comme (3.86) est un polynôme de degré trois, la détermination des paramètres ℓ^{ij} et u^{ij} par cette méthode de type gradient avec calcul d'un pas optimal se fait maintenant de manière entièrement analytique.

3.3.1.2 Modèle hermitien

Calculons maintenant le gradient et estimons le pas optimal, pour la diagonalisation conjointe d'un ensemble de K matrices $\mathbf{R}_k^{(h)}$ de dimensions $N_s \times N_s$ respectant le modèle (2.16). Si $\mathbf{D}_k \in \mathbb{C}^{N_s \times N_s}$, alors on ne s'intéressera qu'à la partie hermitienne des $\mathbf{R}_k^{(h)}$ (2.17).

Calcul du gradient : Comme pour le modèle complexe symétrique, pour le calcul du gradient, noté ici $\nabla_{\mathbf{r}_{ij}}^{(H)} = \nabla_{2, \mathbf{r}_{ij}}^{(H)} + \nabla_{\mathbf{r}_{ij}}^{(H)}$ (3.63), on va exprimer $f_k^{(H)}$, $g_{kq}^{(H)}$ et $h_{kq}^{(H)}$ en fonction de ℓ^{ij} et u^{ij} . $f_k^{(H)}$, son conjugué et leurs gradients respectifs sont déduits en égalant (3.52) et (3.64)

$$\begin{aligned} f_k^{(H)}(\mathbf{r}_{ij}) &= \ell^{ij*} R_{k,ii}^{(h)} + (1 + \ell^{ij*} u^{ij*}) R_{k,ij}^{(h)} + \ell^{ij*} u^{ij} R_{k,ij}^{(h)*} + u^{ij} (1 + \ell^{ij*} u^{ij*}) R_{k,jj}^{(h)} \\ \frac{f_k^{(H)}(\mathbf{r})}{\partial \mathbf{r}_{ij}^*} &= \begin{pmatrix} \ell^{ij*} R_{k,ij}^{(h)} + \ell^{ij*} u^{ij} R_{k,jj}^{(h)} \\ R_{k,ii}^{(h)} + 2\Re(u^{ij*} R_{k,ij}^{(h)}) + |u^{ij}|^2 R_{k,jj}^{(h)} \end{pmatrix} \\ f_k^{(H)*}(\mathbf{r}_{ij}) &= \ell^{ij} R_{k,ii}^{(h)} + (1 + \ell^{ij} u^{ij}) R_{k,ij}^{(h)*} + \ell^{ij} u^{ij*} R_{k,ij}^{(h)} + u^{ij*} (1 + \ell^{ij} u^{ij}) R_{k,jj}^{(h)} \\ \frac{f_k^{(H)*}(\mathbf{r}_{ij})}{\partial \mathbf{r}_{ij}^*} &= \begin{pmatrix} \ell^{ij} R_{k,ij}^{(h)} + (1 + \ell^{ij} u^{ij}) R_{k,jj}^{(h)} \\ 0 \end{pmatrix}. \end{aligned} \quad (3.87)$$

Quant aux autres fonctions, on a directement $g_{kq}^{(H)} = g_{kq}^{(T)}$ (3.76) et $h_{kq}^{(H)} = h_{kq}^{(T)}$ (3.77). Finalement, en injectant (3.87) dans (3.66) d'une part et (3.76) et (3.77) dans (3.67) d'une part, on obtient l'expression du gradient $\nabla_{\mathbf{r}_{ij}}^{(H)}$.

Estimation du pas optimal : Là encore, comme pour le cas complexe symétrique, la détermination du pas optimal μ se fait par la minimisation du critère $\mathcal{J}^{(H)}(\mathbf{B}'^{ij})$ (3.71). On calcule donc sa dérivée (3.72). Commençons par le terme $\frac{\partial \mathcal{J}_2^{(H)}(\mathbf{B}'^{ij})}{\partial \mu}$ (3.73), qui dépend de la fonction $f_k^{(H)}(\mathbf{r}'_{ij})$. En s'appuyant sur (3.87) et en remplaçant \mathbf{r}_{ij} par $\mathbf{r}'_{ij} = \mathbf{r}_{ij} - \mu \nabla_{\mathbf{r}_{ij}}^{(H)}$, on

déduit les expressions

$$\begin{aligned} f_k^{(H)}(\mathbf{r}'_{ij}) &= \delta_f^{(H)} \mu^3 + \gamma_f^{(H)} \mu^2 + \beta_f^{(H)} \mu + \alpha_f^{(H)} \\ \frac{\partial f_k^{(H)*}(\mathbf{r}'_{ij})}{\partial \mu} &= 3\delta_f^{(H)*} \mu^2 + 2\gamma_f^{(H)*} \mu + \beta_f^{(H)*}, \end{aligned} \quad (3.88)$$

dont les coefficients valent

$$\begin{aligned} \alpha_f^{(H)} &= \ell^{ij} R_{k,ii}^{(h)} + \ell^{ij} u^{ij*} R_{k,ij}^{(h)} + (1 + \ell^{ij} u^{ij}) R_{k,ij}^{(h)*} + u^{ij*} (1 + \ell^{ij} u^{ij}) R_{k,jj}^{(s)} \\ \beta_f^{(H)} &= - \left(2 \left(\ell^{ij} \Re \left(\nabla_{u^{ij}}^{(H)*} R_{k,ij}^{(h)} \right) + \nabla_{\ell^{ij}}^{(H)} \Re \left(u^{ij*} R_{k,ij}^{(h)} \right) \right) \right. \\ &\quad \left. + \left(u^{ij*} \left(\ell^{ij} \nabla_{u^{ij}}^{(H)} + u^{ij} \nabla_{\ell^{ij}}^{(H)} \right) + (1 + \ell^{ij} u^{ij}) \nabla_{u^{ij}}^{(H)*} \right) R_{k,jj}^{(h)} + \nabla_{\ell^{ij}}^{(H)} R_{k,ii}^{(h)} \right) \\ \gamma_f^{(H)} &= 2 \nabla_{\ell^{ij}}^{(H)} \Re \left(\nabla_{u^{ij}}^{(H)*} R_{k,ij}^{(h)} \right) + \left(2 \nabla_{\ell^{ij}}^{(H)} \Re \left(u^{ij*} \nabla_{u^{ij}}^{(H)} \right) + \ell^{ij} \left| \nabla_{u^{ij}}^{(H)} \right|^2 \right) R_{k,jj}^{(h)} \\ \delta_f^{(H)} &= - \nabla_{\ell^{ij}}^{(H)} \left| \nabla_{u^{ij}}^{(H)} \right|^2 R_{k,jj}^{(h)}. \end{aligned} \quad (3.89)$$

On constate aussi que $g_{kq}^{(H)}(\mathbf{r}'_{ij}) = g_{kq}^{(T)}(\mathbf{r}'_{ij})$ (3.80) et $h_{kq}^{(H)}(\mathbf{r}'_{ij}) = h_{kq}^{(T)}(\mathbf{r}'_{ij})$ (3.82).

Avec $f_k^{(H)}$, $g_{kq}^{(H)}$ et $h_{kq}^{(H)}$, on remonte à $\frac{\partial \mathcal{J}^{(H)}(\mathbf{B}'^{ij})}{\partial \mu}$ (3.72) qui s'écrit sous la forme du polynôme d'ordre cinq (3.84) dont les coefficients valent (3.85), en remplaçant toutefois les indices $(\cdot)^{(T)}$ par $(\cdot)^{(H)}$. Tout comme pour le modèle complexe symétrique, il est possible d'approximer le polynôme d'ordre cinq précédent par le polynôme d'ordre trois (3.86), si on suppose être proche d'une solution diagonalisante. Ceci permet une résolution entièrement analytique du problème. Finalement, le pas optimal μ est obtenu en calculant les racines soit du polynôme complet, soit du polynôme approximé et en retenant celle minimisant (3.71).

3.3.2 Approche adaptée

Dans ce paragraphe, nous proposons une méthode de détermination des matrices élémentaires basée sur une recherche de vecteur propre. On a constaté que la transformation (3.50) modifie les composantes hors-diagonales d'une matrice \mathbf{R}_k donnée de façon spécifique. Ainsi, comme le montre la figure 3.3, les éléments d'indices (i, j) et (j, i) subissent une transformation quadratique, les $4(N_s - 2)$ autres éléments hors-diagonaux situés sur les lignes et colonnes i et j subissent une transformation linéaire et les autres éléments hors-diagonaux restent constants. L'influence des paramètres ℓ^{ij} et u^{ij} est plus importante sur les composantes ayant subi une transformation quadratique que sur celles impactées par une transformation linéaire. Afin de diagonaliser conjointement l'ensemble des matrices \mathbf{R}_k , on propose donc de

se concentrer sur l'optimisation du critère $\mathcal{J}_2^{(\dagger)}$ (3.52). On se concentre alors uniquement sur la minimisation des termes les plus pondérés par les paramètres à estimer. Ceci semble être une bonne stratégie pour réduire la complexité du problème et, potentiellement, améliorer la vitesse de convergence par rapport à l'approche classique.

Dans les deux paragraphes suivants, nous montrerons que le fait de se concentrer uniquement sur la transformation quadratique de (3.50) se justifie dans le cadre de l'approche adaptée, aussi bien pour la diagonalisation conjointe de matrices complexes symétriques que pour celle de matrices hermitiennes. Nous détaillerons, ensuite, la méthode d'optimisation développée pour chacun de ces ensembles de matrices.

On rappelle que l'approche adaptée, repose sur l'hypothèse que l'on se trouve proche d'une solution diagonalisante. Pour la stratégie d'estimation couplée des paramètres, cela signifie que pour tout $k \in \mathcal{K}$ et pour tout $(a, b) \in \mathcal{N}_s^2$ avec $a \neq b$,

$$|R_{k,ab}^{(\dagger)}| \ll 1, \quad |u^{ij}| \ll 1 \quad \text{et} \quad |\ell^{ij}| \ll 1. \quad (3.90)$$

3.3.2.1 Modèle complexe symétrique

On considère K matrices cibles complexes symétriques $\mathbf{R}_k^{(s)} \in \mathbb{C}^{N_s \times N_s}$ suivant le modèle (2.12). Pour diagonaliser conjointement cet ensemble, on cherche à optimiser le critère (3.51) dont les termes non constants (3.52) et (3.53) s'écrivent pour des matrices complexes symétriques

$$\mathcal{J}_2^{(T)}(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K | \ell^{ij} R_{k,ii}^{(s)} + (1 + 2\ell^{ij} u^{ij}) R_{k,ij}^{(s)} + u^{ij} (1 + \ell^{ij} u^{ij}) R_{k,jj}^{(s)} |^2 \quad (3.91)$$

$$\mathcal{J}'^{(T)}(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K \sum_{q \in \mathcal{Q}} \left(| \ell^{ij} R_{k,iq}^{(s)} + (1 + \ell^{ij} u^{ij}) R_{k,jq}^{(s)} |^2 + | R_{k,iq}^{(s)} + u^{ij} R_{k,jq}^{(s)} |^2 \right). \quad (3.92)$$

Supposons, maintenant, que l'on soit proche d'une solution diagonalisante, (3.90) permet d'approximer (3.92) de telle sorte que

$$\mathcal{J}'^{(T)}(\mathbf{B}^{ij}) \approx 2 \sum_{k=1}^K \sum_{q \in \mathcal{Q}} \left(| R_{k,jq}^{(s)} |^2 + | R_{k,iq}^{(s)} |^2 \right) = C_{\mathcal{J}'}. \quad (3.93)$$

Ainsi $\mathcal{J}'^{(T)}$ est approximativement constant, ce qui n'est bien entendu pas le cas pour $\mathcal{J}_2^{(T)}$. Dans le cadre de l'approche adaptée, on peut donc schématiser l'impact de la transformation (3.50) sur une matrice \mathbf{R}'_k par la figure 3.4. Ceci nous amène à l'approximation suivante du

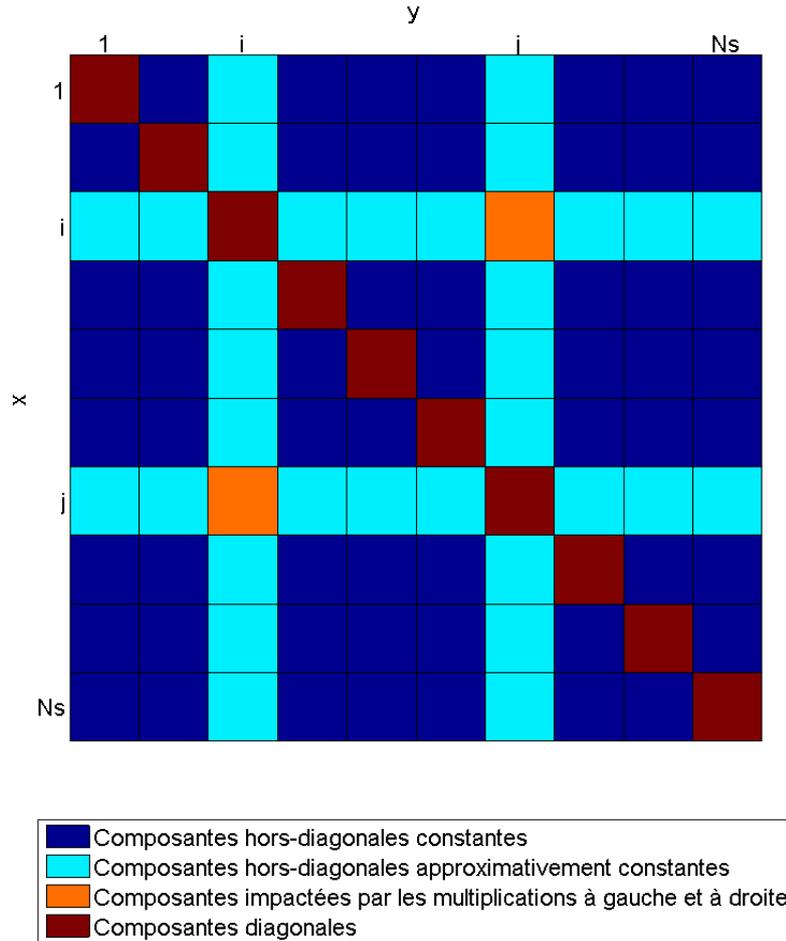


FIGURE 3.4 – Schéma de l'impact de la transformation (3.50) sur une matrice \mathbf{R}'_k lorsqu'on utilise l'approche adaptée.

critère (3.51)

$$\mathcal{J}^{(T)}(\mathbf{B}^{ij}) \approx \mathcal{J}_2^{(T)}(\mathbf{B}^{ij}) + (C_{\mathcal{J}'} + C_{cc}). \quad (3.94)$$

Finalement, sous l'hypothèse que les \mathbf{R}_k soient proches d'être des matrices diagonales, minimiser $\mathcal{J}_2^{(T)}$ (3.91) revient, approximativement, à minimiser $\mathcal{J}^{(T)}$ (3.51).

Détaillons maintenant la méthode analytique permettant l'estimation simultanée de ℓ^{ij} et u^{ij} par l'optimisation de $\mathcal{J}_2^{(T)}$. Commençons par réécrire (3.91) sous la forme quadratique suivante

$$\mathcal{J}_2^{(T)}(\mathbf{B}^{ij}) = 2\mathbf{p}_s^H \mathbf{O}_s \mathbf{p}_s \quad (3.95)$$

où

$$\mathbf{p}_s = \begin{pmatrix} \ell^{ij} \\ 1 + 2\ell^{ij}u^{ij} \\ u^{ij}(1 + \ell^{ij}u^{ij}) \end{pmatrix} \quad \text{et} \quad \mathbf{O}_s = \sum_{k=1}^K (\mathbf{o}_{s,k}^* \mathbf{o}_{s,k}^T). \quad (3.96)$$

avec

$$\mathbf{o}_{s,k}^T = \begin{pmatrix} R_{k,ii}^{(s)} & R_{k,ij}^{(s)} & R_{k,jj}^{(s)} \end{pmatrix}. \quad (3.97)$$

Notons que la matrice \mathbf{O}_s est définie non négative et que, sous les hypothèses (3.90), $\|\mathbf{p}_s\| \approx 1$. En se basant sur ces deux remarques, nous proposons la procédure suivante pour minimiser $\mathcal{J}_2^{(T)}$. On calcule le vecteur propre de norme un associé à la plus petite valeur propre de \mathbf{O}_s . Nous notons ce vecteur propre \mathbf{e}_s . Il est important de signaler que comme la matrice \mathbf{O}_s est de dimensions 3×3 , le calcul de ses valeurs propres et de ses vecteurs propres se fait analytiquement. On estime les paramètres de \mathbf{p}_s de telle sorte que

$$\mathbf{e}_s = \begin{pmatrix} e_{s_1} & e_{s_2} & e_{s_3} \end{pmatrix}^T = \alpha \mathbf{p}_s, \quad (3.98)$$

où α est un paramètre complexe. En développant (3.98), on obtient le système de trois équations à trois inconnues suivant

$$\begin{cases} e_{s_1} = \alpha \ell^{ij} \\ e_{s_2} = \alpha (1 + 2\ell^{ij}u^{ij}) \\ e_{s_3} = \alpha u^{ij} (1 + \ell^{ij}u^{ij}). \end{cases} \quad (3.99)$$

Les trois paramètres sont calculés en résolvant le système non-linéaire (3.99). Finalement, on obtient

$$\begin{cases} 0 = e_{s_1} u^{ij2} - e_{s_2} u^{ij} + e_{s_3} \\ \alpha = e_{s_2} - 2e_{s_1} u^{ij} \\ \ell^{ij} = \frac{e_{s_1}}{\alpha}. \end{cases} \quad (3.100)$$

Dans ce système, la première équation est un polynôme d'ordre deux en u^{ij} . Il y a donc deux solutions pour ce système. Néanmoins, nous savons que lorsque nous sommes proches d'une solution diagonalisante, les matrices \mathbf{R}_k sont mises à jour par des matrices élémentaires \mathbf{B}^{ij} tendant vers la matrice identité. Ainsi, lorsque l'on converge, ℓ^{ij} et u^{ij} doivent tendre vers zéro. Nous proposons donc de retenir la valeur de u^{ij} de plus petit module.

3.3.2.2 Modèle hermitien

On désire, maintenant, diagonaliser conjointement K matrices cibles hermitiennes $\mathbf{R}_k^{(h)} \in \mathbb{C}^{N_s \times N_s}$ respectant le modèle (2.16). On doit, pour cela, optimiser le critère (3.51) dont le terme (3.52) s'écrit pour des matrices hermitiennes

$$\mathcal{J}_2^{(H)}(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K \left| \ell^{ij*} R_{k,ii}^{(h)} + (1 + \ell^{ij*} u^{ij*}) R_{k,ij}^{(h)} + \ell^{ij*} u^{ij} R_{k,jj}^{(h)*} + u^{ij} (1 + \ell^{ij*} u^{ij*}) R_{k,jj}^{(h)} \right|^2 \quad (3.101)$$

et $\mathcal{J}'^{(H)}(\mathbf{B}^{ij}) = \mathcal{J}'^{(T)}(\mathbf{B}^{ij})$ (3.92). Ainsi, si on suppose que $\mathbf{R}_k^{(h)}$ sont proches d'être diagonales, on a $\mathcal{J}'^{(H)}(\mathbf{B}^{ij}) \approx C_{\mathcal{J}'}$ (3.93), ce qui implique l'approximation

$$\mathcal{J}^{(H)}(\mathbf{B}^{ij}) \approx \mathcal{J}_2^{(H)}(\mathbf{B}^{ij}) + (C_{\mathcal{J}'} + C_{cc}). \quad (3.102)$$

Finalement, comme pour le modèle complexe symétrique, sous l'hypothèse d'être proche d'une solution diagonalisante, l'estimation des matrices élémentaires se fera dans le but de minimiser le critère $\mathcal{J}_2^{(H)}$ (3.101).

Résolution directe : Appliquons la même méthode que dans le cas complexe symétrique. Réécrivons (3.101) sous la forme quadratique

$$\mathcal{J}_2^{(H)}(\mathbf{B}^{ij}) = 2\mathbf{p}_h^H \mathbf{O}_h \mathbf{p}_h \quad (3.103)$$

où

$$\mathbf{p}_h = \begin{pmatrix} \ell^{ij*} \\ 1 + \ell^{ij*} u^{ij*} \\ \ell^{ij*} u^{ij} \\ u^{ij} (1 + \ell^{ij*} u^{ij*}) \end{pmatrix} \quad \text{et} \quad \mathbf{O}_h = \sum_{k=1}^K (\mathbf{o}_{h,k}^* \mathbf{o}_{h,k}^T). \quad (3.104)$$

avec

$$\mathbf{o}_{h,k}^T = \begin{pmatrix} R_{k,ii}^{(h)} & R_{k,ij}^{(h)} & R_{k,ij}^{(h)*} & R_{k,jj}^{(h)} \end{pmatrix}. \quad (3.105)$$

Si l'on veut minimiser (3.103) en tentant une résolution analytique comme précédemment, l'idée serait d'identifier \mathbf{p}_h grâce au vecteur propre \mathbf{e}_h associé à la plus petite valeur propre de la matrice définie non négative \mathbf{O}_h et cela à un facteur complexe β près. Toutefois il est facile de montrer que le système d'équations surdéterminé $\mathbf{e}_h = \beta \mathbf{p}_h$ n'a pas de solution. Même si

on peut écrire ce système non-linéaire comme

$$\begin{cases} e_{h_1} = \beta \ell^{ij*} \\ e_{h_2} = \beta(1 + \ell^{ij*} u^{ij*}) \\ e_{h_3} = \beta \ell^{ij*} u^{ij} \\ e_{h_4} = \beta u^{ij}(1 + \ell^{ij*} u^{ij*}), \end{cases} \quad (3.106)$$

celui-ci implique l'égalité $\frac{e_{h_3}}{e_{h_1}} = \frac{e_{h_4}}{e_{h_2}}$. Cette contrainte est bien trop forte et n'a que très peu de chance d'être vérifiée. Nous devons trouver une approche différente pour éviter ce problème.

Résolution approchée : Comme nous avons fait l'hypothèse que les $\mathbf{R}_k^{(h)}$ sont proches de matrices diagonales, on a (3.90). De ce fait, le terme $\ell^{ij*} u^{ij} R_{k,jj}^{(h)*}$ dans (3.101) peut être négligé devant les trois autres termes. On considère donc le critère approximé, noté $\mathcal{J}_{2a}^{(H)}$, défini par

$$\mathcal{J}_{2a}^{(H)}(\mathbf{B}^{ij}) = 2 \sum_{k=1}^K | \ell^{ij*} R_{k,ii}^{(h)} + (1 + \ell^{ij*} u^{ij*}) R_{k,ij}^{(h)} + u^{ij} (1 + \ell^{ij*} u^{ij*}) R_{k,jj}^{(h)} |^2 \approx \mathcal{J}_2^{(H)}(\mathbf{B}^{ij}). \quad (3.107)$$

On écrit ce critère approximé, sous la forme quadratique suivante

$$\mathcal{J}_{2a}^{(H)}(\mathbf{B}^{ij}) = 2 \mathbf{p}_h'^H \mathbf{O}'_h \mathbf{p}'_h \quad (3.108)$$

où

$$\mathbf{p}'_h = \begin{pmatrix} \ell^{ij*} \\ 1 + \ell^{ij*} u^{ij*} \\ u^{ij} (1 + \ell^{ij*} u^{ij*}) \end{pmatrix} \quad \text{et} \quad \mathbf{O}'_h = \sum_{k=1}^K (\mathbf{o}'_{h,k} \mathbf{o}'_{h,k}{}^T). \quad (3.109)$$

avec

$$\mathbf{o}'_{h,k}{}^T = \begin{pmatrix} R_{k,ii}^{(h)} & R_{k,ij}^{(h)} & R_{k,jj}^{(h)} \end{pmatrix}. \quad (3.110)$$

Afin de minimiser (3.108), on suit la même procédure qu'au paragraphe 3.3.2.1. On calcule analytiquement le vecteur propre, dénoté \mathbf{e}'_h , associé à la plus petite valeur propre de la matrice définie non négative \mathbf{O}'_h de dimensions 3×3 . On obtient le système non-linéaire de trois équations suivant

$$\begin{cases} e'_{h_1} = \gamma \ell^{ij*} \\ e'_{h_2} = \gamma(1 + \ell^{ij*} u^{ij*}) \\ e'_{h_3} = \gamma u^{ij} (1 + \ell^{ij*} u^{ij*}), \end{cases} \quad (3.111)$$

où γ est un paramètre complexe. Les paramètres ℓ^{ij} et u^{ij} sont ensuite déterminés analytiquement, en résolvant ce système. On obtient finalement

$$\begin{cases} \hat{u}^{ij} = \frac{e'_{h_3}}{e'_{h_2}} \\ \alpha = e'_{h_2} + e'_{h_1} u^{ij*} \\ \hat{\ell}^{ij} = \left(\frac{e'_{h_1}}{\gamma} \right)^* . \end{cases} \quad (3.112)$$

3.3.2.3 Etape de normalisation

Afin d'améliorer la robustesse de l'algorithme, nous devons faire attention à ce que la norme de \mathbf{B}^{ij} n'augmente pas trop au cours des itérations. Le fait que les valeurs de ℓ^{ij} et u^{ij} résultent d'une division dans (3.100) et (3.112) peut induire des problèmes. C'est pourquoi, nous proposons de normaliser \mathbf{B}^{ij} en utilisant une multiplication à gauche par une matrice de pondération diagonale. Cette matrice, notée \mathbf{W} , ne diffère de la matrice identité de dimensions $N_s \times N_s$ que par ses composantes W_{ii} et W_{jj} , respectivement égales aux paramètres réels w et w^{-1} . Notons que comme $\det(\mathbf{W}) = 1$, alors la matrice $\mathbf{W}\mathbf{B}^{ij}$ respecte bien la contrainte $\det(\mathbf{W}\mathbf{B}^{ij}) = 1$. Le paramètre w est déterminé en minimisant la norme au carré de $\mathbf{W}\mathbf{B}^{ij}$ définie par

$$\|\mathbf{W}\mathbf{B}^{ij}\|^2 = (1 + |u^{ij}|^2)w^2 + (|\ell^{ij}|^2 + |1 + \ell^{ij}u^{ij}|^2)w^{-2} + N_s - 2. \quad (3.113)$$

Il ne reste plus qu'à dériver (3.113) par rapport à w

$$\frac{\partial \|\mathbf{W}\mathbf{B}^{ij}\|^2}{\partial w} = 2 \left((1 + |u|^2)w - (|\ell|^2 + |1 + \ell u|^2)w^{-3} \right). \quad (3.114)$$

En égalisant (3.114) à zéro, on obtient finalement

$$w = \frac{(|\ell|^2 + |1 + \ell u|^2)^{1/4}}{(1 + |u|^2)^{1/4}}. \quad (3.115)$$

3.3.2.4 Remarque

Dans ce paragraphe 3.3.2, on voudrait idéalement estimer les paramètres u^{ij} et ℓ^{ij} minimisant les formes quadratiques $\mathcal{J}_2^{(T)}$ et $\mathcal{J}_2^{(H)}$. Il semble que, directement, ce soit un problème difficile. Comme, sous l'hypothèse (3.90), la norme des vecteurs \mathbf{p}_s en (3.96) et \mathbf{p}'_h en (3.109) est proche de un, on va déterminer u^{ij} et ℓ^{ij} comme solutions des équations (3.99) et (3.111) basées sur un calcul préalable d'un vecteur propre associé à la plus petite valeur propre des

matrices définies non négatives \mathbf{O}_s et \mathbf{O}'_h . Cette procédure ne correspond pas à une minimisation directe de $\mathcal{J}_2^{(T)}$ et $\mathcal{J}_2^{(H)}$ mais elle s'est avérée très intéressante dans toutes nos simulations informatiques.

3.3.3 Description des algorithmes

Les pseudo-codes ci-dessous décrivent les algorithmes génériques résultant des deux approches proposées. L'algorithme 3, de type gradient, proposé dans le cadre de l'approche classique est dénommé G-CLUJA (pour *Gradient descent method for Coupled LU Jacobi-like Algorithm*), tandis l'algorithme 4, basé sur une recherche de vecteur propre et développé pour l'approche adaptée, est dénommé A-CLUJA (pour *Adapted approach for CLUJA*).

Algorithme 3 G-CLUJA

```

1: input :  $\{\mathbf{M}_k\} \subset \mathcal{M}^{(s)}$  (2.11) ou  $\{\mathbf{M}_k\} \subset \mathcal{M}^{(h)}$  (2.15)

2: Initialiser  $\mathbf{B} \leftarrow \mathbf{B}_o$  de dimensions  $N_s \times N_o$ 
3: Mettre à jour les matrices  $\mathbf{M}_k$  suivant (3.4)
4: repeat
5:   for  $1 \leq i < j \leq N_s$  do
6:     Initialiser  $\mathbf{r}_{ij}^{(0)}$  aléatoirement ou par toute autre meilleure solution
7:     for  $1 \leq n \leq N_{iter}$  do ▷ Méthode du gradient
8:       Calculer  $\nabla_{\mathbf{r}_{ij}}^{(\dagger)}$  en utilisant (3.76), (3.77) et (3.75) ou (3.87)
9:       Résoudre (3.84) ou (3.86) ▷ Calcul du pas optimal
10:      Prendre pour  $\mu$  la racine minimisant (3.71)
11:      Mettre à jour  $\mathbf{r}_{ij}^{(n+1)} = \mathbf{r}_{ij}^{(n)} - \mu \nabla_{\mathbf{r}_{ij}}^{(\dagger)}$ 
12:    end for
13:    Construire  $\mathbf{B}^{ij}$  à partir de  $\mathbf{r}_{ij}$ 
14:    Mettre à jour  $\mathbf{R}_k \leftarrow \mathbf{B}^{ij} \mathbf{R}_k \mathbf{B}^{ij \dagger}$ 
15:    Mettre à jour  $\mathbf{B} \leftarrow \mathbf{B}^{ij} \mathbf{B}$ 
16:  end for
17: until convergence

18: output :  $\mathbf{B}$ 

```

On propose enfin une combinaison des algorithmes G-CLUJA et A-CLUJA. En effet, l'initialisation de $\mathbf{r}_{ij}^{(0)}$ à la ligne 6 de G-CLUJA peut être remplacée par le calcul des paramètres ℓ^{ij} et u^{ij} de la ligne 6 de A-CLUJA. Nous dénommons cette méthode combinée C-CLUJA (pour *Combined method for CLUJA*).

Algorithme 4 A-CLUJA

-
- 1: **input** : $\{\mathbf{M}_k\} \subset \mathcal{M}^{(s)}$ (2.11) ou $\{\mathbf{M}_k\} \subset \mathcal{M}^{(h)}$ (2.15)
 - 2: Initialiser $\mathbf{B} \leftarrow \mathbf{B}_o$ de dimensions $N_s \times N_o$
 - 3: Mettre à jour les matrices \mathbf{M}_k suivant (3.4)
 - 4: **repeat**
 - 5: **for** $1 \leq i < j \leq N_s$ **do**
 - 6: Calculer ℓ^{ij} et u^{ij} en utilisant (3.100) ou (3.112)
 - 7: Construire \mathbf{B}^{ij}
 - 8: Calculer la matrice \mathbf{W} ▷ Etape de normalisation
 - 9: Mettre à jour $\mathbf{B}^{ij} \leftarrow \mathbf{W}\mathbf{B}^{ij}$
 - 10: Mettre à jour $\mathbf{R}_k \leftarrow \mathbf{B}^{ij}\mathbf{R}_k\mathbf{B}^{ij\dagger}$
 - 11: Mettre à jour $\mathbf{B} \leftarrow \mathbf{B}^{ij}\mathbf{B}$
 - 12: **end for**
 - 13: **until** convergence
 - 14: **output** : \mathbf{B}
-

3.4 Simulations

Dans ce paragraphe, nous illustrons le comportement des différents algorithmes de diagonalisation conjointe de matrices développés dans cette thèse : ALUJA, G-CLUJA, A-CLUJA et C-CLUJA. Nous comparons les performances des algorithmes proposés avec celles d'algorithmes de la littérature que nous présenterons dans les paragraphes suivants.

Afin d'évaluer les performances des différents algorithmes, nous utilisons l'indice suggéré dans [80, 85, 84] et défini par

$$I(\mathbf{S}) = \frac{1}{2N_s(N_s - 1)} \left(\sum_{i=1}^{N_s} \left(\sum_{j=1}^{N_s} \frac{|S_{ij}|^2}{\max_m |S_{im}|^2} - 1 \right) + \sum_{j=1}^{N_s} \left(\sum_{i=1}^{N_s} \frac{|S_{ij}|^2}{\max_m |S_{mj}|^2} - 1 \right) \right), \quad (3.116)$$

où $\mathbf{S} = (S_{ij}) = \mathbf{B}\mathbf{A}$ pour tout $(i, j) \in \mathcal{N}_s$. Cet indice de performance compare la matrice globale \mathbf{S} au produit d'une matrice diagonale \mathbf{D} et d'une matrice de permutation \mathbf{P} correspondant aux indéterminations classiques du problème. Ainsi lorsque $\mathbf{S} = \mathbf{D}\mathbf{P}$ (2.43), $I(\mathbf{S})$ vaut zéro.

Pour toutes les simulations, on considère K matrices suivant le modèle synthétique bruité

$$\mathbf{M}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^\dagger + t\mathbf{N}_k, \quad (3.117)$$

où \mathbf{N}_k est un bruit blanc gaussien et t est le niveau de bruit. Pour le modèle (3.117), les différentes matrices sont générées de manière aléatoire. La matrice facteur \mathbf{A} de dimensions

$N_o \times N_s$ est tirée suivant une loi uniforme dans l'intervalle $[-1, 1]$. Les parties réelle et imaginaire des matrices diagonales \mathbf{D}_k de dimensions $N_s \times N_s$ et des matrices \mathbf{N}_k de dimensions $N_o \times N_o$ suivent une loi normale centrée réduite. La matrice diagonalisante est initialisée par \mathbf{B}_o de dimensions $N_s \times N_o$, dont les parties réelle et imaginaire sont aussi tirées selon une loi normale centrée réduite. Pour l'algorithme G-CLUJA, le vecteur de paramètres $\mathbf{r}_{ij}^{(n)}$ pour la méthode du gradient est initialisé par $\mathbf{r}_{ij}^{(0)} = \begin{pmatrix} 0.1 & 0.1 \end{pmatrix}^T$.

Au moyen de simulations de type Monte-Carlo (n_{draws} tirages indépendants), nous étudions le comportement des différents algorithmes en fonction des trois paramètres : le nombre de matrices K , la dimension N_s des matrices et le niveau de bruit t . Afin de pouvoir comparer les différents algorithmes, nous fixons, pour chaque simulation, un nombre n_{sweeps} de balayages des couples (i, j) , $i < j$ qui seront effectués pour chacun des algorithmes. Aucun autre critère d'arrêt n'est considéré. Pour les différentes simulations, nous affichons soit

- les indices de performance pour l'intégralité des tirages, ce qui donne un aperçu global du comportement des algorithmes
- l'indice de performance moyen des tirages, qui donne une idée de la robustesse des algorithmes (en effet, $I(\mathbf{S})$ tend vers zéro lorsque les matrices \mathbf{M}_k sont approximativement diagonalisées et tend vers un quand l'algorithme diverge, ainsi les plus « mauvais tirages » pondéreront davantage l'indice de performance moyen)
- l'indice de performance médian des tirages, qui illustre le comportement optimal des algorithmes, mais ne nous renseigne pas sur leur robustesse (sauf, bien sûr, si plus de 50% des tirages donnent de mauvais résultats).

Remarques : G-CLUJA et C-CLUJA ont été testés pour différentes valeurs du nombre d'itérations n_{ite} de l'algorithme du gradient. La valeur n_{ite} minimum permettant la convergence de ces algorithmes pour chacun des tirages des simulations proposées a été déterminée empiriquement.

- Pour G-CLUJA, on prendra $n_{\text{ite}} = 10$ itérations.
- Pour C-CLUJA, grâce à la l'initialisation spécifique de $\mathbf{r}_{ij}^{(0)}$, on peut se limiter à seulement $n_{\text{ite}} = 2$ itérations.

3.4.1 Modèle complexe symétrique

Dans le cadre du modèle symétrique, nous comparons les performances des algorithmes ALUJA, G-CLUJA, A-CLUJA et C-CLUJA avec l'algorithme LUJ1D [1] adapté au cas complexe symétrique au paragraphe 3.2.1.1 et que l'on nomme CS-LUJ1D (pour *Complex Symmetric LUJ1D*). Le tableau 3.2 résume les paramètres de simulation pour chacune des figures

de ce paragraphe.

Figure	N_o	N_s	K	t	n_{draws}	n_{sweeps}	affichage
3.5	15	15	15	0	50	100	tous
3.6	15	15	15	0	50	100	méd. et moy.
3.7	15	15	15	10^{-2}	50	100	tous
3.8	15	15	15	10^{-2}	50	100	méd. et moy.
3.9	15	10	15	$\{10^{-6}, \dots, 10^{-1}\}$	200	150	méd.
3.10	40	$\{3, \dots, 30\}$	15	10^{-3}	300	120	méd.
3.11	40	30	15	10^{-3}	300	120	méd. et moy.
3.12	15	10	$\{2, \dots, 50\}$	10^{-3}	100	300	méd.
3.13	15	10	2	10^{-3}	100	300	méd. et moy.
3.14	50	50	3	0	50	100	méd. et moy.
3.15	50	50	3	0	50	100	tous

TABLE 3.2 – Paramètres de simulation pour les algorithmes de diagonalisation conjointe d'ensembles de matrices complexes symétriques

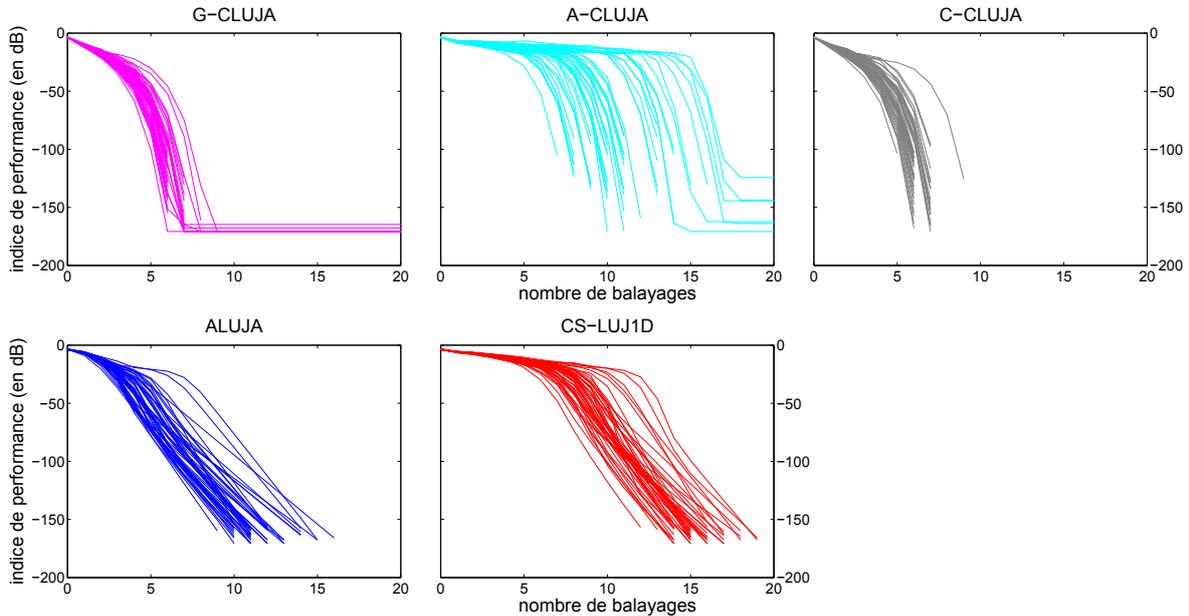


FIGURE 3.5 – Indice de performance en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.

Dans un premier temps, nous nous intéressons à un problème de diagonalisation conjointe d'un ensemble de 15 matrices 15×15 dont la dimension N_s vaut 10. Le comportement global des différents algorithmes est illustré dans un contexte non bruité par les figures 3.5 et 3.6 et,

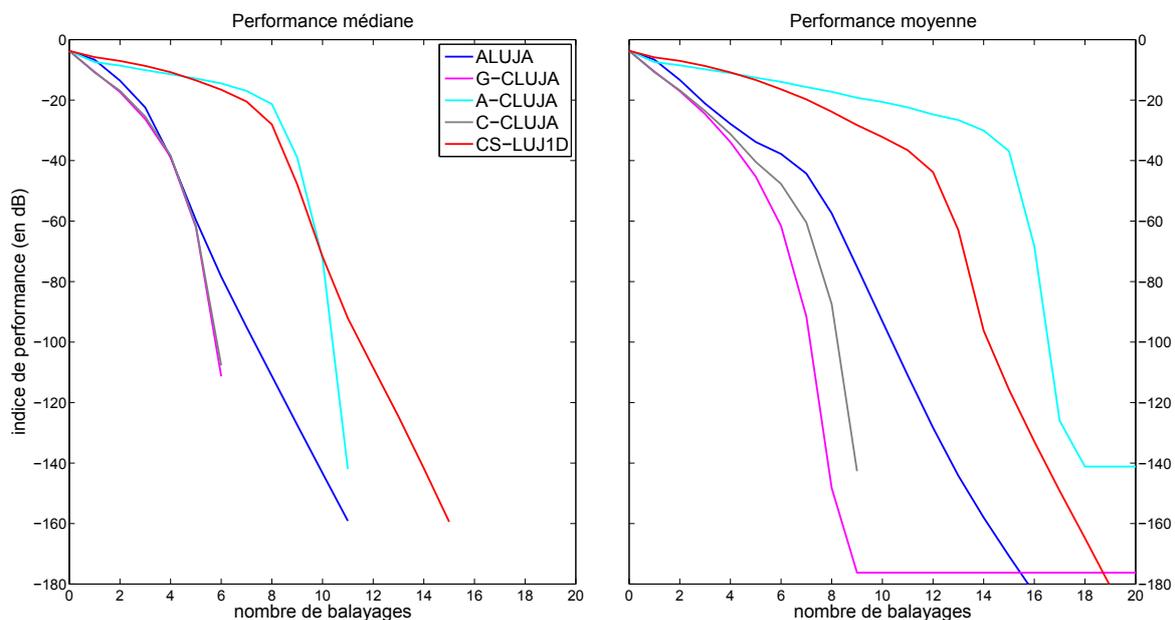


FIGURE 3.6 – Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.

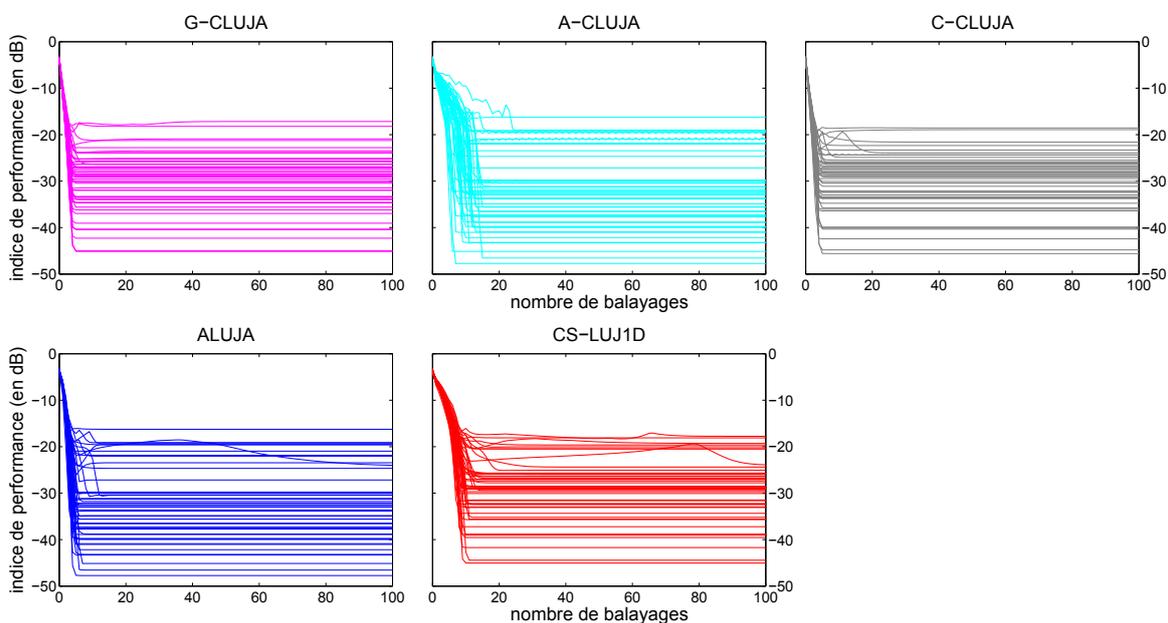


FIGURE 3.7 – Indice de performance en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.

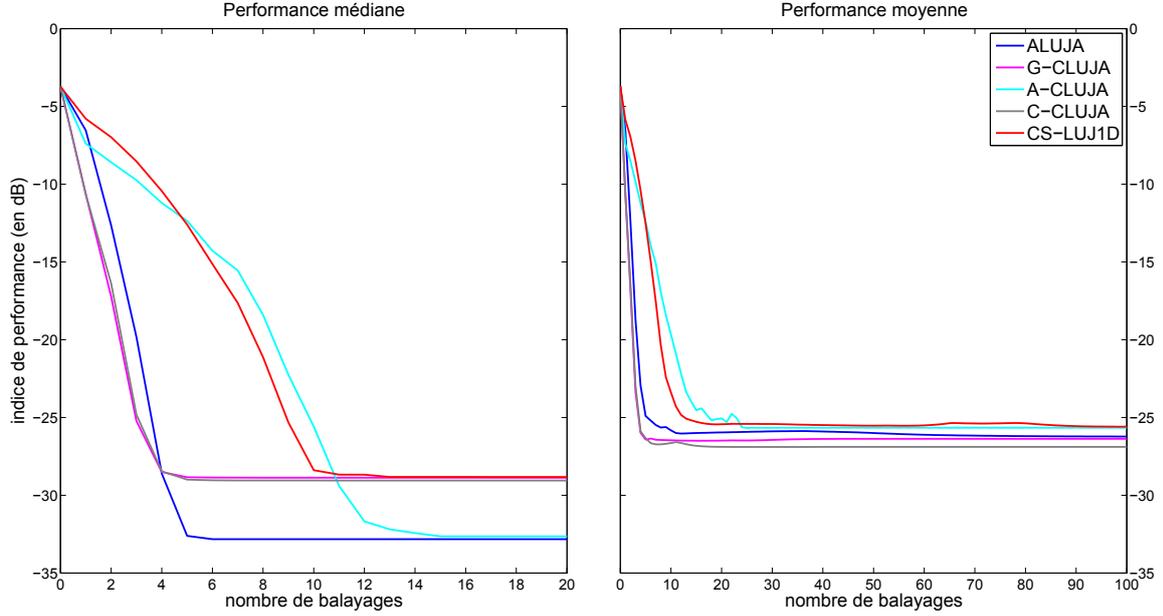


FIGURE 3.8 – Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.

Algorithmes	ALUJA	A-CLUJA	G-CLUJA	C-CLUJA	CS-LUJ1D
moyenne	$2.385e-3$	$2.718e-3$	$2.306e-3$	$2.047e-3$	$2.760e-3$
écart-type	$4.555e-3$	$4.841e-3$	$3.557e-3$	$2.793e-3$	$3.972e-3$
minimum	$1.685e-5$	$1.685e-5$	$3.065e-5$	$2.761e-5$	$3.137e-5$
maximum	$2.368e-2$	$2.367e-2$	$1.917e-2$	$1.396e-2$	$1.673e-2$
médiane	$5.221e-4$	$5.441e-4$	$1.297e-3$	$1.244e-3$	$1.311e-3$
écart interquartile	$9.231e-4$	$1.788e-3$	$1.907e-3$	$1.993e-3$	$2.078e-3$

TABLE 3.3 – Quelques statistiques après convergence pour 15 matrices complexes symétriques de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.

en présence de bruit ($t = 10^{-2}$), par les figures 3.7 et 3.8. Dans ces deux cadres de simulations, ALUJA mais surtout G-CLUJA et C-CLUJA se comparent favorablement en terme de vitesse de convergence par rapport à CS-LUJ1D. Seul A-CLUJA nécessite quelques balayages de plus pour converger dans le cas bruité. Toujours en présence de bruit, les cinq algorithmes atteignent des niveaux de performance similaires. Il est important de remarquer que, malgré l'initialisation aléatoire de B_o qui, potentiellement, ne rapproche pas d'une solution diagonalisante, les algorithmes basés sur l'approche adaptée (ALUJA et A-CLUJA) convergent tout de même. Le tableau 3.3 met en avant quelques éléments statistiques (moyenne, écart-type, mi-

nimum, maximum, médiane, écart interquartile) pour chaque algorithme après convergence, dans le même cadre de simulation que celui des figures 3.7 et 3.8. Les meilleurs résultats pour chacune des statistiques apparaissent en gras. Ces statistiques révèlent qu'ALUJA atteint les meilleures performances minimale et médiane, avec cependant le deuxième écart-type le plus élevé. Elles soulignent aussi le bon niveau de performance global de C-CLUJA, qui présente les meilleures performances moyenne et maximum, ainsi que sa bonne dispersion avec l'écart-type le plus faible.

La figure 3.9 montre l'influence du niveau de bruit sur le niveau de performance médian des algorithmes. Bien évidemment, plus le bruit est important moins, plus les indices de performance des algorithmes augmentent. Toutefois, on note que ce paramètre pénalise tous les algorithmes de la même façon.

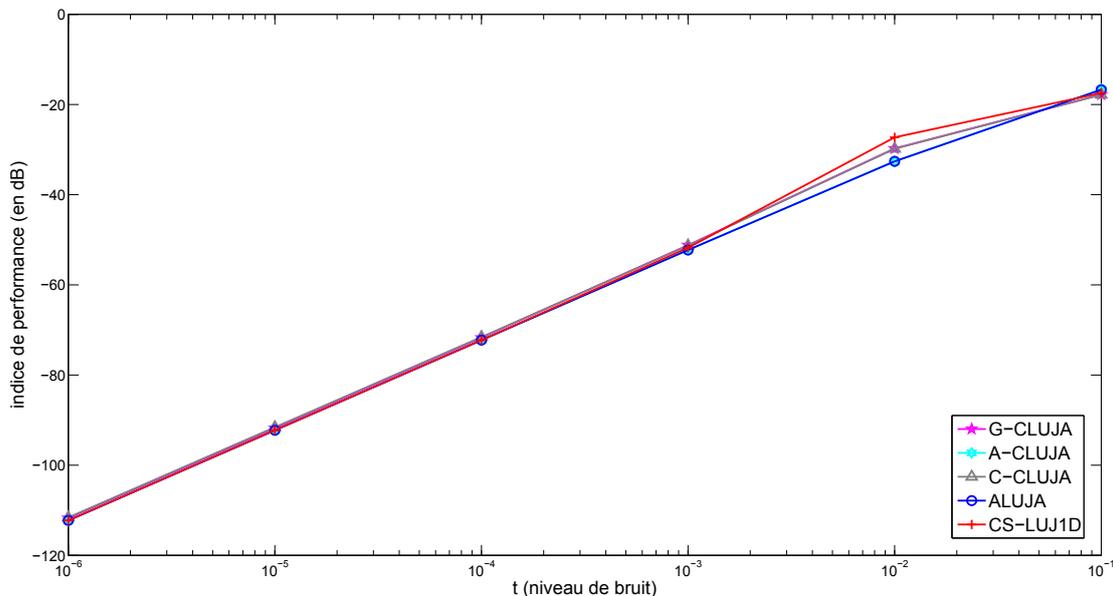


FIGURE 3.9 – Indice de performance médian après convergence en fonction du niveau de bruit t pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ et pour 200 tirages indépendants.

La figure 3.10 illustre l'influence de la dimension N_s sur l'indice de performance médian de chacun des algorithmes. On constate que les indices de performance des algorithmes augmentent lorsque N_s augmente. On remarque que pour un ensemble de 15 matrices, quand N_s devient supérieur à 15, les performances de l'algorithme A-CLUJA se détériorent. La figure 3.11 montre le comportement médian et moyen des algorithmes au cours des balayages, pour $N_s = 30$ sur la figure précédente. Dans ce contexte, l'algorithme A-CLUJA diverge tandis qu'ALUJA, G-CLUJA et C-CLUJA conservent leur bonne vitesse de convergence.

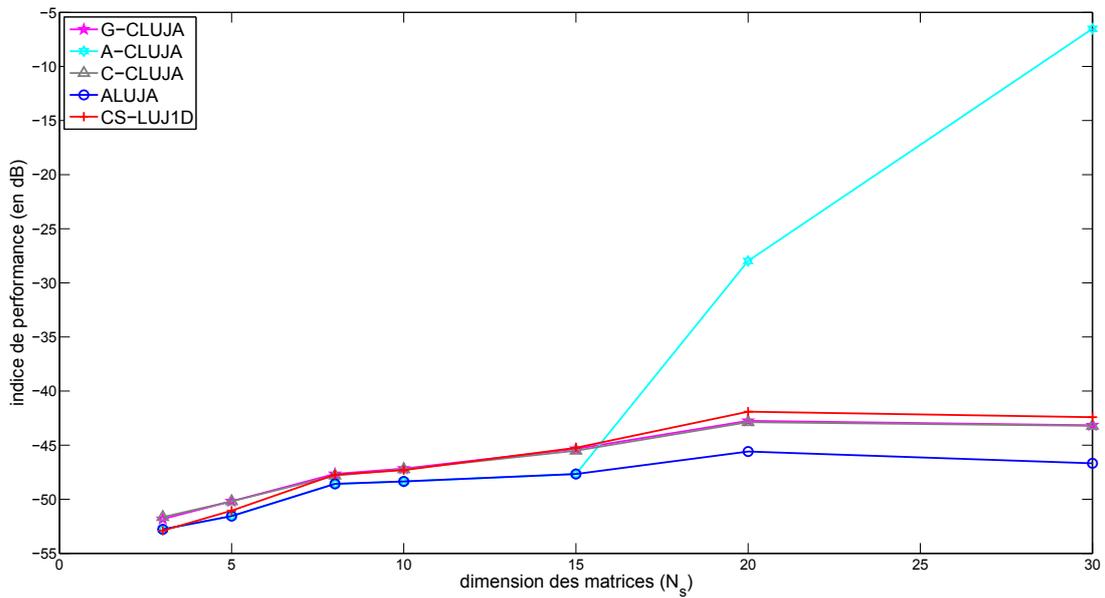


FIGURE 3.10 – Indice de performance médian après convergence en fonction de la dimension N_s des 15 matrices complexes symétriques 40×40 dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.

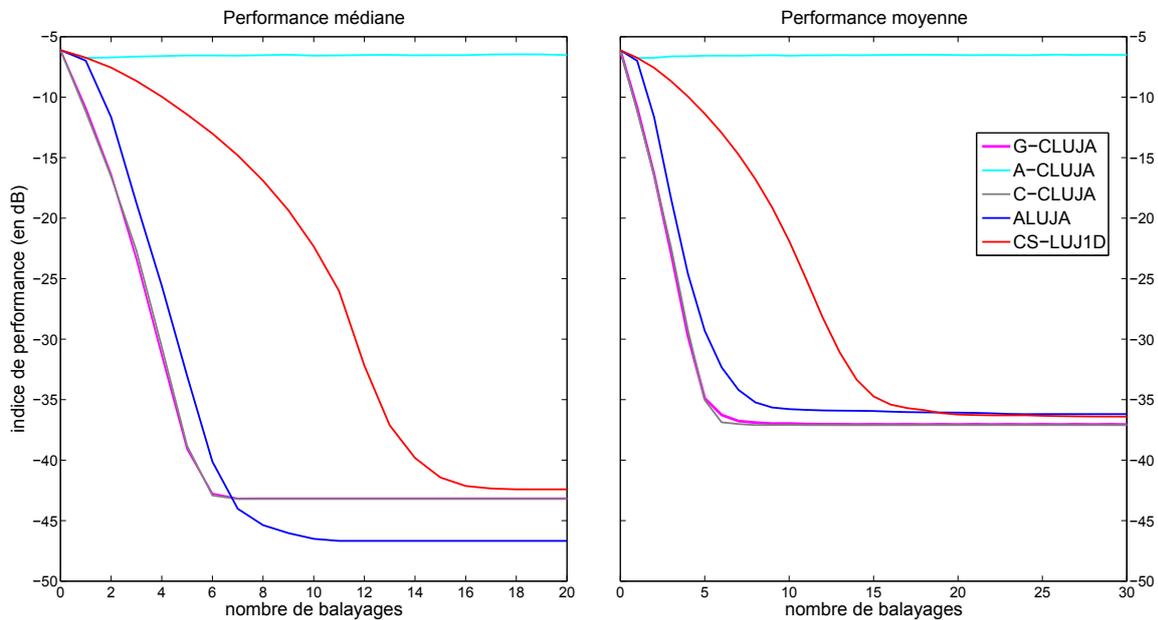


FIGURE 3.11 – Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices complexes symétriques de dimensions 40×40 avec $N_s = 30$ dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.

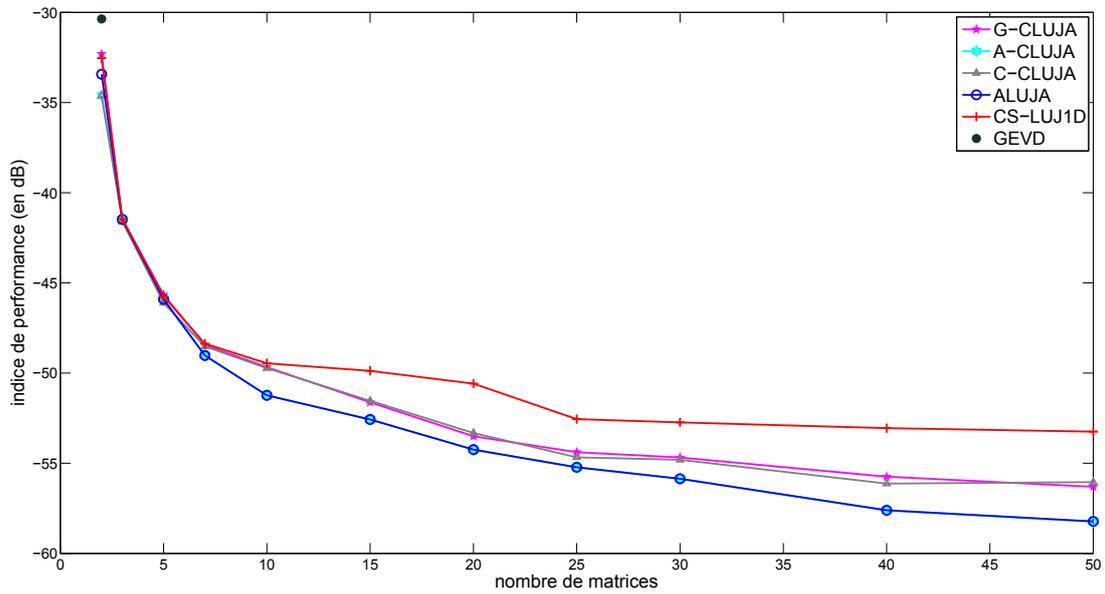


FIGURE 3.12 – Indice de performance médian après convergence en fonction du nombre K de matrices complexes symétriques de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 200 tirages indépendants.

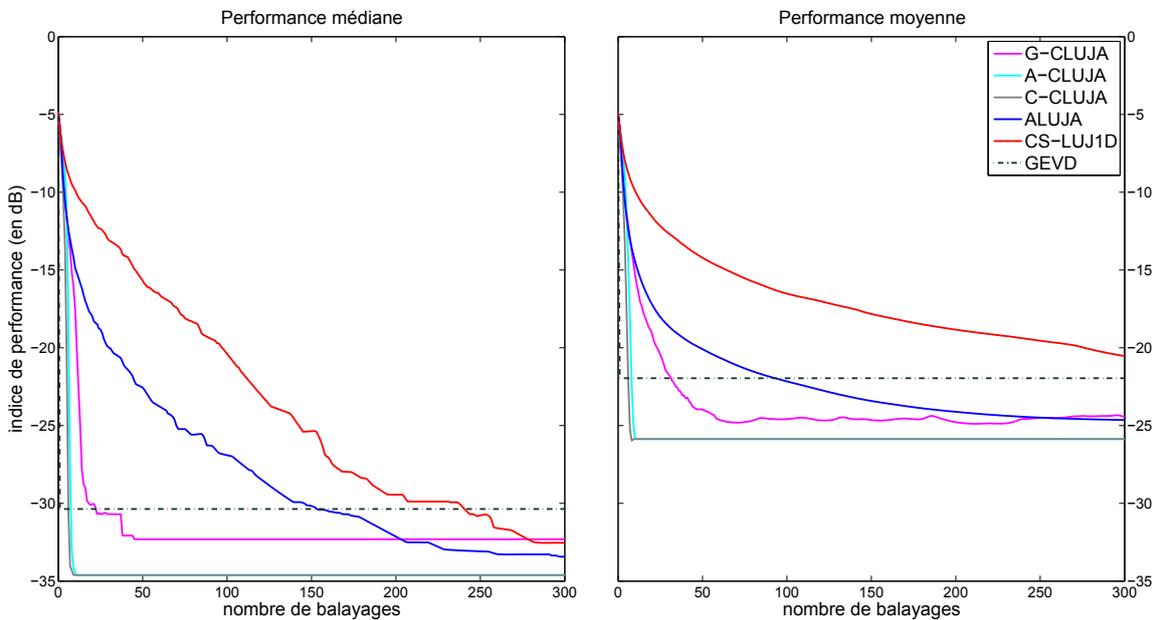


FIGURE 3.13 – Indices de performance médians et moyens en fonction du nombre de balayages pour 2 matrices complexes symétriques de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 100 tirages indépendants.

Enfin, intéressons nous à l'influence du nombre K de matrices à diagonaliser conjointement (voir figure 3.12). La principale information fournie par cette figure est que lorsque K diminue, les performances de chaque algorithme se dégradent. En effet, pour $K \geq 5$, les indices de performance médians des algorithmes sont inférieurs à -45dB , tandis que pour $K = 2$, ils sont tous supérieurs à -35dB . Lorsque l'on considère des ensembles de deux matrices, on peut utiliser la GEVD. On constate ici, qu'en termes de niveau de performance médian, les cinq algorithmes itératifs surpassent la GEVD. La figure 3.13 pointe ce contexte particulier où $K = 2$, en représentant les performances médianes et moyennes des algorithmes en fonction des balayages. Cette simulation révèle que le paramètre K impacte aussi la vitesse de convergence de certains algorithmes. On remarque, en effet, qu'A-CLUJA et C-CLUJA convergent en seulement une dizaine de balayages contre environ 50 pour G-CLUJA. ALUJA et CS-LUJ1D, quant à eux, n'ont pas assez de 300 balayages pour converger. Notons que CS-LUJ1D est le seul algorithme ne présentant pas un meilleur niveau de performance moyen que la GEVD après 300 balayages.

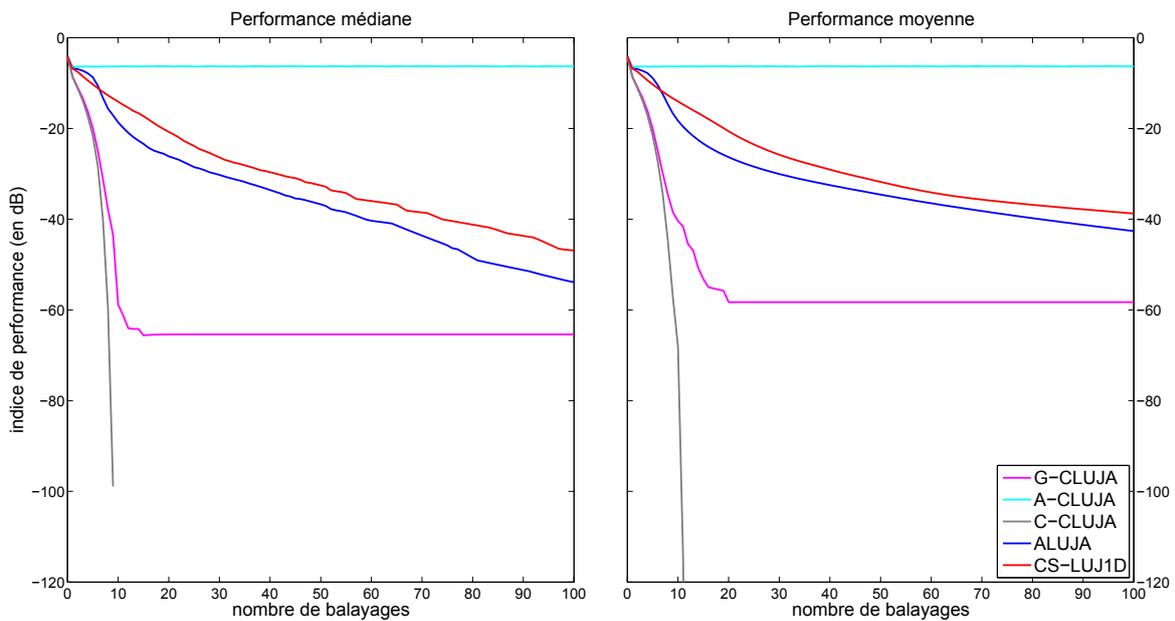


FIGURE 3.14 – Indices de performance médians et moyens en fonction du nombre de balayages pour 3 matrices complexes symétriques de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.

Finalement, les figures 3.14 et 3.15 illustrent un cas particulièrement ardu pour les algorithmes itératifs de diagonalisation conjointe de matrices. On considère des ensembles de trois matrices de grandes dimensions ($N_s = N_o = 50$), dans un contexte non bruité. Dans ces conditions, peu d'algorithmes itératifs parviennent à diagonaliser conjointement et exactement ces

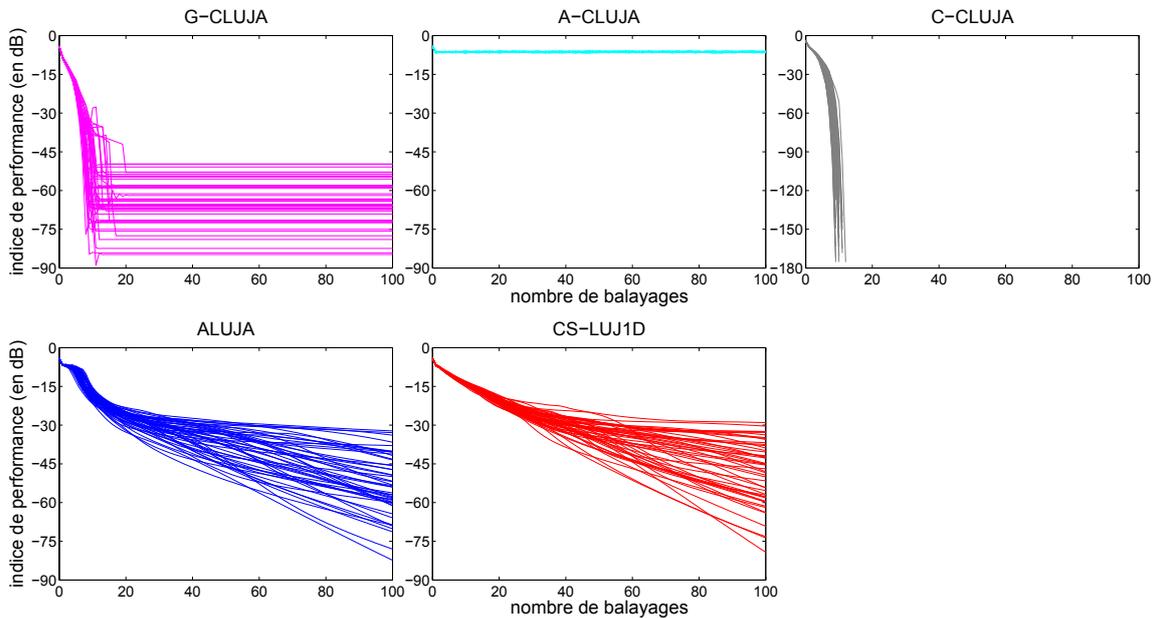


FIGURE 3.15 – Indice de performance en fonction du nombre de balayages pour 3 matrices complexes symétriques de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.

matrices. Sans surprise, pour des matrices de telle taille, A-CLUJA diverge. G-CLUJA se comporte comme en présence de bruit en convergeant à environ -60dB. ALUJA et CS-LUJ1D affichent une vitesse de convergence bien moins bonne que les algorithmes basés sur une méthode du gradient. Enfin l’algorithme C-CLUJA diagonalise exactement tous les ensembles de matrices considérés, avec de plus une bonne vitesse de convergence. En effet, cet algorithme, combinant A-CLUJA et G-CLUJA, atteint un indice de performance moyen égal à zéro en 12 balayages.

3.4.2 Modèle hermitien

Dans le cadre du modèle hermitien, nous comparons les performances des algorithmes ALUJA, G-CLUJA, A-CLUJA et C-CLUJA avec quatre des nombreux algorithmes traitant des ensembles de matrices hermitiennes que l’on trouve dans la littérature. Parmi eux, trois sont des algorithmes de type Jacobi, à savoir LUCJD [127], dont la méthode est résumée au paragraphe 3.2.1.2, JTJD [56] et CJD_i [78] et le dernier est un algorithme itératif utilisant une méthode de Gauss, dénommé U-WEDGE [114]. Le tableau 3.4 résume les paramètres de simulation pour chacune des figures de ce paragraphe.

Tout d’abord, on s’intéresse au comportement global des différents algorithmes dans un

Figure	N_o	N_s	K	t	n_{draws}	n_{sweeps}	affichage
3.16	15	15	15	0	50	100	tous
3.17	15	15	15	0	50	100	méd. et moy.
3.18	15	15	15	10^{-2}	50	100	tous
3.19	15	15	15	10^{-2}	50	100	méd. et moy.
3.20	15	10	15	$\{10^{-6}, \dots, 10^{-1}\}$	200	150	méd.
3.21	30	$\{3, \dots, 30\}$	15	10^{-3}	300	120	méd.
3.22	30	30	15	10^{-3}	300	120	méd. et moy.
3.23	15	10	$\{2, \dots, 50\}$	10^{-3}	100	300	méd.
3.24	15	10	2	10^{-3}	100	300	méd. et moy.
3.25	50	50	3	0	50	100	méd. et moy.
3.26	50	50	3	0	50	100	tous

TABLE 3.4 – Paramètres de simulation pour les algorithmes de diagonalisation conjointe d'ensembles de matrices complexes hermitiennes

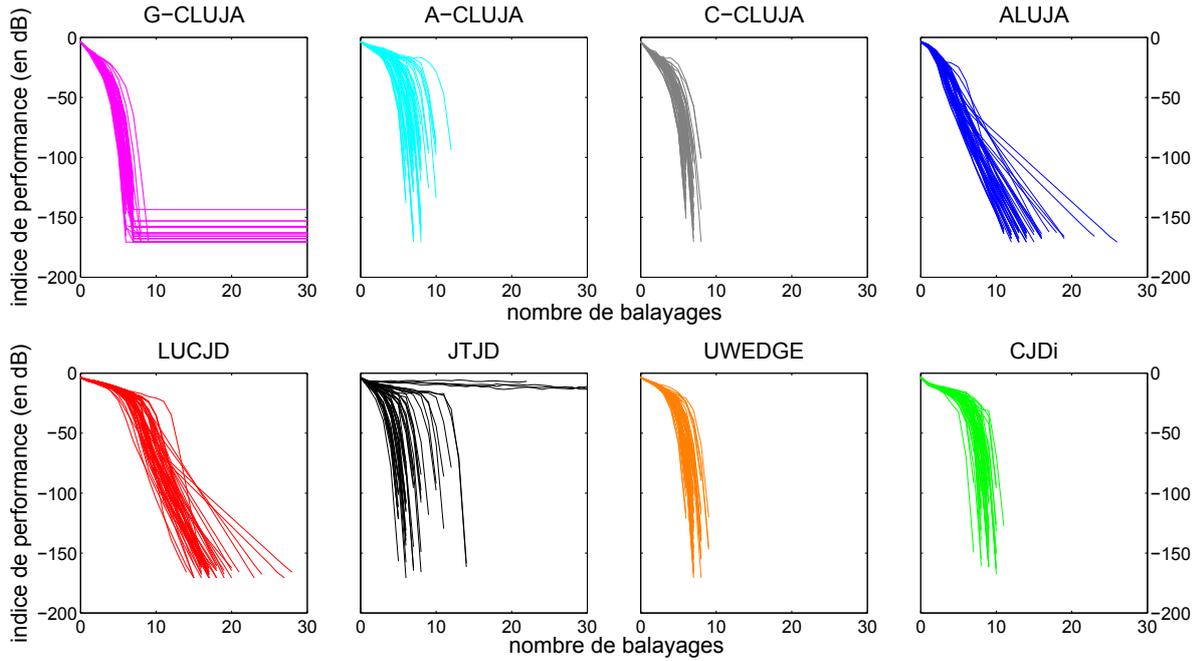


FIGURE 3.16 – Indice de performance en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.

contexte non bruité avec les figures 3.16 et 3.17 et en présence de bruit (avec $t = 10^{-2}$) avec les figures 3.18 et 3.19. Dans le contexte non bruité, ALUJA et LUCJD présentent une convergence linéaire, le premier étant sensiblement plus rapide, dans le sens où ALUJA atteint -160dB en moins d'itérations que LUCJD. Les six autres algorithmes convergent

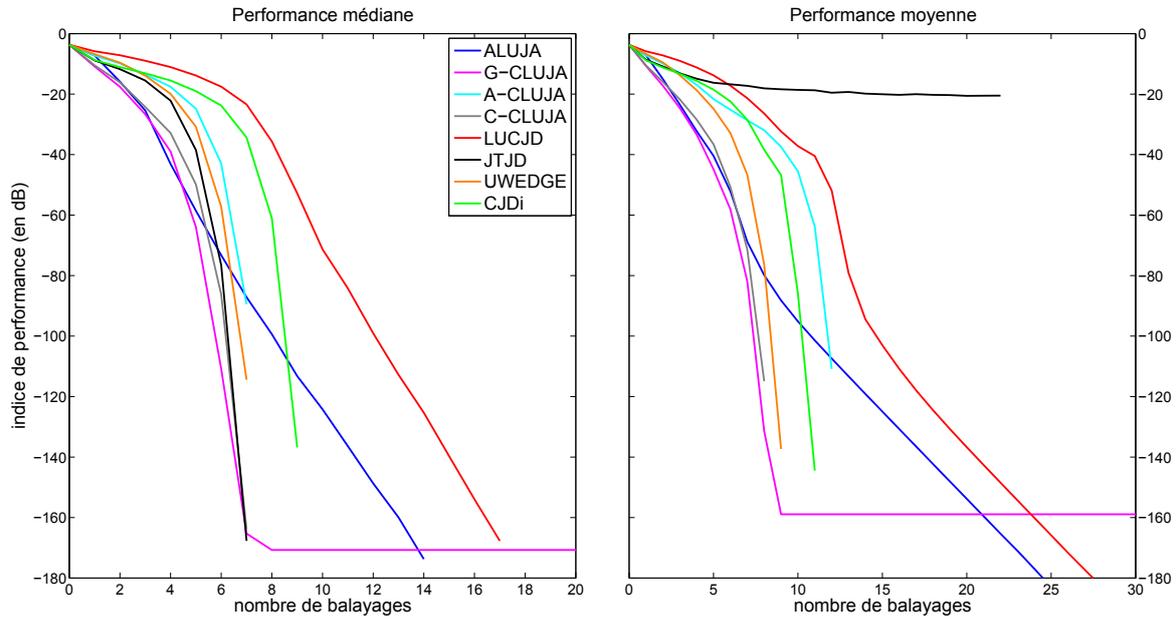


FIGURE 3.17 – Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte non bruité et pour 50 tirages indépendants.

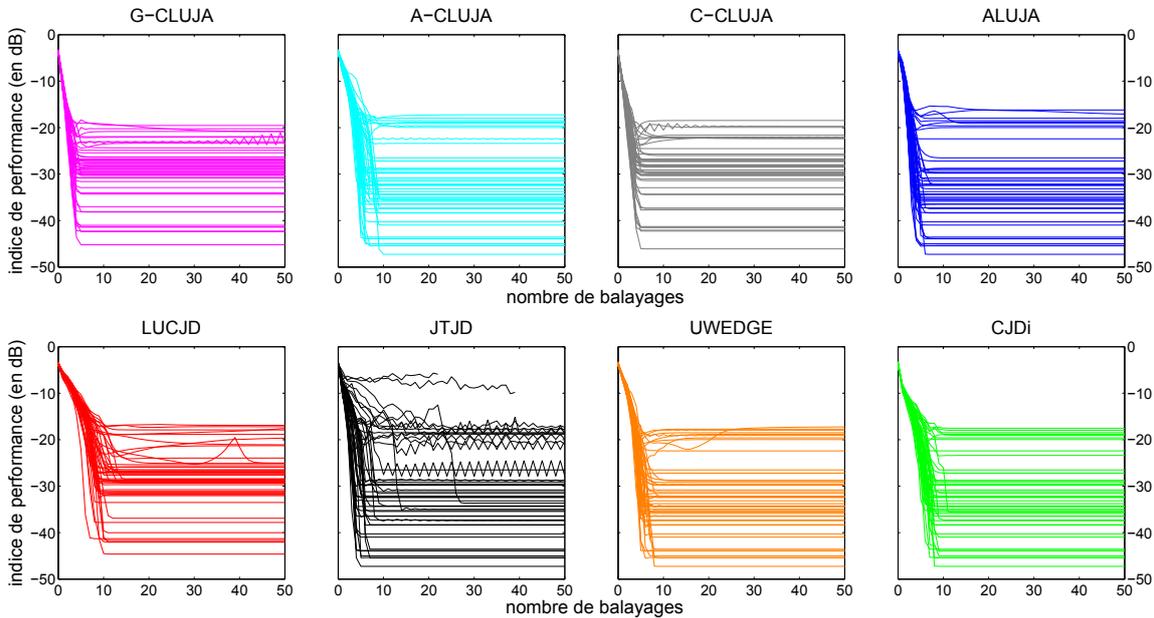


FIGURE 3.18 – Indice de performance en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.

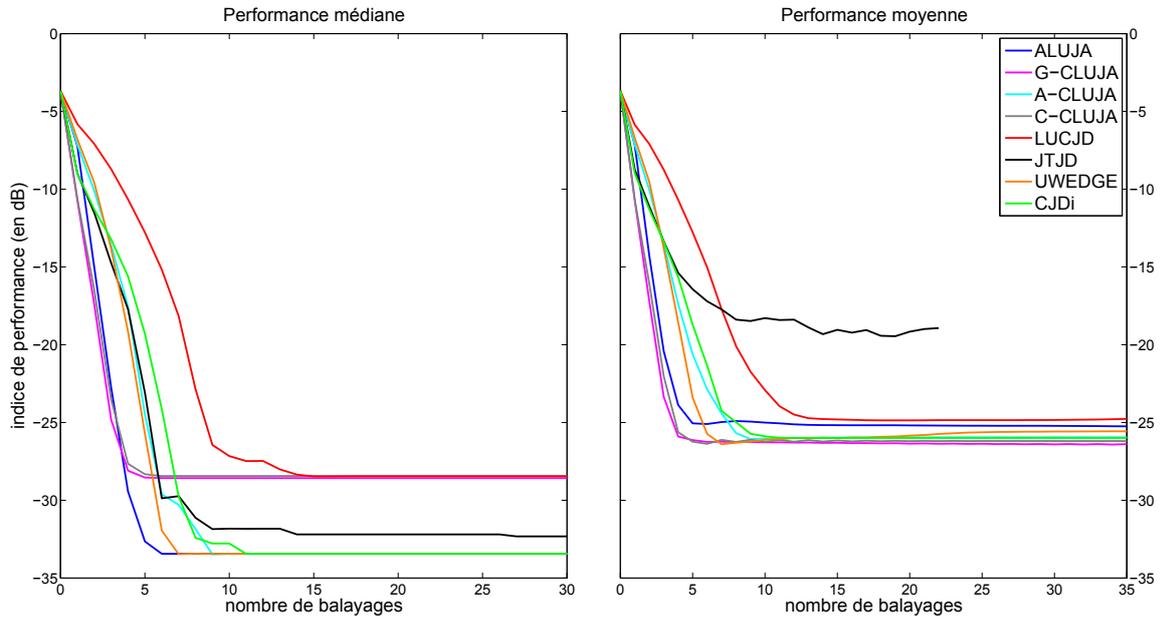


FIGURE 3.19 – Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.

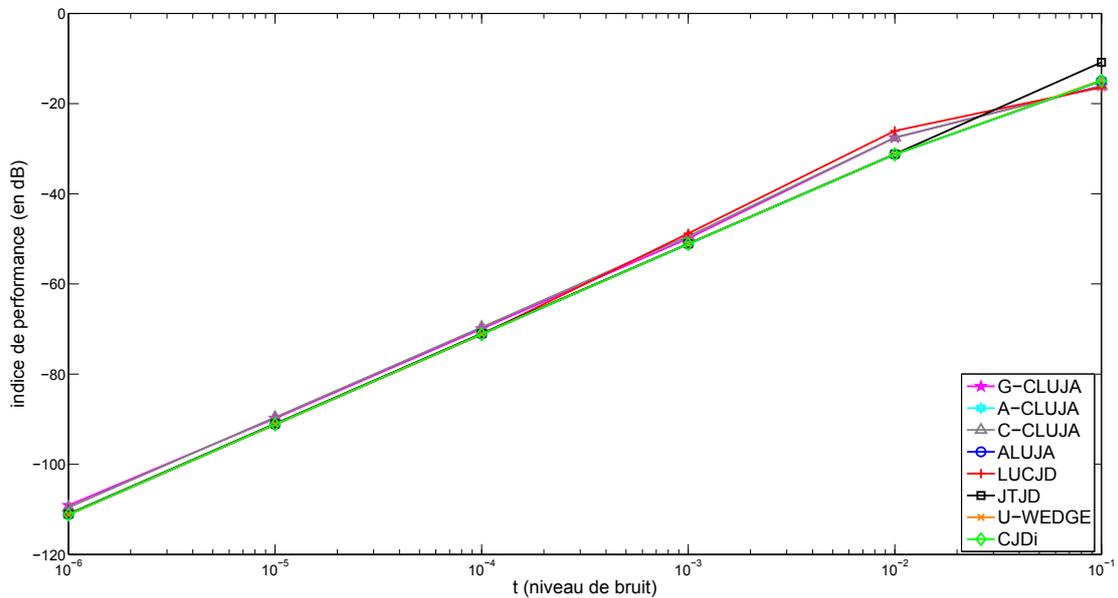


FIGURE 3.20 – Indice de performance médian après convergence en fonction du niveau de bruit t pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ et pour 200 tirages indépendants.

quadratiquement, G-CLUJA et C-CLUJA ayant les meilleures vitesses de convergence devant U-WEDGE. Notons que pour certains ensembles de matrices, JTJD diverge et ne peut pas traiter les n_{sweeps} balayages. Ceci explique pourquoi, dans le cas bruité, une des courbes s'arrête au 22^{ème} balayage. En termes de niveau de performance médian, ALUJA, A-CLUJA, U-WEDGE, JTJD et CJD*i* se détachent d'environ 5dB par rapport aux autres algorithmes.

Algorithmes	ALUJA	A-CLUJA	G-CLUJA	C-CLUJA
moyenne	$2.942e - 3$	$2.542e - 3$	$2.271e - 3$	$2.406e - 3$
écart-type	$5.681e - 3$	$4.765e - 3$	$2.599e - 3$	$3.023e - 3$
minimum	$1.882e - 5$	$1.882e - 5$	$3.019e - 5$	$2.463e - 5$
maximum	$2.395e - 2$	$1.890e - 2$	$1.110e - 2$	$1.425e - 2$
médiane	$4.524e - 4$	$4.524e - 4$	$1.387e - 3$	$1.433e - 3$
écart interquartile	$1.121e - 3$	$1.121e - 3$	$1.648e - 3$	$1.780e - 3$
Algorithmes	LUCJD	JTJD	U-WEDGE	CJD <i>i</i>
moyenne	$3.384e - 3$	N/A	$2.785e - 3$	$2.515e - 3$
écart-type	$5.129e - 3$	N/A	$5.166e - 3$	$4.677e - 3$
minimum	$3.456e - 5$	$1.882e - 5$	$1.882e - 5$	$1.882e - 5$
maximum	$2.034e - 2$	N/A	$1.880e - 2$	$1.761e - 2$
médiane	$1.426e - 3$	$5.858e - 4$	$4.524e - 4$	$4.524e - 4$
écart interquartile	$1.909e - 3$	$2.044e - 3$	$1.121e - 3$	$1.121e - 3$

TABLE 3.5 – Quelques statistiques après convergence pour 15 matrices hermitiennes de dimensions 15×15 avec $N_s = 15$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.

Le tableau 3.5 fournit quelques éléments statistiques (moyenne, écart-type, minimum, maximum, médiane, écart interquartile) pour chaque algorithme après convergence dans un cadre de simulation similaire à celui de la figure 3.18. Les meilleures statistiques y apparaissent en gras. Elles révèlent qu'ALUJA et A-CLUJA obtiennent les meilleurs minimum, médiane et écart interquartile avec U-WEDGE et CJD*i*. Cependant, les écart-types de ces quatre algorithmes sont nettement moins bons que ceux de C-CLUJA et surtout G-CLUJA. Les algorithmes de type gradient présentent donc moins de dispersion. On peut aussi noter qu'ils atteignent les meilleurs niveaux de performance moyens.

La figure 3.20 permet d'évaluer l'influence du niveau de bruit sur l'indice de performance médian des algorithmes. Comme dans le cas complexe symétrique, on observe que ce paramètre influe sur chacun des algorithmes de manière similaire.

La figure 3.21 illustre l'influence de la dimension N_s des matrices sur l'indice de performance médian des algorithmes et la figure 3.22 détaille le cas $N_s = 30$. Comme précédemment, pour un nombre K de matrices fixé, l'augmentation de la taille des matrices entraîne une augmentation de la valeur de l'indice de performance pour chacun des algorithmes. On note que, lorsque $N_s = 30$, l'indice de performance médian de JTJD n'est pas défini (ce qui signifie

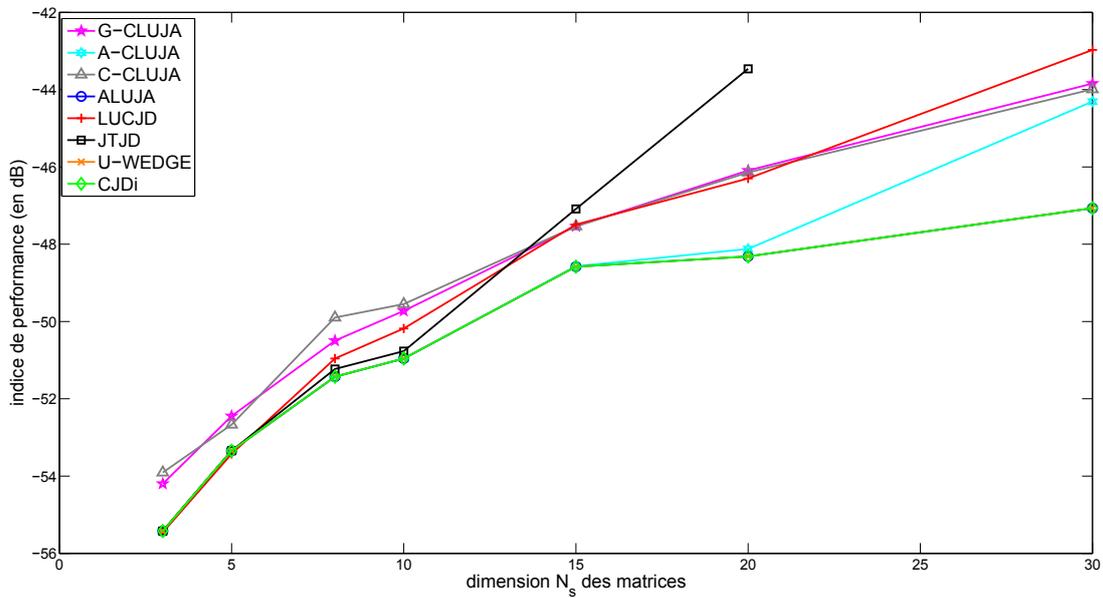


FIGURE 3.21 – Indice de performance médian après convergence en fonction de la dimension N_s des 15 matrices hermitiennes 30×30 dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.

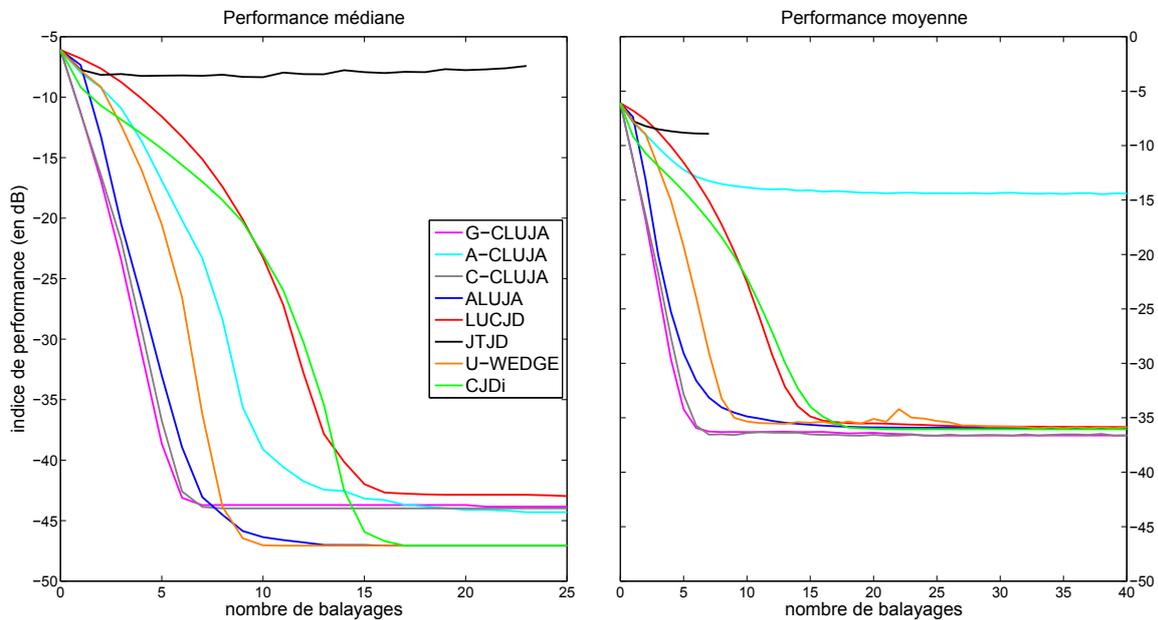


FIGURE 3.22 – Indices de performance médians et moyens en fonction du nombre de balayages pour 15 matrices hermitiennes de dimensions 30×30 avec $N_s = 30$ dans un contexte bruité $t = 10^{-3}$ et pour 300 tirages indépendants.

que l'algorithme diverge dans plus de 50% des tirages) et l'indice de performance moyen de A-CLUJA est supérieur -15dB . La taille N_s des matrices n'a que peu d'influence sur les autres algorithmes.

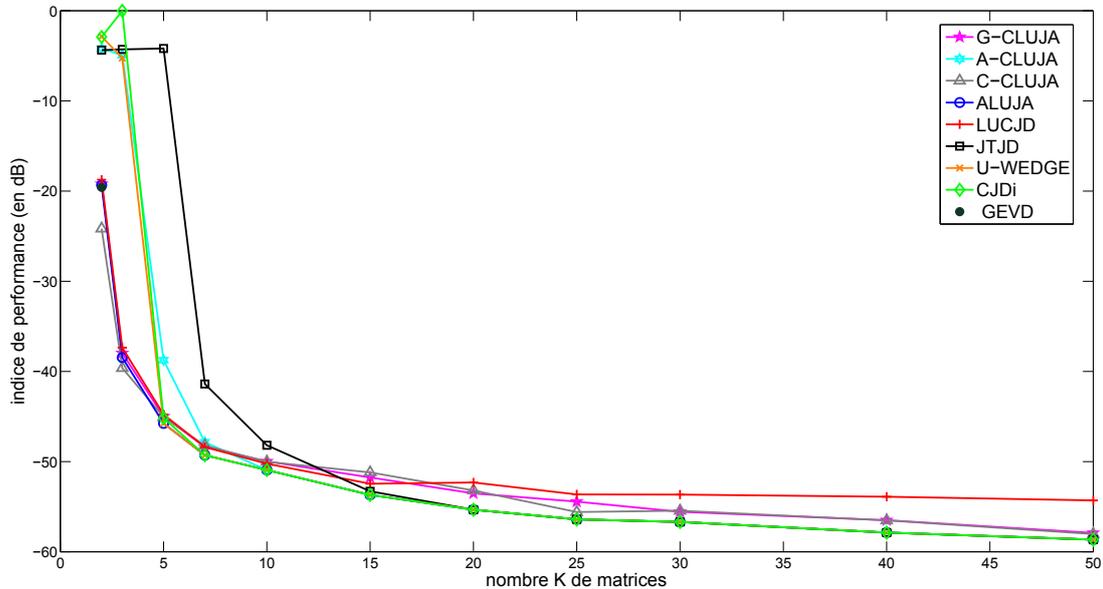


FIGURE 3.23 – Indice de performance médian après convergence en fonction du nombre K de matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 200 tirages indépendants.

La figure 3.23 montre l'influence du nombre K de matrices contenues dans l'ensemble à diagonaliser sur le niveau de performance médian des différents algorithmes et la figure 3.24 illustre, plus particulièrement, le cas où $K = 2$. Plus l'ensemble contient de matrices, plus les indices de performance des algorithmes sont faibles. Toutefois, cette amélioration est moins nette pour LUCJD. Les ensembles de peu de matrices font surgir des problèmes de robustesse. En effet, pour $K = 2$, U-WEDGE, JTJD, CJDj et A-CLUJA divergent. Dans ce même cas, les algorithmes basés sur la stratégie d'estimation découplée mettent beaucoup plus d'itérations que G-CLUJA et C-CLUJA pour converger. Ces derniers algorithmes présentent un comportement oscillant. Enfin, on note que seul C-CLUJA affiche un indice de performance médian inférieur à celui de la GEVD et celui-ci vaut environ -24dB .

Enfin, pour les figures 3.25 et 3.26, on considère la diagonalisation conjointe d'ensembles de trois matrices de dimensions $N_o = N_s = 50$ dans un cas non bruité. Dans ce contexte, les algorithmes U-WEDGE, JTJD, CJDj et A-CLUJA divergent. ALUJA ne converge que pour quelques-uns des tirages. LUCJD affiche une convergence très lente par rapport aux algorithmes reposant sur une méthode du gradient. G-CLUJA ne diagonalise pas de façon

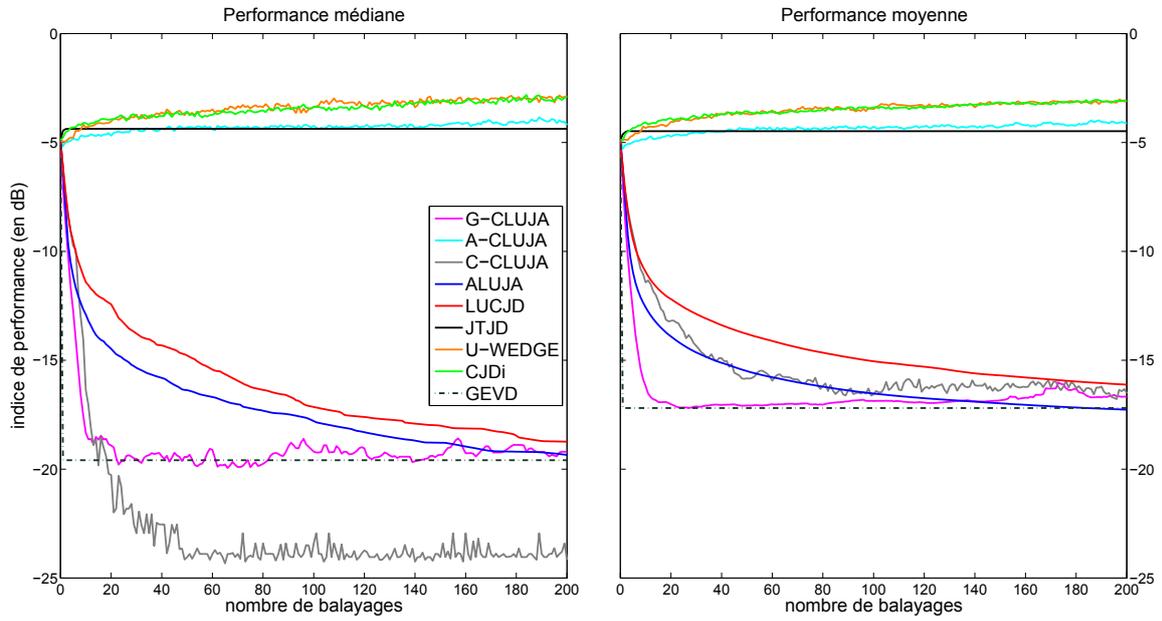


FIGURE 3.24 – Indices de performance médians et moyens en fonction du nombre de balayages pour 2 matrices hermitiennes de dimensions 15×15 avec $N_s = 10$ dans un contexte bruité $t = 10^{-3}$ et pour 100 tirages indépendants.

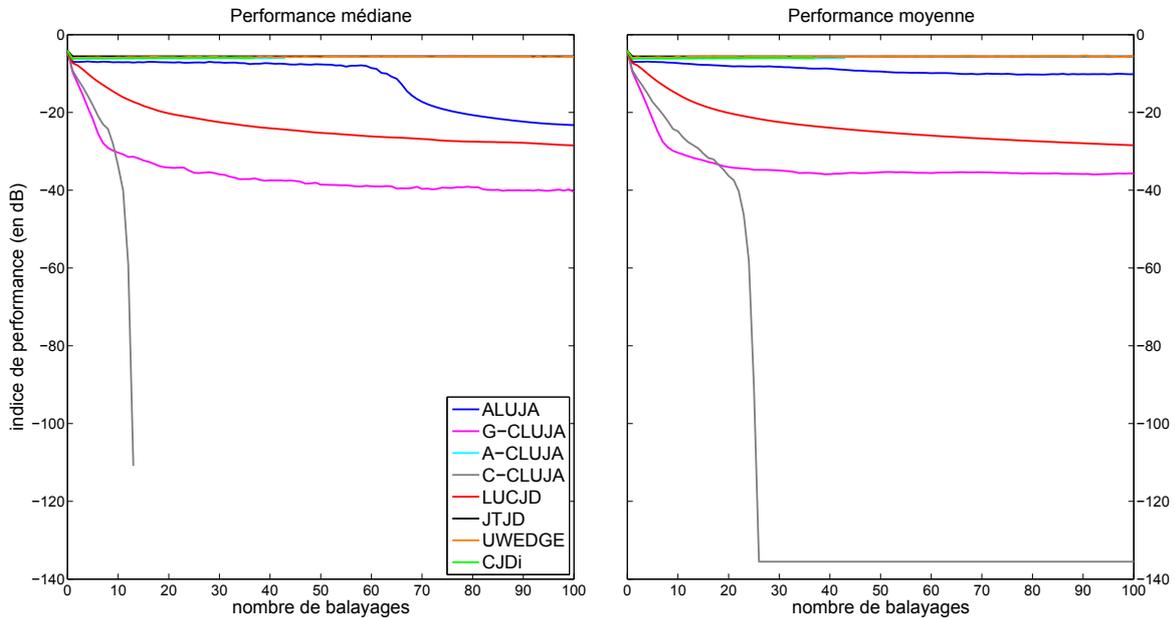


FIGURE 3.25 – Indices de performance médians et moyens en fonction du nombre de balayages pour 3 matrices hermitiennes de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.

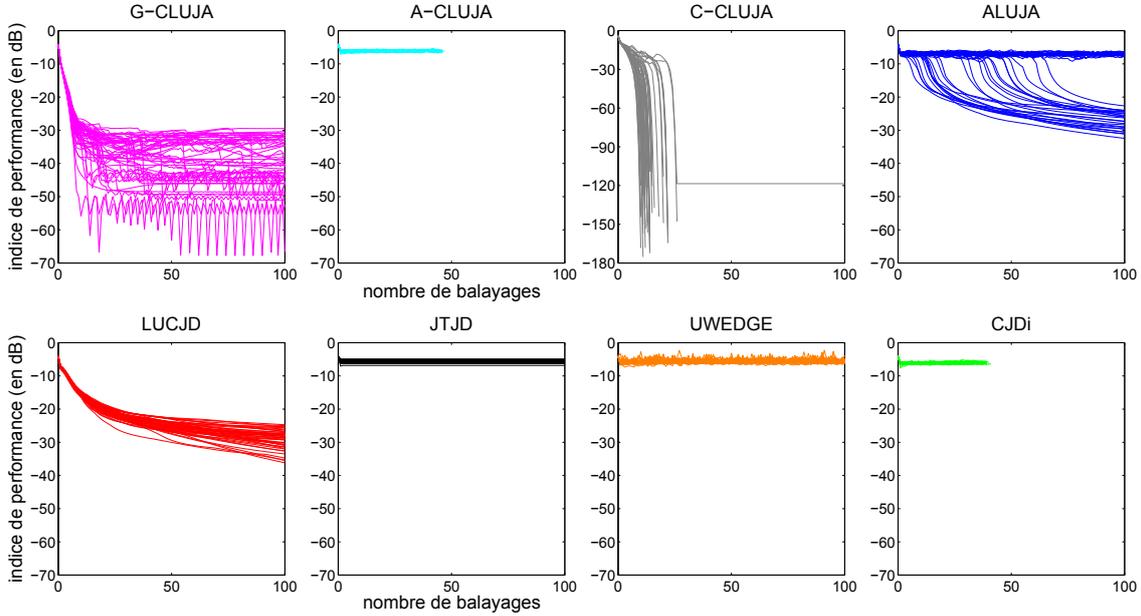


FIGURE 3.26 – Indice de performance en fonction du nombre de balayages pour 3 matrices hermitiennes de dimensions 50×50 avec $N_s = 50$ dans un contexte non bruité et pour 50 tirages indépendants.

exacte les matrices des n_{draw} ensembles considérés. Finalement, comme dans le cas complexe symétrique, seul C-CLUJA obtient des indices de performance nuls pour l'intégralité des ensembles de matrices à diagonaliser conjointement.

3.4.3 Complexité algorithmique

Intéressons nous, maintenant, à la complexité des algorithmes proposés et de ceux avec lesquels nous les avons comparés. Les complexités sont données en opérations à virgule flottante ou flop pour *F*loating-*p*oint *O*perations. Ici, on considère qu'un flop est constitué d'une multiplication suivie d'une addition.

Tous les algorithmes de ce chapitre sont structurés en deux étapes distinctes : l'estimation des paramètres et la mise à jour des matrices cibles. Pour beaucoup d'algorithmes comme U-WEDGE, JTJD, CJDi, ainsi que les algorithmes proposés reposant sur l'approche adaptée, à savoir ALUJA et A-CLUJA, la complexité de l'étape d'estimation des paramètres est négligeable devant celle de l'étape de mise à jour. La mise à jour d'une matrice-cible par une matrice élémentaire vaut $\mathcal{O}(2N_s)$ flop, cette opération est répétée pour toutes les matrices cibles (K fois) et pour toutes les matrices élémentaires ($\frac{N_s(N_s-1)}{2}$ fois). Pour un balayage, la mise à jour complète des matrices cibles vaut donc $\mathcal{O}(KN_s^3)$ flop. Ceci correspond à la

complexité des algorithmes sus-cités.

Algorithmes	Complexité (en flop/balayage)
ALUJA	$\mathcal{O}(KN_s^3)$
A-CLUJA	$\mathcal{O}(KN_s^3)$
G-CLUJA ($n_{\text{ite}} = 10$)	$\mathcal{O}((10n_{\text{ite}} + 1)KN_s^3)$
C-CLUJA ($n_{\text{ite}} = 2$)	$\mathcal{O}((10n_{\text{ite}} + 1)KN_s^3)$
CS-LUJ1D	$\mathcal{O}(2KN_s^3)$
LUCJD	$\mathcal{O}(2KN_s^3)$
U-WEDGE	$\mathcal{O}(KN_s^3)$
JTJD	$\mathcal{O}(KN_s^3)$
CJDi	$\mathcal{O}(KN_s^3)$

TABLE 3.6 – Récapitulatifs des complexités de chacun des algorithmes de diagonalisation conjointe de matrices proposés et utilisés dans ce chapitre

Concernant les algorithmes CS-LUJ1D et LUCJD, la complexité de l'estimation de tous leurs paramètres n'est plus négligeable, mais exactement du même ordre de grandeur que celle de la mise à jour globale des matrices cibles. La complexité de ces deux algorithmes vaut donc $\mathcal{O}(2KN_s^3)$ flop par balayage.

Enfin, pour G-CLUJA et C-CLUJA, la complexité algorithmique d'une des n_{ite} itérations de la méthode du gradient présente un coût dix fois plus élevé qu'une mise à jour des matrices cibles. Ainsi, la complexité de ces deux algorithmes vaut $\mathcal{O}((10n_{\text{ite}} + 1)KN_s^3)$ flop par balayage. Notons que, pour toutes les simulations de ce chapitre, on a pris $n_{\text{ite}} = 10$ pour G-CLUJA et $n_{\text{ite}} = 2$ pour C-CLUJA. Le tableau 3.6 récapitule les complexités de chaque algorithme.

Conclusion

Comme l'attestent les résultats obtenus suite aux simulations numériques, deux des algorithmes proposés dans ce chapitre sortent du lot, en se comparant favorablement aux algorithmes de la littérature. Tout d'abord, ALUJA offre une bonne estimation de la matrice diagonalisante. En effet, cet algorithme, dont l'implémentation est très simple, fournit un bon niveau de performance avec de plus, une faible complexité algorithmique. Cependant, dans des situations particulières, il peut nécessiter un grand nombre d'itérations pour converger. L'algorithme C-CLUJA offre, quand à lui, une bonne vitesse de convergence. De plus, cette combinaison des algorithmes G-CLUJA et A-CLUJA présente un niveau de performance tout à fait correct et converge pour tous les tirages, même dans les configurations les plus défavorables envisagées dans ce chapitre.

Algorithmes de diagonalisation conjointe non-unitaire de tenseurs d'ordre trois

Dans ce chapitre, nous développons des méthodes pour la diagonalisation conjointe non-unitaire de tenseurs complexes d'ordre trois. Pour traiter les problèmes de séparation multi-entrées multi-sorties de sources de télécommunications numériques, nous nous concentrons sur des ensembles de tenseurs présentant des symétries et plus précisément, ceux reposant sur le modèle symétrique (2.38) et sur la décomposition INDSCAL_h (2.36). Nous nous intéressons aussi à la diagonalisation et à la diagonalisation conjointe de tenseurs non-symétriques reposant sur le modèle (2.23). Même si notre approche est différente, cela nous permettra de nous comparer aux méthodes de décomposition CP classique de la littérature comme par exemple l'ALS ou les méthodes de type gradient conjugué ou Levenberg-Marquardt.

4.1 Introduction

Tout d'abord, rappelons rapidement comment est formulé le problème de diagonalisation conjointe de tenseurs d'ordre trois dans le cas du modèle non-symétrique (2.23). On considère un ensemble de K ($K \geq 1$) tenseurs d'ordre trois $\mathbf{T}(k)$ respectant le modèle générique suivant pour tout $k \in \mathcal{K}$

$$\mathbf{T}(k) = \mathbf{D}(k) \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \mathbf{A}_3. \quad (4.1)$$

où pour tout $x \in \{1, 2, 3\}$, $\mathbf{A}_x \in \mathcal{A}_{N_{o_x} \times N_s}$ (2.3) avec $N_s \leq \min_x(N_{o_x})$ et où $\mathbf{D}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ sont diagonaux et non nuls. Dans le cas symétrique, on aura $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}_3 = \mathbf{A}$ et dans le cas INDSCAL_h , on aura par exemple $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}$ et $\mathbf{A}_3 = \mathbf{A}^*$ (notons toutefois que toutes les combinaisons sont envisageables tant qu'une des trois matrices facteurs correspond à la conjuguée des deux autres). On cherche à estimer les matrices diagonalisantes $\mathbf{B}_x \in \mathcal{B}_{N_s \times N_{o_x}}$,

avec $x \in \{1, 2, 3\}$ telles que les tenseurs transformés

$$\mathbf{T}'(k) = \mathbf{T}(k) \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \times_3 \mathbf{B}_3 \quad (4.2)$$

soient aussi diagonaux que possible. Pour cela, nous avons choisi de mettre en œuvre les méthodes présentées au paragraphe 2.5. Nous développons des algorithmes de type Jacobi reposant sur une paramétrisation LU des matrices diagonalisantes. Toutefois, seules les matrices carrées peuvent être décomposées à l'aide d'une procédure de type Jacobi. Or, ici, les matrices \mathbf{B}_x sont toutes trois rectangulaires et de dimensions $N_s \times N_{o_x}$. En supposant que nous connaissons toutes les dimensions du problème (N_s et N_{o_x} pour $x \in \{1, 2, 3\}$), nous proposons de décomposer les matrices \mathbf{B}_x telles que

$$\mathbf{B}_x = \mathbf{B}_{xr} \mathbf{B}_{xo} \quad (4.3)$$

où $\mathbf{B}_{xo} \in \mathbb{C}^{N_s \times N_{o_x}}$ sont trois matrices de rang plein en ligne et $\mathbf{B}_{xr} \in \mathbb{C}^{N_s \times N_s}$ sont trois matrices inversibles. Les matrices \mathbf{B}_{xo} correspondent à une initialisation pour les matrices diagonalisantes \mathbf{B}_x respectives et \mathbf{B}_{xr} sont leurs matrices diagonalisantes réduites que l'on doit estimer. Après les initialisations $\mathbf{B}_x \leftarrow \mathbf{B}_{xo}$, l'ensemble des tenseurs $\mathbf{T}(k)$ de dimensions $N_{o_1} \times N_{o_2} \times N_{o_3}$ (4.1) est mis à jour par la transformation suivante

$$\mathbf{R}(k) = \mathbf{T}(k) \times_1 \mathbf{B}_{1o} \times_2 \mathbf{B}_{2o} \times_3 \mathbf{B}_{3o} \quad (4.4)$$

où $\mathbf{R}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ pour tout $k \in \mathcal{K}$.

Remarques : En séparation aveugle de source, on connaît seulement les N_{o_x} . Toutefois, il existe, dans la littérature, de nombreuses méthodes permettant d'estimer N_s à partir du jeu d'observations considéré et ainsi d'effectuer la réduction de dimension escomptée.

Finalement, à partir des tenseurs cibles réduits $\mathbf{R}(k)$, on veut estimer les matrices diagonalisantes réduites \mathbf{B}_{xr} que l'on décompose de la manière suivante, à l'aide d'une procédure de type Jacobi

$$\mathbf{B}_{xr} = \prod_{i=1}^{N_s-1} \prod_{j=i+1}^{N_s} \mathbf{B}_{xr}^{ij} \quad (4.5)$$

Pour chaque couple $(i, j) \in \mathcal{N}_s^2$, $i < j$, on propose, si possible, d'estimer analytiquement et simultanément les matrices \mathbf{B}_{1r}^{ij} , \mathbf{B}_{2r}^{ij} et \mathbf{B}_{3r}^{ij} en optimisant le critère

$$\mathcal{J}(\mathbf{B}_{xr}^{ij}) = \sum_{k=1}^K \|\text{ZDiag}\{\mathbf{R}'(k)\}\|^2 \quad (4.6)$$

avec

$$\mathbf{R}'(k) = \mathbf{R}(k) \times_1 \mathbf{B}_{1r} \times_2 \mathbf{B}_{2r} \times_3 \mathbf{B}_{3r}. \quad (4.7)$$

On considèrera deux approches d'optimisation différentes pour ce critère :

- Une approche classique où l'on minimise directement (4.6) sans hypothèse particulière
- Une approche adaptée décrite au paragraphe 2.5.2, où l'on minimise le critère (4.6) en supposant, cette fois, que les tenseurs $\mathbf{R}(k)$ sont proches d'être diagonaux.

Enfin, pour tout $x \in \{1, 2, 3\}$, afin d'assurer la pseudo-inversibilité des \mathbf{B}_x et d'éviter toute solution singulière, nous considérons la contrainte $\det(\mathbf{B}_{xr}^{ij}) = 1$. Pour imposer cette contrainte, nous factorisons chaque \mathbf{B}_{xr}^{ij} à l'aide d'une décomposition LU. De ce fait, chacune de ces matrices a pour partie utile

$$\tilde{\mathbf{B}}_{xr}^{ij} = \tilde{\mathbf{L}}_x^{ij} \tilde{\mathbf{U}}_x^{ij} = \begin{pmatrix} 1 & 0 \\ \ell_x^{ij} & 1 \end{pmatrix} \begin{pmatrix} 1 & u_x^{ij} \\ 0 & 1 \end{pmatrix}. \quad (4.8)$$

On propose, ici, d'estimer les paramètres des matrices \mathbf{L}_x^{ij} et \mathbf{U}_x^{ij} selon la stratégie découpée envisagée au paragraphe 2.5.3. Elle consiste à estimer $\mathbf{B}_{xr}^{ij} = \mathbf{U}_x^{ij}$ à \mathbf{L}_x^{ij} fixée, pour chacun des couples (i, j) avec $1 \leq i < j \leq N_s$, puis à faire de même pour l'estimation des matrices élémentaires \mathbf{L}_x^{ij} .

Finalement, quel que soit l'ensemble de tenseurs d'ordre trois considéré (symétrique, INDSCAL_h ou non-symétrique), nous avons développé deux méthodes basées sur cette stratégie découpée d'estimation des paramètres.

- La première méthode, présentée au paragraphe 4.2.1, est basée sur l'optimisation du critère (4.6) selon l'approche classique. Pour cet algorithme, chacun des paramètres u_x^{ij} et ℓ_x^{ij} est estimé indépendamment et analytiquement pour tout $x \in \{1, 2, 3\}$ et tout $(i, j) \in \mathcal{N}_s^2$, $i < j$. On dénomme l'algorithme T-ALUJA pour *Tensorial ALUJA*.
- La seconde méthode repose sur la minimisation du critère (4.6), en supposant que l'on se trouve proche d'une solution diagonalisante. Ici, pour un couple $(i, j) \in \mathcal{N}_s^2$, $i < j$ fixé, les trois matrices \mathbf{U}_x^{ij} (respectivement \mathbf{L}_x^{ij}) seront estimées simultanément. L'algorithme résultant est nommé T-ALUJA-A pour *T-ALUJA for an Adapted approach* et il est présenté au paragraphe 4.2.2.

Pour ces deux méthodes, l'optimisation du critère se fait par la recherche des racines de sa dérivée.

Le tableau 4.1 récapitule les algorithmes développés dans cette thèse pour la diagonalisation conjointe de tenseurs complexes d'ordre trois.

Pour une meilleure lisibilité, dans les paragraphes suivants de ce chapitre, nous n'écrirons plus l'indice r des matrices élémentaires réduites \mathbf{B}_{xr}^{ij} . Pour un couple (i, j) , $i < j$ donné, on

Stratégie Approche	Découplée
Classique	T-ALUJA : §4.2.1 Modèle non-symétrique Modèle INDSCAL _h Modèle symétrique
Adaptée	T-ALUJA-A : §4.2.2 Modèle non-symétrique Modèle INDSCAL _h Modèle symétrique

TABLE 4.1 – Récapitulatif des algorithmes tensoriels de diagonalisation conjointe non-unitaire proposés selon l’approche liée au critère d’optimisation et la stratégie d’estimation des paramètres choisis

rappelle et on définit les ensembles que l’on utilisera dans les paragraphes suivants :

$$\mathcal{P} = \mathcal{N}_s \setminus \{i\} \quad \mathcal{S} = \mathcal{N}_s^2 \setminus \{(i, i)\}. \quad (4.9)$$

4.2 Stratégie d’estimation découplée

Rappelons, tout d’abord, la décomposition de chacune des matrices diagonalisantes réduites

$$\mathbf{B}_x = \prod_{i=1}^{N_s-1} \prod_{j=i+1}^{N_s} \mathbf{L}_x^{ij} \prod_{i=1}^{N_s-1} \prod_{j=i+1}^{N_s} \mathbf{U}_x^{ij} \quad (4.10)$$

où les matrices \mathbf{L}_x^{ij} et \mathbf{U}_x^{ij} sont entièrement caractérisées par leurs parties utiles définies respectivement par

$$\tilde{\mathbf{U}}_x^{ij} = \begin{pmatrix} 1 & u_x^{ij} \\ 0 & 1 \end{pmatrix} \quad \text{et} \quad \tilde{\mathbf{L}}_x^{ij} = \begin{pmatrix} 1 & 0 \\ \ell_x^{ij} & 1 \end{pmatrix}. \quad (4.11)$$

Dans ce paragraphe, dans le cadre de l’estimation découplée des paramètres, pour un couple $(i, j) \in \mathcal{N}_s^2$, $i < j$ donné, le but est de déterminer les matrices \mathbf{U}_x^{ij} avec $x \in \{1, 2, 3\}$ à \mathbf{L}_x^{ij} fixées de telle sorte que

$$\hat{u}_x^{ij} = \arg \min_{\mathbf{U}_x^{ij}} \mathcal{J}(\mathbf{U}_x^{ij}). \quad (4.12)$$

Le calcul des matrices \mathbf{L}_x^{ij} avec $x \in \{1, 2, 3\}$ à \mathbf{U}_x^{ij} fixées se fait de façon similaire. Dans la suite de ce paragraphe, on se contentera de développer les calculs pour la détermination des paramètres u_x^{ij} . En effet, comme les \mathbf{L}^{ij} sont les transposées de matrices triangulaires supérieures similaires aux \mathbf{U}_x^{ij} , on déduira donc directement la solution optimale des paramètres

ℓ_x^{ij} , en interchangeant les indices i et j dans les développements suivants.

Soit un ensemble de K tenseurs cibles réduits $\mathbf{R}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ admettant le modèle (2.26), la stratégie d'estimation découplée est donc basée sur la minimisation du critère

$$\mathcal{J}(\mathbf{U}_x^{ij}) = \sum_{k=1}^K \|\text{ZDiag}\{\mathbf{R}'(k)\}\|^2, \quad (4.13)$$

avec

$$\mathbf{R}'(k) = \mathbf{R}(k) \times_1 \mathbf{U}_1^{ij} \times_2 \mathbf{U}_2^{ij} \times_3 \mathbf{U}_3^{ij}. \quad (4.14)$$

Dans cette transformation (4.14), chacune des trois matrices \mathbf{U}_x^{ij} modifie une tranche spécifique des tenseurs $\mathbf{R}(k)$. Pour tout $k \in \mathcal{K}$, \mathbf{U}_1^{ij} transforme la tranche de $\mathbf{R}(k)$ définie par $\{(i, m, n) \mid (m, n) \in \mathcal{N}_s^2\}$, \mathbf{U}_2^{ij} la tranche définie par $\{(m, i, n) \mid (m, n) \in \mathcal{N}_s^2\}$ et \mathbf{U}_3^{ij} celle définie par $\{(m, n, i) \mid (m, n) \in \mathcal{N}_s^2\}$. Les composantes situées aux intersections de ces plans sont modifiées par deux voire trois des paramètres u_x^{ij} . On dira des composantes impactées par deux matrices \mathbf{U}_x^{ij} qu'elles ont subi une transformation bilinéaire. La seule composante ayant subi une transformation trilinéaire est l'élément diagonal $R'_{iii}(k)$, qui ne nous intéresse pas pour la diagonalisation. La figure 4.1 illustre l'impact de la transformation sur un tenseur $\mathbf{R}'(k)$ et met en évidence les composantes qui ont été modifiées entre ce tenseur transformé et le tenseur cible $\mathbf{R}(k)$ correspondant.

Ainsi on peut développer le critère (4.13) de la manière suivante

$$\mathcal{J}(\mathbf{U}_x^{ij}) = \mathcal{J}'(\mathbf{U}_x^{ij}) + C_{tdc} \quad (4.15)$$

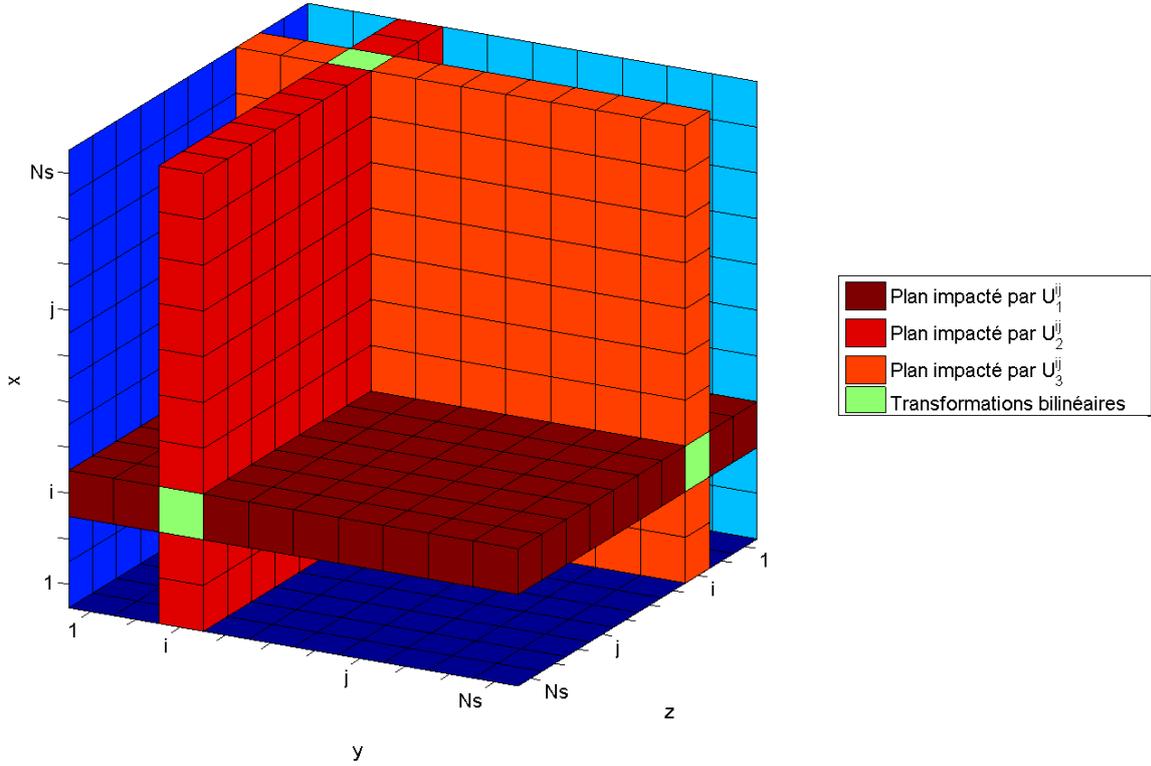
avec

$$\begin{aligned} \mathcal{J}'(\mathbf{U}_x^{ij}) = \sum_{k=1}^K \left[\sum_{l \in \mathcal{P}} \left(|R'_{iil}(k)|^2 + |R'_{ili}(k)|^2 + |R'_{lii}(k)|^2 \right) \right. \\ \left. + \sum_{(m,n) \in \mathcal{P}^2} \left(|R'_{imn}(k)|^2 + |R'_{min}(k)|^2 + |R'_{imn}(k)|^2 \right) \right] \end{aligned} \quad (4.16)$$

$$C_{tdc} = \sum_{k=1}^K \sum_{\substack{(l,m,n) \in \mathcal{P}^3 \\ l \neq m \neq n}} |R'_{lmn}(k)|^2. \quad (4.17)$$

Minimiser $\mathcal{J}(\mathbf{U}_x^{ij})$ revient donc à minimiser $\mathcal{J}'(\mathbf{U}_x^{ij})$.

Pour tout $k \in \mathcal{K}$ et pour tout $(l, m, n) \in \mathcal{P}^3$, dans (4.16) les termes hors-diagonaux d'intérêt

FIGURE 4.1 – Schéma de l'impact de la transformation (4.14) sur un tenseur $\mathbf{R}'(k)$.

des $\mathbf{R}'(k)$ s'écrivent

$$\begin{aligned}
 R'_{iil}(k) &= R_{iil}(k) + R_{jil}(k)u_1^{ij} + R_{ijl}(k)u_2^{ij} + R_{jil}(k)u_1^{ij}u_2^{ij} \\
 R'_{ili}(k) &= R_{ili}(k) + R_{jli}(k)u_1^{ij} + R_{ilj}(k)u_3^{ij} + R_{jlj}(k)u_1^{ij}u_3^{ij} \\
 R'_{lii}(k) &= R_{lii}(k) + R_{lji}(k)u_2^{ij} + R_{lij}(k)u_3^{ij} + R_{ljj}(k)u_2^{ij}u_3^{ij} \\
 R'_{imn}(k) &= R_{imn}(k) + R_{jmn}(k)u_1^{ij} \\
 R'_{min}(k) &= R_{min}(k) + R_{mjn}(k)u_2^{ij} \\
 R'_{mni}(k) &= R_{mni}(k) + R_{mnj}(k)u_3^{ij}.
 \end{aligned} \tag{4.18}$$

$$\tag{4.19}$$

Maintenant que nous avons l'expression exacte du critère, nous développons des méthodes de détermination des paramètres u_x^{ij} , tout d'abord selon l'approche classique, puis selon l'approche adaptée.

4.2.1 Approche classique

Idéalement, cette approche consiste à estimer les paramètres u_1^{ij} , u_2^{ij} et u_3^{ij} simultanément en minimisant directement (4.16). Cependant les composantes ayant subi une transformation bilinéaire (4.18) présentent respectivement des termes croisés en $u_1^{ij}u_2^{ij}$, $u_1^{ij}u_3^{ij}$ et $u_2^{ij}u_3^{ij}$. Dans (4.16), c'est le module au carré de chacune de ces composantes qui est considéré. Si on minimise directement ce critère pour estimer les trois matrices \mathbf{U}_x^{ij} , on obtiendra, dans le meilleur des cas, un système de trois équations à trois inconnues présentant des termes croisés en $(|u_1^{ij}|^2 u_2^{ij})$, $(|u_1^{ij}|^2 u_3^{ij})$, $(u_1^{ij} |u_2^{ij}|^2)$, $(|u_2^{ij}|^2 u_3^{ij})$, $(u_1^{ij} |u_3^{ij}|^2)$ et $(u_2^{ij} |u_3^{ij}|^2)$. Un tel système ne semble pas pouvoir être résolu analytiquement. Notons que c'est aussi le cas pour les modèles complexes INDSCAL_h et symétrique. L'optimisation directe du critère (4.16) semble fournir une solution analytique, uniquement pour le modèle réel symétrique.

On envisage donc une autre méthode. On propose d'estimer chacune des matrices \mathbf{U}_x^{ij} , $x \in \{1, 2, 3\}$ indépendamment. Ainsi pour un couple $(i, j) \in \mathcal{N}_s^2$, $i < j$ donné, on détermine, dans un premier temps, \mathbf{U}_1^{ij} à \mathbf{U}_2^{ij} et \mathbf{U}_3^{ij} fixées, on met ensuite à jour les tenseurs cibles $\mathbf{R}(k)$ ainsi que la matrice diagonalisante \mathbf{B}_1 puis on fait de même pour l'estimation de \mathbf{U}_2^{ij} et \mathbf{U}_3^{ij} . Par la suite, on ne développera les calculs que pour l'estimation de \mathbf{U}_1^{ij} . Celles de \mathbf{U}_2^{ij} et \mathbf{U}_3^{ij} pourront être simplement déduites par permutation d'indices.

Le paramètre u_1^{ij} est obtenu en minimisant le critère (4.13) appliqué à la seule matrice \mathbf{U}_1^{ij} et que l'on dénote $\mathcal{J}_{(\times_1)}$, de telle sorte que

$$\hat{u}_1^{ij} = \arg \min_{\mathbf{U}_1^{ij}} \mathcal{J}_{(\times_1)}(\mathbf{U}_1^{ij}) \quad (4.20)$$

avec

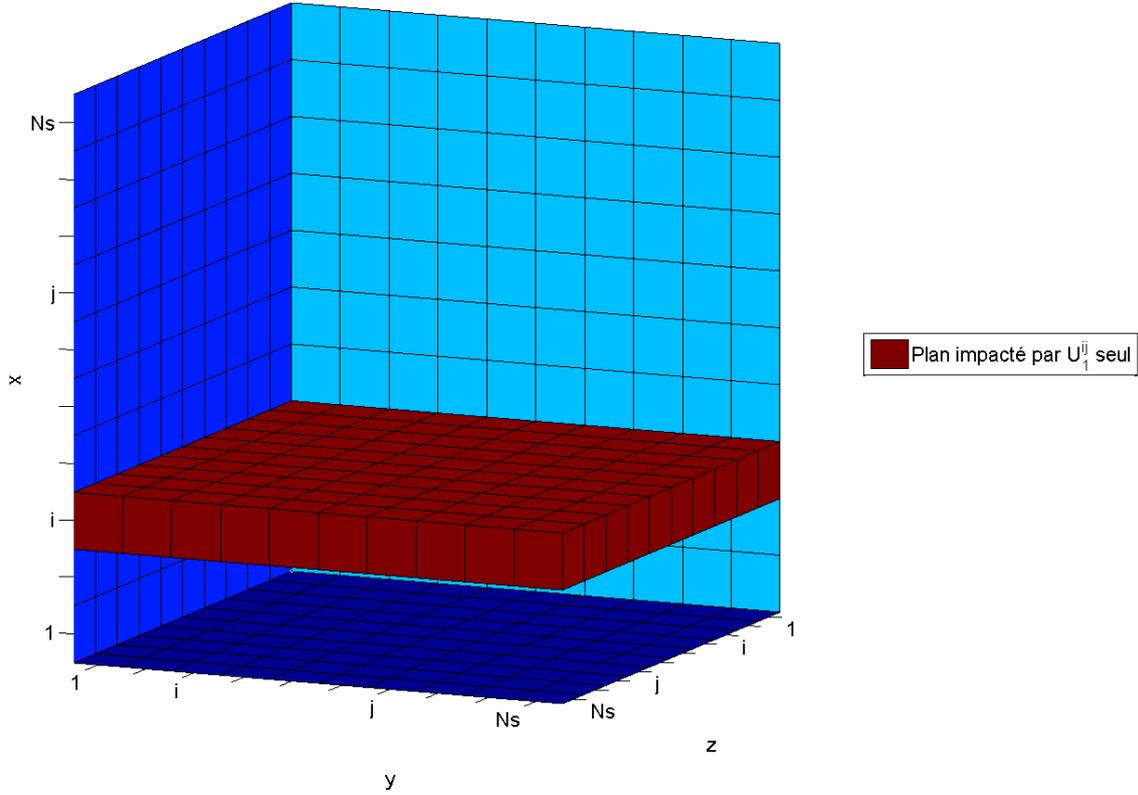
$$\mathcal{J}_{(\times_1)}(\mathbf{U}_1^{ij}) = \sum_{k=1}^K \|\text{ZDiag}\{\mathbf{R}'_1(k)\}\|^2, \quad (4.21)$$

où

$$\mathbf{R}'_1(k) = \mathbf{R}(k) \times_1 \mathbf{U}_1^{ij}. \quad (4.22)$$

Cette transformation (4.22) modifie uniquement la tranche $\{(i, m, n) \mid (m, n) \in \mathcal{N}_s^2\}$ des $\mathbf{R}(k)$ comme l'illustre la figure 4.2. Pour $(m, n) \in \mathcal{N}_s^2$, les composantes de cette tranche prennent la forme linéaire suivante

$$R'_{1,imn}(k) = R_{imn}(k) + u_1^{ij} R_{jmn}(k). \quad (4.23)$$

FIGURE 4.2 – Schéma de l'impact de la transformation (4.22) sur un tenseur $\mathbf{R}'_1(k)$.

Finalement, on peut développer le critère (4.21) comme

$$\mathcal{J}_{(\times_1)}(\mathbf{U}_1^{ij}) = \sum_{k=1}^K \left(\sum_{(m,n) \in \mathcal{S}} |R_{imn}(k) + u_1^{ij} R_{jmn}(k)|^2 \right) + C_{tdc2} \quad (4.24)$$

où C_{tdc2} est une constante valant

$$C_{tdc2} = \sum_{k=1}^K \sum_{(l,m,n) \in \mathcal{R}} |R_{lmn}(k)|^2. \quad (4.25)$$

avec l'ensemble $\mathcal{R} = \{(l, m, n) \mid l \in \mathcal{P} \text{ et } (m, n) \in \mathcal{N}_s^2\} \setminus \{(o, o, o) \mid o \in \mathcal{P}\}$.

Nous proposons, maintenant, de minimiser (4.24) afin de déterminer le paramètre u_1^{ij} pour les différents modèles de tenseurs envisagés, à savoir non-symétrique, INDSCAL_h et symétrique.

Modèle non-symétrique : Comme tous les développements précédents de ce chapitre ont été faits dans le cadre du modèle non-symétrique (4.1), pour estimer u_1^{ij} , il suffit de minimiser (4.24). Pour cela on dérive cette expression par rapport à u_1^{ij*} de la manière suivante

$$\frac{\partial \mathcal{J}_{(\times_1)}(\mathbf{U}_1^{ij})}{\partial u_1^{ij*}} = \sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}} R_{jmn}^*(k) (R_{imn}(k) + u_1^{ij} R_{jmn}(k)). \quad (4.26)$$

En égalisant à zéro (4.26), on obtient la valeur optimale du paramètre u_1^{ij}

$$\hat{u}_1^{ij} = - \frac{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}} R_{jmn}^*(k) R_{imn}(k)}{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}} |R_{jmn}(k)|^2}. \quad (4.27)$$

Les paramètres u_2^{ij} et u_3^{ij} peuvent être calculés de manière similaire à u_1^{ij} ou simplement déduits de (4.27) par permutation d'indices. On a alors

$$\hat{u}_2^{ij} = - \frac{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}} R_{mjn}^*(k) R_{min}(k)}{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}} |R_{mjn}(k)|^2} \quad \text{et} \quad \hat{u}_3^{ij} = - \frac{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}} R_{mnj}^*(k) R_{mni}(k)}{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}} |R_{mnj}(k)|^2}. \quad (4.28)$$

Enfin les valeurs optimales des paramètres ℓ_x^{ij} , $x \in \{1, 2, 3\}$ sont déduites des expressions précédentes des \hat{u}_x^{ij} correspondants, en interchangeant les indices i et j et en sommant sur $(p, q) \in \mathcal{N}_s^2 \setminus \{(j, j)\}$. Ainsi on obtient

$$\begin{aligned} \ell_1^{ij} &= - \frac{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}'} R_{imn}^*(k) R_{jmn}(k)}{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}'} |R_{imn}(k)|^2}, \\ \ell_2^{ij} &= - \frac{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}'} R_{min}^*(k) R_{mjn}(k)}{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}'} |R_{min}(k)|^2}, \\ \ell_3^{ij} &= - \frac{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}'} R_{mni}^*(k) R_{mnj}(k)}{\sum_{k=1}^K \sum_{(m,n) \in \mathcal{S}'} |R_{mni}(k)|^2}. \end{aligned} \quad (4.29)$$

Modèle INDSCAL_h et modèle symétrique : On considère, maintenant, un ensemble de K tenseurs cibles reposant soit sur le modèle INDSCAL_h

$$\mathbf{R}(k) = \mathbf{D}(k) \times_1 \mathbf{A} \times_2 \mathbf{A} \times_3 \mathbf{A}^*, \quad (4.30)$$

soit sur le modèle symétrique

$$\mathbf{R}(k) = \mathbf{D}(k) \times_{(3)} \mathbf{A}. \quad (4.31)$$

La diagonalisation conjointe de chacun de ces ensembles repose sur l'estimation d'une matrice diagonalisante \mathbf{B} . Selon la procédure de Jacobi et la paramétrisation LU, \mathbf{B} s'écrit comme un produit de $\frac{N_s(N_s-1)}{2}$ matrices élémentaires \mathbf{L}^{ij} et $\frac{N_s(N_s-1)}{2}$ matrices élémentaires \mathbf{U}^{ij} (4.10). Par la stratégie découlée de détermination des paramètres et par l'utilisation du critère inverse selon l'approche classique, on a montré qu'une méthode pour obtenir une résolution analytique est d'estimer chacune des matrices \mathbf{L}^{ij} et \mathbf{U}^{ij} de manière indépendante. Pour un couple $(i, j) \in \mathcal{N}_s$, $i < j$, le calcul du paramètre u^{ij} associé se fait par l'optimisation du critère spécifique (4.24). Notons que la transformation engendrée par l'opérateur \times_1 est exactement la même pour les deux modèles considérés dans ce paragraphe. Ainsi, pour des ensembles de tenseurs INDSCAL_h ou symétriques, le critère (4.24) s'écrit

$$\mathcal{J}_{(\times_1)}(\mathbf{U}^{ij}) = \sum_{k=1}^K \left(\sum_{(p,q) \in \mathcal{S}} |R_{ipq}(k) + u^{ij} R_{jqp}(k)|^2 \right) + C_{tdc2}. \quad (4.32)$$

La valeur optimale de u^{ij} est l'unique racine de la dérivée de (4.32) par rapport à u^{ij} . Finalement u^{ij} vaut (4.27) et on en déduit la valeur optimale de ℓ^{ij} .

4.2.2 Approche adaptée

On considère un ensemble de K tenseurs cibles non-symétriques $\mathbf{R}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$. Dans le paragraphe 4.2, pour un couple $(i, j) \in \mathcal{N}_s^2$, $i < j$ fixé, on a vu que le calcul simultané des matrices élémentaires \mathbf{U}_x^{ij} , $x \in \{1, 2, 3\}$ pouvait se faire par la minimisation du critère $\mathcal{J}'(\mathbf{U}_x^{ij})$ (4.16). Dans le paragraphe 4.2.1, il a été expliqué que la minimisation directe de ce critère ne semblait pas permettre la résolution analytique recherchée. Ici, nous allons montrer qu'avec une approche différente, la détermination simultanée des trois matrices \mathbf{U}_x^{ij} peut se faire de façon analytique.

Tout d'abord, considérons que les tenseurs $\mathbf{R}(k)$ sont diagonaux. Cela signifie que, pour tout $k \in \mathcal{K}$, tous les éléments hors-diagonaux des $\mathbf{R}(k)$ sont nuls et si on fait l'hypothèse raisonnable que $\sum_{k=1}^K |R_{jjj}(k)|^2 \neq 0$, alors $u_1^{ij} = u_2^{ij} = u_3^{ij} = 0$ est la solution minimisant (4.16).

Ainsi, si l'on suppose que les tenseurs cibles sont proches d'être des tenseurs diagonaux, alors les composantes hors-diagonales des $\mathbf{R}(k)$ sont très petites devant un et il en est de même pour u_x^{ij} , avec $x \in \{1, 2, 3\}$. Finalement, sous cette hypothèse, pour tout $k \in \mathcal{K}$ et pour tout $(a, b, c) \in \{\mathcal{N}_s^3 \setminus \{(p, p, p) \mid p \in \mathcal{N}_s\}\}$, on a les propriétés suivantes

$$|R_{abc}(k)| \ll 1 \quad \text{et} \quad |u_x^{ij}| \ll 1. \quad (4.33)$$

Ainsi, on constate que certains termes du critère $\mathcal{J}'(\mathbf{U}_x^{ij})$ deviennent dominants. En effet, dans (4.18) et (4.19), pour tout $l \in \mathcal{P}$ et pour tout $(m, n) \in \mathcal{P}^2 \setminus \{(j, j)\}$, on a

$$\begin{aligned} |R_{iil}(k)| &\geq |R_{jil}(k)u_1^{ij} + R_{ijl}(k)u_2^{ij} + R_{jil}(k)u_1^{ij}u_2^{ij}| \\ |R_{ili}(k)| &\geq |R_{jli}(k)u_1^{ij} + R_{ilj}(k)u_3^{ij} + R_{jli}(k)u_1^{ij}u_3^{ij}| \\ |R_{lil}(k)| &\geq |R_{lji}(k)u_2^{ij} + R_{lij}(k)u_3^{ij} + R_{lji}(k)u_2^{ij}u_3^{ij}| \\ |R_{imn}(k)| &\geq |R_{jmn}(k)u_1^{ij}| \\ |R_{min}(k)| &\geq |R_{mjn}(k)u_2^{ij}| \\ |R_{mni}(k)| &\geq |R_{mnj}(k)u_3^{ij}|. \end{aligned} \quad (4.34)$$

Finalement les termes du critère (4.16) peuvent être approximés par

$$\begin{aligned} R'_{iil}(k) &\approx R_{iil}(k) \\ R'_{ili}(k) &\approx R_{ili}(k) \\ R'_{lil}(k) &\approx R_{lil}(k) \\ R'_{imn}(k) &\approx R_{imn}(k) \quad \text{et} \quad R'_{ijj}(k) = R_{ijj}(k) + R_{jjj}(k)u_1^{ij} \\ R'_{min}(k) &\approx R_{min}(k) \quad \text{et} \quad R'_{jij}(k) = R_{jij}(k) + R_{jjj}(k)u_2^{ij} \\ R'_{mni}(k) &\approx R_{mni}(k) \quad \text{et} \quad R'_{jji}(k) = R_{jji}(k) + R_{jjj}(k)u_3^{ij}. \end{aligned} \quad (4.35)$$

Ceci signifie que selon l'approche adaptée, la transformation (4.14) influe principalement sur les trois composantes situées aux positions (i, j, j) , (j, i, j) et (j, j, i) de chacun des tenseurs cibles $\mathbf{R}(k)$. La figure 4.3 révèle, pour un tenseur $\mathbf{R}'(k)$, les composantes ayant été modifiées par cette transformation dans le cadre de l'approche adaptée. Notons aussi que chacune de ces trois composantes non constantes s'exprime comme une fonction linéaire d'un seul et unique paramètre u_x^{ij} . A partir des équations (4.35), on réécrit le critère (4.16), que l'on notera pour

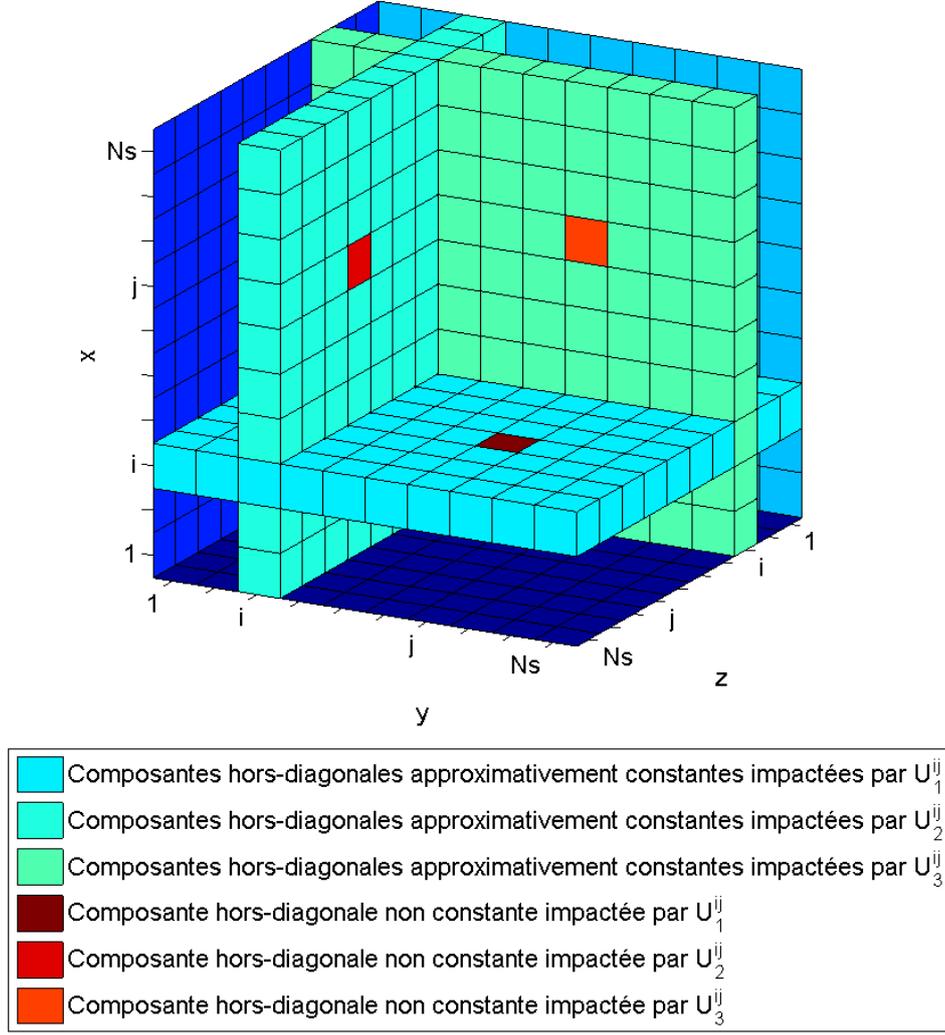


FIGURE 4.3 – Schéma de l'impact de la transformation (4.14) sur un tenseur $\mathbf{R}'(k)$ dans le cadre de l'approche adaptée.

l'approche adaptée $\mathcal{J}'_a(\mathbf{U}_x^{ij})$. On obtient

$$\mathcal{J}'_a(\mathbf{U}_x^{ij}) = \sum_{k=1}^K \left(|R'_{ijj}(k)|^2 + |R'_{jij}(k)|^2 + |R'_{jji}(k)|^2 \right) + C_{tda} \approx \mathcal{J}'(\mathbf{U}_x^{ij}) \quad (4.36)$$

avec

$$C_{tda} = \sum_{k=1}^K \left[\sum_{l \in \mathcal{P}} \left(|R'_{iil}(k)|^2 + |R'_{ili}(k)|^2 + |R'_{lii}(k)|^2 \right) + \sum_{(m,n) \in \mathcal{P}^2 \setminus \{(j,j)\}} \left(|R'_{imn}(k)|^2 + |R'_{min}(k)|^2 + |R'_{imn}(k)|^2 \right) \right]. \quad (4.37)$$

Pour les différents ensembles de tenseurs cibles envisagés, nous allons donc calculer simultanément et analytiquement les paramètres u_1^{ij} , u_2^{ij} et u_3^{ij} en minimisant le critère (4.36).

Modèle non-symétrique : Dans (4.36), le critère est écrit pour la diagonalisation conjointe d'un ensemble de tenseurs non-symétriques. De ce fait, pour déterminer u_x^{ij} pour $x \in \{1, 2, 3\}$, il suffit de minimiser directement $\mathcal{J}'_a(\mathbf{U}_x^{ij})$. Pour cela, on calcule les dérivées de (4.36) par rapport aux conjugués de chacun des u_x^{ij} . On obtient alors

$$\frac{\partial \mathcal{J}'_a(\mathbf{U}_x^{ij})}{\partial u_1^{ij*}} = \sum_{k=1}^K R_{jjj}^*(k) (R_{ijj}(k) + R_{jjj}(k)u_1^{ij}) \quad (4.38)$$

$$\frac{\partial \mathcal{J}'_a(\mathbf{U}_x^{ij})}{\partial u_2^{ij*}} = \sum_{k=1}^K R_{jjj}^*(k) (R_{jij}(k) + R_{jjj}(k)u_2^{ij}) \quad (4.39)$$

$$\frac{\partial \mathcal{J}'_a(\mathbf{U}_x^{ij})}{\partial u_3^{ij*}} = \sum_{k=1}^K R_{jjj}^*(k) (R_{jji}(k) + R_{jjj}(k)u_3^{ij}). \quad (4.40)$$

L'expression de chacune des dérivées partielles est une fonction linéaire très simple ne dépendant que d'un seul u_x^{ij} . En les égalisant chacune à zéro, on obtient analytiquement les valeurs optimales des trois paramètres

$$\hat{u}_1^{ij} = -\frac{\sum_{k=1}^K R_{jjj}^*(k)R_{ijj}(k)}{\sum_{k=1}^K |R_{jjj}(k)|^2}, \quad \hat{u}_2^{ij} = -\frac{\sum_{k=1}^K R_{jjj}^*(k)R_{jij}(k)}{\sum_{k=1}^K |R_{jjj}(k)|^2} \quad \text{et} \quad \hat{u}_3^{ij} = -\frac{\sum_{k=1}^K R_{jjj}^*(k)R_{jji}(k)}{\sum_{k=1}^K |R_{jjj}(k)|^2}. \quad (4.41)$$

Les valeurs optimales $\hat{\ell}_x^{ij}$ sont, bien entendu, déduites de (4.41) en remplaçant tous les indices i par des j et inversement.

Modèle INDSCAL_h : Nous considérons maintenant un ensemble de K tenseurs cibles $\mathbf{R}^{(h)}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ admettant la décomposition INDSCAL_h (4.30). Un tel modèle présente des propriétés de symétrie. En effet, pour tout $(a, b) \in \mathcal{N}_s^2$ on a

$$R_{abc}^{(h)}(k) = R_{bac}^{(h)}(k). \quad (4.42)$$

Le modèle non-symétrique correspond exactement au modèle INDSCAL_h lorsqu'on pose $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}_3^* = \mathbf{A}$. La diagonalisation conjointe de cet ensemble de tenseurs se fait donc comme pour le modèle non-symétrique, en estimant une matrice $\mathbf{B} = \mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B}_3^*$ admettant la décomposition (4.10). Ainsi, pour déterminer le paramètre u^{ij} , il s'agit de minimiser le critère

(4.36), dans lequel on remplace u_1^{ij} et u_2^{ij} par u^{ij} et u_3^{ij} par u^{ij*} . De plus, en prenant en compte la propriété de symétrie (4.42), le critère que l'on dénote $\mathcal{J}_a^{(h)}$ s'écrit

$$\mathcal{J}_a^{(I)}(\mathbf{U}^{ij}) = \sum_{k=1}^K \left(2 | R_{ijj}^{(h)}(k) + R_{jjj}^{(h)}(k)u^{ij} |^2 + | R_{jji}^{(h)}(k) + R_{jjj}^{(h)}(k)u^{ij*} |^2 \right). \quad (4.43)$$

On minimise ensuite (4.43), en le dérivant par rapport à u^{ij*} comme il suit

$$\frac{\partial \mathcal{J}_a^{(I)}(\mathbf{U}^{ij})}{\partial u^{ij*}} = \sum_{k=1}^K \left(2R_{jjj}^{(h)*}(k) \left(R_{ijj}^{(h)}(k) + R_{jjj}^{(h)}(k)u^{ij} \right) + R_{jji}^{(h)}(k) \left(R_{jji}^{(h)*}(k) + R_{jjj}^{(h)*}(k)u^{ij} \right) \right). \quad (4.44)$$

Et finalement, on obtient la valeur optimale

$$\hat{u}^{ij} = - \frac{\sum_{k=1}^K \left(2R_{jjj}^{(h)*}(k)R_{ijj}^{(h)}(k) + R_{jji}^{(h)}(k)R_{jji}^{(h)*}(k) \right)}{3 \sum_{k=1}^K | R_{jjj}^{(h)}(k) |^2}. \quad (4.45)$$

Modèle symétrique : On s'intéresse, enfin, à un ensemble de K tenseurs symétriques $\mathbf{R}^{(s)}(k) \in \mathbb{C}^{N_s \times N_s \times N_s}$ reposant sur le modèle simplifié (4.31). Les tenseurs $\mathbf{R}^{(s)}(k)$ sont symétriques, ce qui signifie que pour tout $(a, b, c) \in \mathcal{N}_s^3$ on a

$$R_{abc}^{(s)}(k) = R_{acb}^{(s)}(k) = R_{bac}^{(s)}(k) = R_{bca}^{(s)}(k) = R_{cab}^{(s)}(k) = R_{cba}^{(s)}(k). \quad (4.46)$$

Pour diagonaliser cet ensemble, on cherche à déterminer une matrice $\mathbf{B} = \mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B}_3$. Ainsi on utilise directement l'expression du critère pour le modèle non-symétrique (4.36), en prenant en compte les symétries (4.46) et en remplaçant les paramètres u_x^{ij} , $x \in \{1, 2, 3\}$ par u^{ij} . Le nouveau critère, dénoté ici $\mathcal{J}_a^{(s)}$ s'écrit

$$\mathcal{J}_a^{(s)}(\mathbf{U}^{ij}) = 3 \sum_{k=1}^K | R_{ijj}^{(s)}(k) + R_{jjj}^{(s)}(k)u^{ij} |^2. \quad (4.47)$$

Sa dérivée par rapport à u^{ij*} vaut

$$\frac{\partial \mathcal{J}_a^{(I)}(\mathbf{U}^{ij})}{\partial u^{ij*}} = 3 \sum_{k=1}^K R_{jjj}^{(s)*}(k) \left(R_{ijj}^{(s)}(k) + R_{jjj}^{(s)}(k)u^{ij} \right). \quad (4.48)$$

Enfin, en égalisant (4.48) à zéro, on obtient la valeur optimale

$$\hat{u}^{ij} = - \frac{\sum_{k=1}^K R_{jjj}^{(s)*}(k) R_{ijj}^{(s)}(k)}{\sum_{k=1}^K |R_{jjj}^{(s)}(k)|^2}. \quad (4.49)$$

Algorithme 5 T-ALUJA

- 1: **input** : $\mathbf{T}(k)$ respectant un des modèles considérés
 - 2: Initialiser $\mathbf{B}_x \leftarrow \mathbf{B}_{x_0}$ de dimensions $N_s \times N_o$
 - 3: Mettre à jour les matrices $\mathbf{T}(k)$ suivant (4.4)
 - 4: **repeat**
 - 5: **for** $1 \leq i < j \leq N_s$ **do** ▷ Etape U
 - 6: **for** $1 \leq x \leq 3$ **do** ▷ pour le modèle non-symétrique
 - 7: Calculer u_x^{ij} en utilisant (4.27) ou (4.28)
 - 8: Construire \mathbf{U}_x^{ij}
 - 9: Mettre à jour $\mathbf{R}(k) \leftarrow \mathbf{R}(k) \times_x \mathbf{U}_x^{ij}$
 - 10: Mettre à jour $\mathbf{B}_x \leftarrow \mathbf{U}_x^{ij} \mathbf{B}_x$
 - 11: **end for**
 - 12: **end for**
 - 13: **for** $1 \leq i < j \leq N_s$ **do** ▷ Etape L
 - 14: **for** $1 \leq x \leq 3$ **do** ▷ pour le modèle non-symétrique
 - 15: Calculer ℓ_x^{ij} en interchangeant les indices i et j dans (4.27) ou (4.28)
 - 16: Construire \mathbf{L}_x^{ij}
 - 17: Mettre à jour $\mathbf{R}(k) \leftarrow \mathbf{R}(k) \times_x \mathbf{L}_x^{ij}$
 - 18: Mettre à jour $\mathbf{B}_x \leftarrow \mathbf{L}_x^{ij} \mathbf{B}_x$
 - 19: **end for**
 - 20: **end for**
 - 21: **until** convergence
 - 22: **output** : $\mathbf{B}_1, \mathbf{B}_2$ et \mathbf{B}_3
-

4.2.3 Description des algorithmes

Dans ce paragraphe, nous décrivons les algorithmes résultant des deux approches proposées au moyen de leurs pseudo-codes. L'algorithme 5 développé dans le cadre de l'approche classique est dénommé T-ALUJA (pour *Tensorial ALUJA*), tandis que l'algorithme 6 développé pour l'approche adaptée est dénommé T-ALUJA-A (pour *T-ALUJA for an Adapted approach*).

Algorithme 6 T-ALUJA-A

```

1: input :  $\mathbf{T}(k)$  respectant un des modèles considérés

2: Initialiser  $\mathbf{B}_x \leftarrow \mathbf{B}_{x_0}$  de dimensions  $N_s \times N_o$ 
3: Mettre à jour les tenseurs  $\mathbf{T}(k)$  suivant (4.4)
4: repeat
5:   for  $1 \leq i < j \leq N_s$  do ▷ Etape U
6:     Calculer  $u_1^{ij}$ ,  $u_2^{ij}$  et  $u_3^{ij}$  en utilisant (4.41) ou (4.45) ou (4.49)
7:     Construire  $\mathbf{U}_1^{ij}$ ,  $\mathbf{U}_2^{ij}$  et  $\mathbf{U}_3^{ij}$ 
8:     Mettre à jour  $\mathbf{R}(k) \leftarrow \mathbf{R}(k) \times_1 \mathbf{U}_1^{ij} \times_2 \mathbf{U}_2^{ij} \times_3 \mathbf{U}_3^{ij}$ 
9:     Mettre à jour  $\mathbf{B}_1 \leftarrow \mathbf{U}_1^{ij} \mathbf{B}_1$ ,  $\mathbf{B}_2 \leftarrow \mathbf{U}_2^{ij} \mathbf{B}_2$  et  $\mathbf{B}_3 \leftarrow \mathbf{U}_3^{ij} \mathbf{B}_3$ 
10:  end for
11:  for  $1 \leq i < j \leq N_s$  do ▷ Etape L
12:    Calculer  $\ell_1^{ij}$ ,  $\ell_2^{ij}$  et  $\ell_3^{ij}$  en interchangeant les indices  $i$  et  $j$  dans (4.41) ou (4.45) ou
    (4.49)
13:    Construire  $\mathbf{L}_1^{ij}$ ,  $\mathbf{L}_2^{ij}$  et  $\mathbf{L}_3^{ij}$ 
14:    Mettre à jour  $\mathbf{R}(k) \leftarrow \mathbf{R}(k) \times_1 \mathbf{L}_1^{ij} \times_2 \mathbf{L}_2^{ij} \times_3 \mathbf{L}_3^{ij}$ 
15:    Mettre à jour  $\mathbf{B}_1 \leftarrow \mathbf{L}_1^{ij} \mathbf{B}_1$ ,  $\mathbf{B}_2 \leftarrow \mathbf{L}_2^{ij} \mathbf{B}_2$  et  $\mathbf{B}_3 \leftarrow \mathbf{L}_3^{ij} \mathbf{B}_3$ 
16:  end for
17: until convergence

18: output :  $\mathbf{B}_1$ ,  $\mathbf{B}_2$  et  $\mathbf{B}_3$ 

```

Remarque : Il est important de signaler qu'en adaptant l'ordre de balayage des matrices \mathbf{U}^{ij} et \mathbf{L}^{ij} , il est tout à fait possible de paralléliser le calcul des paramètres ainsi que les mises à jour des tenseurs cibles et des matrices diagonalisantes. De ce fait, pour un $x \in \{1, 2, 3\}$ et pour un $i \in \mathcal{N}_s$ donnés, il est possible de déterminer indépendamment les $N_s - 1$ paramètres u^{ji} avec $i > j$ et ℓ^{ij} avec $i < j$ et de faire toutes les mises à jour nécessaires tout aussi indépendamment.

4.3 Simulations

Dans ce paragraphe, nous illustrons le comportement des deux algorithmes proposés : T-ALUJA et T-ALUJA-A. Dans les paragraphes précédents nous avons pu voir que pour chacun des deux algorithmes, le calcul des paramètres se fait de manière semblable, quelle que soit la nature des ensembles de tenseurs envisagés. De plus, les tests en simulations numériques pour chacun des algorithmes, nous ont montré un comportement similaire, que l'on considère un ensemble de tenseurs non-symétriques (4.1), symétriques (4.31) ou admettant la décomposition INDSCAL_h (4.30). Afin d'éviter la redondance des simulations, nous nous concentrerons, ici, sur le cas des tenseurs non-symétriques et nous comparerons ainsi

T-ALUJA et T-ALUJA-A à des algorithmes de décomposition CP de la littérature. Les deux autres modèles de tenseurs seront spécifiquement utilisés au chapitre 5, dans le cadre de l'application aux signaux de télécommunications numériques.

4.3.1 Algorithmes de décomposition CP

Nous proposons de comparer les algorithmes de diagonalisation conjointe non-unitaire de tenseurs d'ordre trois développés dans cette thèse, à trois algorithmes de décomposition CP d'un tenseur d'ordre quatre développés par Sorber *et al.* dans la toolbox Matlab tensorlab [107]. Rappelons que le lien entre l'ensemble de tenseurs d'ordre trois et le tenseur d'ordre quatre considéré est montré à la fin du paragraphe 2.2.1. Les algorithmes de décomposition CP sont basés sur des méthodes de moindres carrés (ce qui signifie qu'ils reposent sur le critère direct (2.74)). Les trois algorithmes considérés sont les suivants :

- Les moindres carrés alternés ou ALS pour *Alternating Least Square*, développé et utilisé initialement dans le cadre de la décomposition CP par Carroll et Chang [16] et Harshman [57]. Cet algorithme itératif, très simple, est une référence dans le domaine. L'ALS consiste ici à estimer tour à tour, une des quatre matrices facteurs, en minimisant au sens des moindres carrés l'erreur quadratique entre les tenseurs et leur modèle. Bien que la convergence ne soit pas systématiquement assurée avec l'ALS, l'algorithme atteint en général un bon niveau de performance et présente une complexité algorithmique relativement faible. Cependant, sa convergence est assez lente. Pour notre comparaison, nous utiliserons une version améliorée de l'ALS avec une recherche linéaire optimisée, aussi appelée ELS pour *Enhanced Line Search* [98, 105] dont le but est de rechercher le pas d'adaptation optimal dans la direction considérée pour garantir la convergence de l'algorithme au prix d'une complexité algorithmique accrue. Dans les simulations, on dénotera cet algorithme CPD-ALS.
- Le gradient conjugué non-linéaire, noté NCG pour *Non-linear Conjugate Gradient*, dont la version utilisée ici a été développée par Sorber *et al.* dans [104]. Cette méthode itérative associe des étapes de descente du gradient selon plusieurs directions. Ainsi la complexité de l'algorithme dépendra du nombre d'itérations internes qui est lié aux différentes étapes de descente du gradient. NCG présente une meilleure vitesse de convergence qu'une simple méthode du gradient. Pour l'algorithme considéré, le pas d'adaptation du gradient est calculé de façon optimale à l'aide d'une méthode de Moré-Thuente. Pour les simulations, on dénommera l'algorithme CPD-NCG.
- L'algorithme Levenberg-Marquardt (LM), développé initialement par Marquardt [76], est basé sur la méthode proposée, quelques années auparavant, par Levenberg [68].

Cet algorithme itératif combine la méthode de Gauss-Newton et celle du gradient. Contrairement à la méthode de Gauss-Newton, LM converge vers une solution même s'il est mal initialisé. LM repose essentiellement sur l'estimation et l'inversion de la matrice Jacobienne associée à un critère des moindres carrés plus un terme de régularisation permettant d'optimiser la recherche de direction. Cette méthode se place parmi les meilleurs algorithmes de décomposition CP en termes de niveau de performance et une vitesse de convergence. Le principal inconvénient de cette méthode des moindres carrés est sa complexité algorithmique bien supérieure à celle des autres algorithmes de décomposition CP, due à l'inversion de la matrice Jacobienne. Cependant, la méthode LM utilisée dans le cadre de nos simulations [106] exploite la structure particulière de la matrice Jacobienne, permettant ainsi de réduire la complexité de l'algorithme. De plus, elle utilise un préconditionneur basé sur une méthode de gradient conjugué qui permet d'éviter d'éventuels problèmes dus à l'inversion de matrices.

4.3.2 Complexité algorithmique

Avant toute chose, intéressons nous à la complexité par balayage (ou par itération) des différents algorithmes que nous allons comparer. Les complexités avancées ici sont celle obtenues pour des tenseurs réels. Pour le cas complexe, il suffit de multiplier tous les termes par un facteur quatre. Pour les trois algorithmes de décomposition CP, les calculs des complexités pour des tenseurs de tout ordre, sont détaillés dans [106]. Les complexités sont données en flop. Pour que les comparaisons soient justes, tous les algorithmes traiteront K tenseurs cibles réduits de dimensions $N_s \times N_s \times N_s$ (soit un tenseur de dimensions $N_s \times N_s \times N_s \times K$ pour les algorithmes de décomposition CP). Ainsi, dans ces conditions, l'ALS classique présente une complexité par itération de $\mathcal{O}(10KN_s^4)$ flop et lorsqu'on l'optimise avec l'ELS, on obtient finalement une complexité de l'ordre de $\mathcal{O}(16KN_s^4)$ flop par itération. Pour NCG, la complexité va dépendre, en plus des caractéristiques du problème, du nombre d'itérations de la méthode de Moré-Thuente it_{mt} pour calculer de manière optimale le pas d'adaptation. Ainsi la complexité de NCG dépendra, essentiellement, du calcul du gradient et de la fonction d'évaluation. Finalement, il vaut $\mathcal{O}(10it_{mt}KN_s^4)$ flop par itération. Enfin, soit it_{lm} le nombre d'itérations internes à l'algorithme LM, et soit it_{cg} le nombre d'itérations nécessaires pour le calcul du préconditionneur, la complexité de l'algorithme LM approximé utilisé ici est de l'ordre de $\mathcal{O}(2(4 + it_{lm})KN_s^4 + it_{lm}it_{cg}(40N_s^2 + \frac{4}{3}N_s^3 + 8N_s^2(K + 3N_s)))$ flop par itération. Notons la grande diminution de complexité par rapport à l'algorithme LM non optimisé qui ne nécessite pas moins de $\mathcal{O}(2(4 + it_{lm})KN_s^4 + it_{lm}\frac{1}{3}N_s^3(K + 3N_s)^3)$ flop par itération.

Détaillons, maintenant, le coût des algorithmes de diagonalisation conjointe de tenseurs d'ordre trois développés au cours de cette thèse. Commençons par le calcul des différents paramètres. Pour T-ALUJA, le calcul d'un seul paramètre nécessite exactement $2K(N_s^2 - 1) + 1$ flop contre seulement $2K + 1$ flop pour T-ALUJA-A. Pour un couple $(i, j) \in \mathcal{N}_s^2$, $i < j$ donné, il y a six paramètres à estimer. Donc, pour un balayage des $\frac{N_s(N_s - 1)}{2}$ couples (i, j) possibles, où un balayage correspond à une itération pour les algorithmes de décomposition CP, l'estimation de tous les paramètres coûte $\mathcal{O}(6KN_s^4)$ flop pour T-ALUJA et $\mathcal{O}(3(2K + 1)N_s^2)$ flop pour T-ALUJA-A. Concernant la mise à jour des tenseurs cibles et des matrices diagonalisantes, la complexité est la même pour les deux algorithmes. Après avoir déterminé un paramètre, on met à jour une tranche de N_s^2 composantes pour un tenseur cible, soit KN_s^2 composantes pour l'ensemble des tenseurs. A chaque estimation d'un paramètre, on met aussi à jour N_s éléments d'une matrice diagonalisante (cette opération est clairement négligeable devant la mise à jour des tenseurs cibles). Au cours d'un balayage, ces deux opérations sont faites pour chacun des $3N_s(N_s - 1)$ paramètres. Ainsi, les mises à jour ont un coût global de $\mathcal{O}(3KN_s^4)$ flop par itération. Finalement, la complexité par balayage des algorithmes proposés est de $\mathcal{O}(9KN_s^4)$ flop pour T-ALUJA et de $\mathcal{O}(3KN_s^4)$ flop pour T-ALUJA-A dont le coût d'estimation des paramètres est négligeable devant le coût des mises à jour.

Algorithmes	Complexité (en flop/itération)
T-ALUJA	$\mathcal{O}(9KN_s^4)$
T-ALUJA-A	$\mathcal{O}(3KN_s^4)$
CPD-ALS	$\mathcal{O}(16KN_s^4)$
CPD-NCG	$\mathcal{O}(10it_{\text{mt}}KN_s^4)$
CPD-LM	$\mathcal{O}\left(2(4 + it_{\text{lm}})KN_s^4 + it_{\text{lm}}it_{\text{cg}}(40N_s^2 + \frac{4}{3}N_s^3 + 8N_s^2(K + 3N_s))\right)$

TABLE 4.2 – Récapitulatif des complexités de chacun des algorithmes de diagonalisation conjointe de tenseurs proposés et de chacun des algorithmes de décomposition CP utilisés dans ce chapitre

Le tableau 4.2 récapitule les complexités de chaque algorithme. On constate que l'algorithme T-ALUJA-A est celui présentant la complexité la plus faible devant T-ALUJA puis CPD-ALS. CPD-NCG et CPD-LM, quant à eux, dépendent du nombre d'itérations interne aux méthodes utilisées. Cependant, même dans le meilleur des cas où $it_{\text{mt}} = it_{\text{lm}} = it_{\text{cg}} = 1$, leurs complexités ne descendraient pas en dessous de $\mathcal{O}(10KN_s^4)$ flop par itération, soit la même complexité que l'ALS non optimisé. Finalement, aucun des algorithmes de décomposition CP présentés ne possède une meilleure complexité algorithmique que T-ALUJA et surtout T-ALUJA-A.

4.3.3 Cadre de simulations

Afin d'évaluer les performances des différents algorithmes, nous utilisons l'indice suggéré dans [80, 85, 84], déjà utilisé dans le chapitre précédent dans le cas matriciel (3.116). Dans le cas tensoriel considéré, cet indice s'écrit pour tout $x \in \{1, 2, 3\}$

$$I(\mathbf{S}_x) = \frac{1}{2N_s(N_s - 1)} \left(\sum_{i=1}^{N_s} \left(\sum_{j=1}^{N_s} \frac{|S_{x,ij}|^2}{\max_m |S_{x,im}|^2} - 1 \right) + \sum_{j=1}^{N_s} \left(\sum_{i=1}^{N_s} \frac{|S_{x,ij}|^2}{\max_m |S_{x,mj}|^2} - 1 \right) \right), \quad (4.50)$$

où $\mathbf{S}_x = (S_{x,ij}) = \mathbf{B}_x \mathbf{A}_x$ pour tout $(i, j) \in \mathcal{N}_s^2$. Ici l'indice de performance (4.50) va être calculé pour chacune des trois matrices diagonalisantes \mathbf{B}_x estimées par les algorithmes T-ALUJA et T-ALUJA-A. Finalement, nous retiendrons la valeur moyenne I de ces trois indices de performance telle que

$$I = \frac{1}{3} \sum_{x=1}^3 I(\mathbf{S}_x). \quad (4.51)$$

Pour les algorithmes de décomposition CP d'un tenseur d'ordre quatre, le protocole est un peu différent. En effet, ces algorithmes basés sur des méthodes de moindres carrés n'estiment pas les matrices diagonalisantes mais directement les quatre matrices facteurs $\widehat{\mathbf{A}}_{x'}$, $x' = \{1, 2, 3, 4\}$. Afin de pouvoir mesurer les performances des algorithmes CPD-ALS, CPD-NCG et CPD-LM et surtout de les comparer avec T-ALUJA et T-ALUJA-A, nous ne nous intéresserons qu'aux trois premières matrices estimées et nous calculerons les matrices globales \mathbf{S}_x , de telle sorte que $\mathbf{S}_x = (\widehat{\mathbf{A}}_x)^\dagger \mathbf{A}_x$.

Pour toutes les simulations, on considère K tenseurs synthétiques bruités reposant sur le modèle suivant

$$\mathbf{T}(k) = \mathbf{D}(k) \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \mathbf{A}_3 + t\mathbf{N}(k), \quad (4.52)$$

où \mathbf{N}_k est un bruit additif gaussien et t le niveau de bruit. Chacun des K tenseurs cibles d'ordre trois est rangé dans une quatrième dimension, formant ainsi un tenseur d'ordre quatre adéquat pour une décomposition CP. Pour le modèle (4.52), les différents tenseurs d'ordre deux et trois sont générés de manière aléatoire. Les parties réelles et imaginaires des trois matrices facteurs \mathbf{A}_x et des tenseurs diagonaux $\mathbf{D}(k)$ de dimensions respectives $N_{o_x} \times N_s$ et $N_s \times N_s \times N_s$ suivent une loi uniforme dans l'intervalle $[-1, 1]$. Les parties réelles et imaginaires des tenseurs $\mathbf{N}(k)$ et des matrices d'initialisation \mathbf{B}_{x_0} , $x \in \{1, 2, 3\}$ de dimensions respectives $N_{o_1} \times N_{o_2} \times N_{o_3}$ et $N_s \times N_{o_x}$ sont tirées selon une loi normale centrée réduite. Pour les algorithmes CPD, on tire aussi une quatrième matrice d'initialisation \mathbf{B}_{4_0} de dimensions $N_s \times K$ tirée aléatoirement selon la même loi que les trois autres matrices d'initialisation.

Au moyen de simulations de type Monte-Carlo (n_{draws} tirages indépendants), nous étu-

dions le comportement des différents algorithmes dans divers contextes. Ainsi, nous ferons varier l'un après l'autre ces trois paramètres : le niveau de bruit t , le nombre de tenseurs K et la dimension N_s des tenseurs réduits. Pour la diagonalisation conjointe de ces ensembles de tenseurs synthétiques respectant le modèle exact, on s'intéresse au niveau atteint par l'indice de performance qui atteste de la précision d'un algorithme. L'indice de performance sera considéré comme mauvais s'il est supérieur à -20dB , moyen s'il est compris entre -20dB et -30dB et bon s'il est inférieur à -30dB . Le nombre de balayages nécessaires pour qu'un algorithme converge (dans le sens où, à partir d'un certain nombre de balayages, l'indice de performance n'évolue plus) est appelé vitesse de convergence. Ici, le niveau de performance atteint par l'indice n'entre pas en compte. Si un algorithme met moins de 30 balayages à converger, on considèrera qu'il présente une bonne vitesse convergence, tandis qu'entre 30 et 100 balayages elle sera qualifiée de moyenne et mauvaise si l'algorithme met plus de 100 balayages à converger ou qu'il diverge. Enfin, on étudie aussi, le pourcentage de tirages pour lequel un algorithme converge en moins de 500 itérations (que l'on abrègera par pourcentage de convergence). Cela peut donner une idée de la robustesse d'un algorithme. Ce pourcentage est considéré comme bon s'il vaut 100%, moyen s'il est compris en 100% et 90% et mauvais s'il est inférieur à 90%.

Figure	$[N_{o_1}, N_{o_2}, N_{o_3}]$	N_s	K	t	affichage
4.4	[12, 15, 20]	10	10	0	tous
4.5	[12, 15, 20]	10	10	0	méd. et moy.
4.6	[12, 15, 20]	10	10	10^{-4}	tous
4.7	[12, 15, 20]	10	10	10^{-4}	méd. et moy.
4.8	[12, 15, 20]	10	10	10^{-2}	tous
4.9	[12, 15, 20]	10	10	10^{-2}	méd. et moy.
4.10	[12, 15, 20]	10	10	$\{10^{-7}, \dots, 10^{-1}\}$	médiane
4.11	[12, 15, 20]	10	10	$\{10^{-7}, \dots, 10^{-1}\}$	moyenne
4.12	[12, 15, 20]	10	$\{1, \dots, 30\}$	10^{-4}	médiane
4.13	[12, 15, 20]	10	$\{1, \dots, 30\}$	10^{-4}	moyenne
4.14	[12, 15, 20]	10	1	10^{-4}	tous
4.15	[12, 15, 20]	10	1	10^{-4}	méd. et moy.
4.16	[32, 35, 40]	$\{3, \dots, 30\}$	10	10^{-4}	médiane
4.17	[32, 35, 40]	30	10	10^{-4}	tous
4.18	[32, 35, 40]	30	10	10^{-4}	méd. et moy.

TABLE 4.3 – Paramètres de simulation pour les algorithmes de diagonalisation conjointe d'ensembles de tenseurs complexes non-symétriques.

Comme les trois algorithmes de décomposition CP sont programmés par défaut pour s'arrêter dans le pire des cas au bout de 500 itérations, nous avons décidé, pour rendre la

comparaison équitable, de prendre ce critère comme seul et unique critère d'arrêt. De ce fait, toutes les simulations de cette section sont réalisées pour 500 itérations (ou balayages) pour chacun des algorithmes. Comme expliqué dans le paragraphe 3.4, nous afficherons soit les performances pour l'intégralité des n_{draws} tirages, soit la performance moyenne, soit la performance médiane. Pour toutes les simulations nous fixerons $n_{\text{draws}} = 50$ et comme dit précédemment $n_{\text{sweeps}} = 500$. Le tableau 4.3 résume les paramètres de simulations pour chacune des figures de ce paragraphe.

4.3.4 Influence des différents paramètres

Maintenant, étudions, respectivement, l'influence des paramètres t , K et N_s sur les algorithmes considérés.

Influence du niveau de bruit : Pour commencer, les figures 4.4 et 4.5 montrent le comportement des différents algorithmes, pour des ensembles de 10 tenseurs non-symétriques $12 \times 12 \times 20$ avec N_s égal à 10 et sans bruit additif. Pour tous les algorithmes, l'indice de performance médian est nul après convergence et seul l'indice de CPD-NCG n'atteint pas ce niveau de performance en moyenne. Ainsi T-ALUJA, T-ALUJA-T, CPD-ALS et CPD-LM diagonalisent parfaitement l'intégralité des ensembles de tenseurs. Notons que l'initialisation aléatoire de \mathbf{B}_{1o} , \mathbf{B}_{2o} et \mathbf{B}_{3o} n'assure pas forcément d'être proche d'une solution diagonalisante, néanmoins T-ALUJA-A converge pour tous les tirages en moins de 6 itérations. T-ALUJA-A présente, en valeur médiane, la meilleure vitesse de convergence devant respectivement CPD-ALS, CPD-LM, T-ALUJA et CPD-NCG. Il est important de remarquer que, même si CPD-ALS fournit de bonnes performances médianes, il peut parfois s'avérer lent et mettre plus de 350 itérations pour converger. Enfin, relevons que, dans ce contexte non bruité, CPD-NCG affiche une mauvaise vitesse de convergence qui ne lui permet parfois pas de converger en moins de 500 itérations.

Maintenant, voyons l'effet qu'un bruit blanc gaussien additif, appliqué à ces mêmes ensembles de tenseurs, peut avoir sur les différents algorithmes. Pour les figures 4.6 et 4.7, le niveau de bruit est de 10^{-4} tandis que pour les figures 4.8 et 4.9, t vaut 10^{-2} . En ce qui concerne le niveau de performance atteint, les trois algorithmes CPD qui atteignent environ le même très bon niveau de performance, surpassent les deux algorithmes proposés. De plus, les algorithmes CPD révèlent une dispersion très faible par rapport au niveau de performance atteint après convergence. Par exemple, sur la figure 4.8, dès la deux-centième itération, CPD-LM affiche, pour tous les ensembles de tenseurs décomposés, des performances comprises entre -79dB et -92dB . Cependant, même si T-ALUJA-A et T-ALUJA n'affichent pas d'aussi bons niveaux de performance que les algorithmes CPD, leurs indices de performance moyens, pour

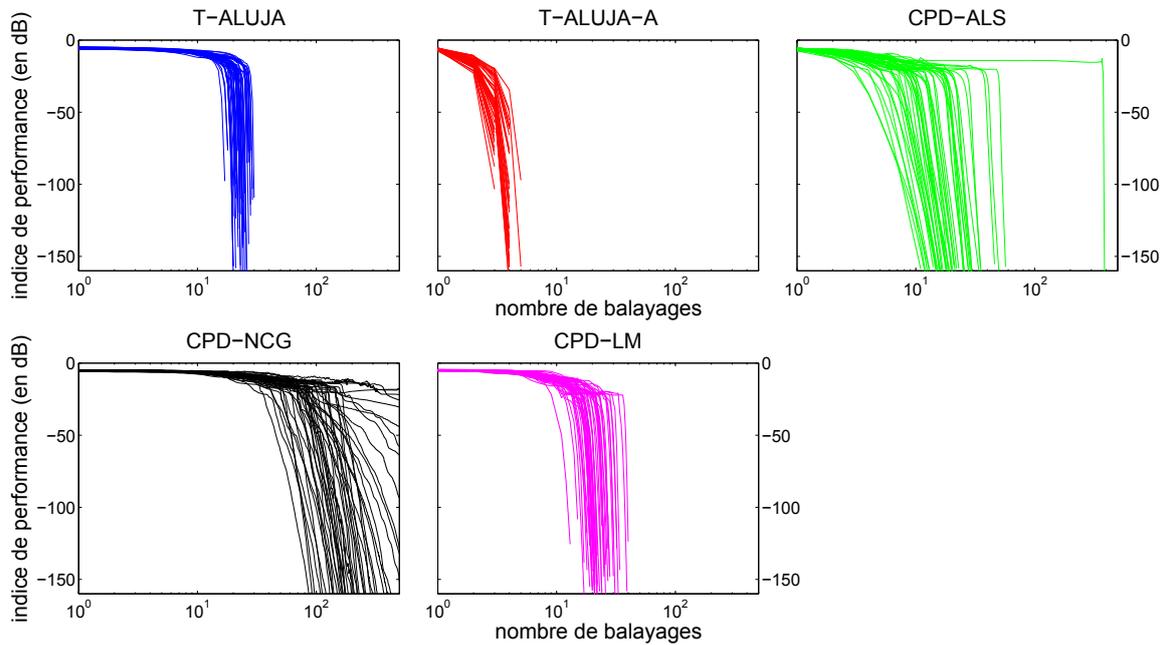


FIGURE 4.4 – Indice de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte non bruité.

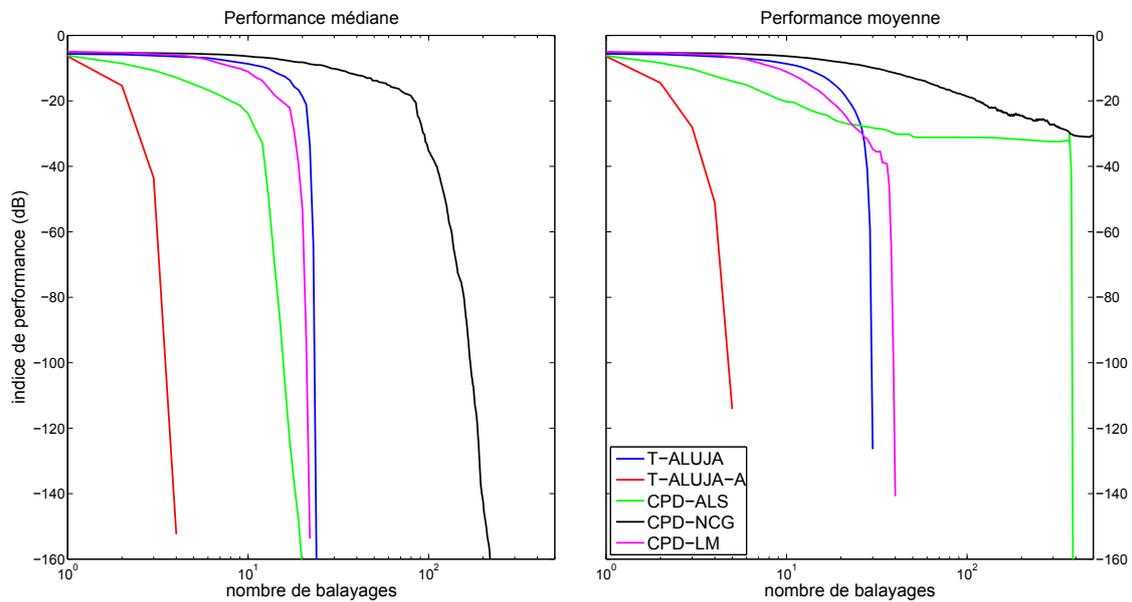


FIGURE 4.5 – Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte non bruité et pour 50 tirages indépendants.

la figure 4.9, reste inférieur à -20dB pour T-ALUJA et inférieur à -45dB pour T-ALUJA-A. De plus, T-ALUJA-A présente, encore une fois, la meilleure vitesse de convergence. On note que, comme dans le cas non bruité, CPD-ALS peut converger en moins de 10 itérations pour certains tirages, comme en plus de 300 itérations pour d'autres. Ces simulations confirment aussi la faible vitesse de convergence de CPD-NCG, ce qui entraîne un mauvais pourcentage de convergence en moins de 500 itérations. Finalement, suivant l'application visée, si on a besoin d'un résultat très précis, on se tournera vers les algorithmes CPD, mais si on désire obtenir des résultats rapidement, même s'ils sont moins précis, alors T-ALUJA-A semble être une bonne alternative.

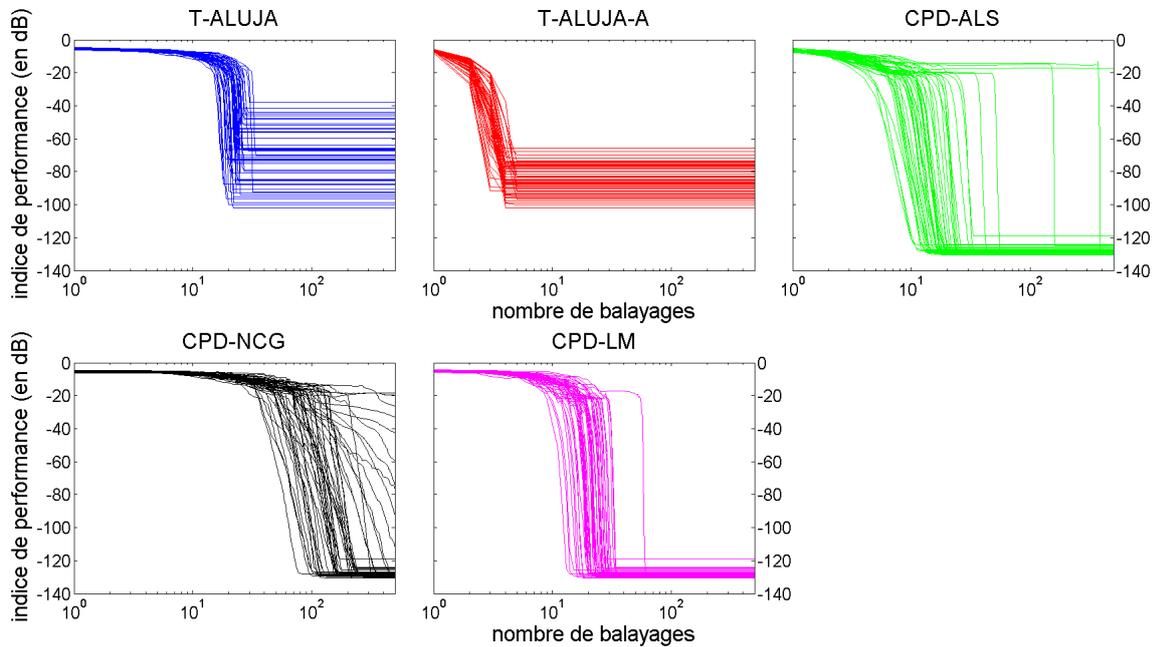


FIGURE 4.6 – Indice de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$.

Enfin, les figures 4.10 et 4.11 illustrent, respectivement, les performances médianes et moyennes atteintes par les différents algorithmes après 500 itérations en fonction du niveau de bruit. On constate une influence linéaire du niveau de bruit sur les performances médianes des algorithmes CPD et T-ALUJA-A. T-ALUJA qui atteint les mêmes niveaux de performance que T-ALUJA-A quand $t < 10^{-5}$, devient moins performant quand t augmente. Les remarques faites précédemment sur la différence de niveau de performance entre T-ALUJA et les trois algorithmes CPD se confirment pour tout $t \neq 0$. Les performances moyennes mettent en relief le fait que CPD-ALS et CPD-NCG ne présentent pas toujours 100% de convergence en moins de 500 itérations. C'est ici le cas de CPD-NCG pour tout t et de CPD-ALS et pour $t < 10^{-2}$.

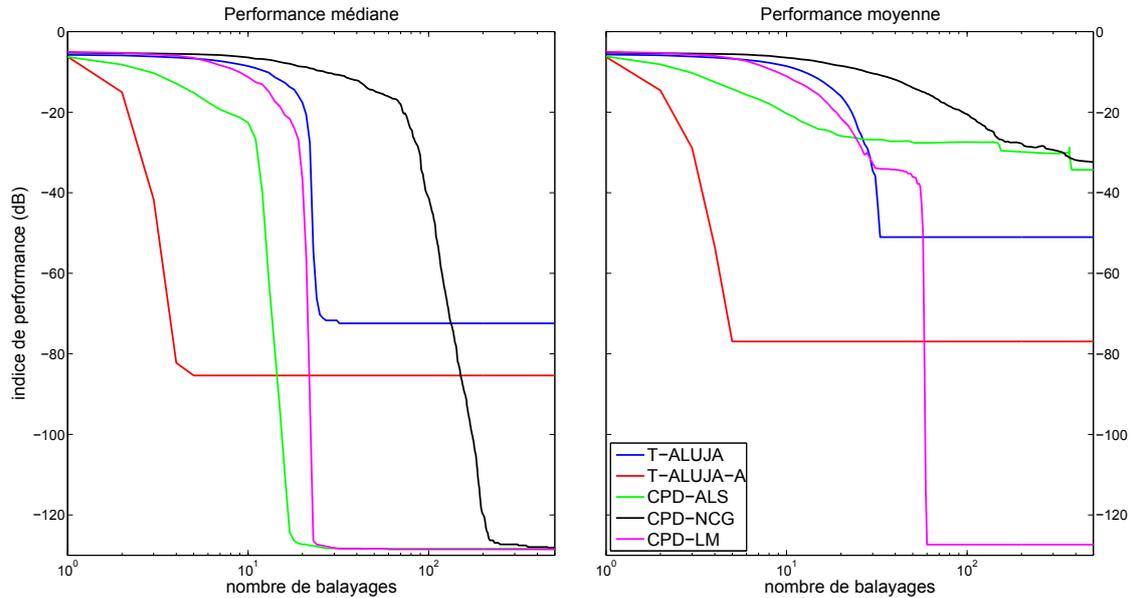


FIGURE 4.7 – Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.

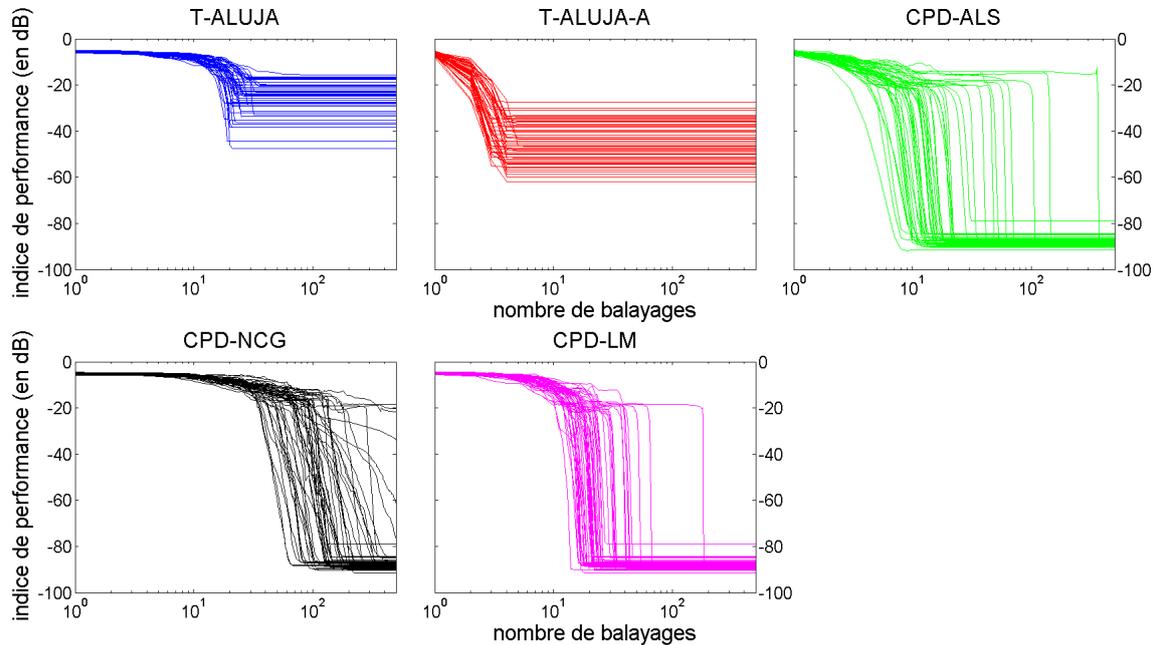


FIGURE 4.8 – Indice de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-2}$.

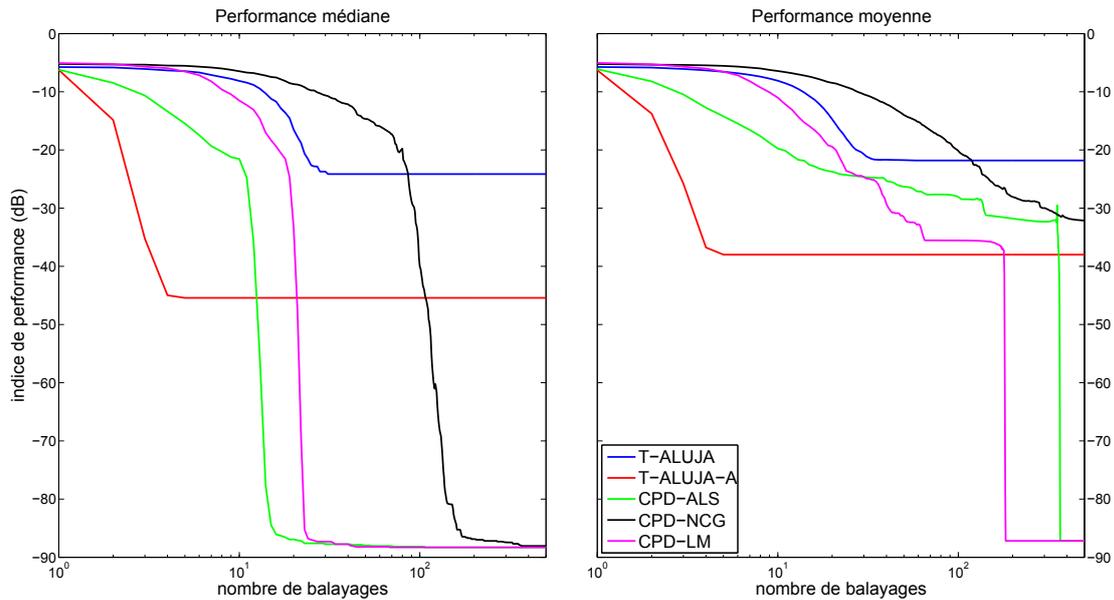


FIGURE 4.9 – Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-2}$ et pour 50 tirages indépendants.

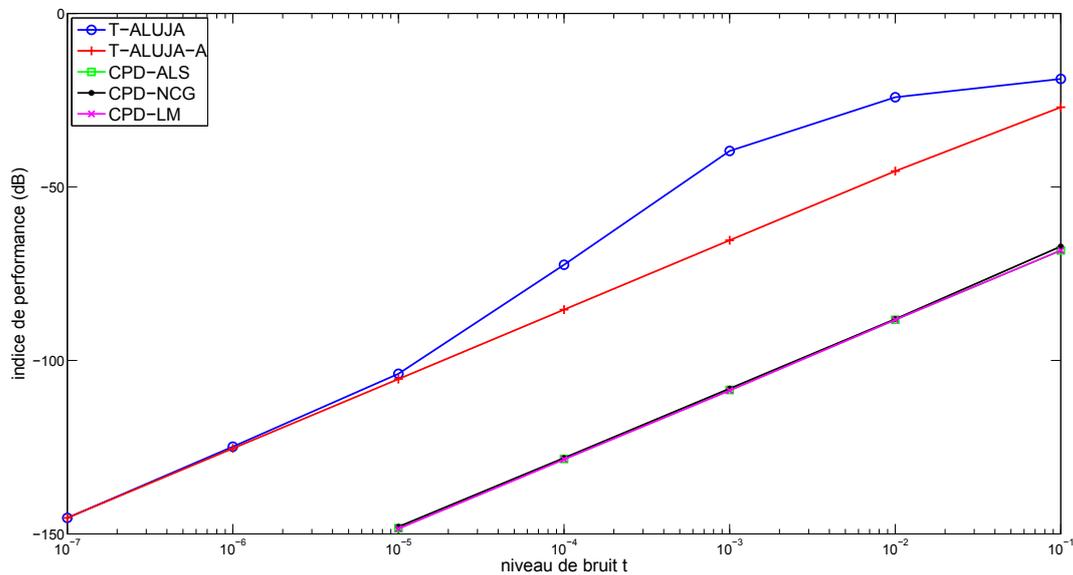


FIGURE 4.10 – Indice de performance médian après 500 itérations en fonction du niveau de bruit t pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ et pour 50 tirages indépendants.

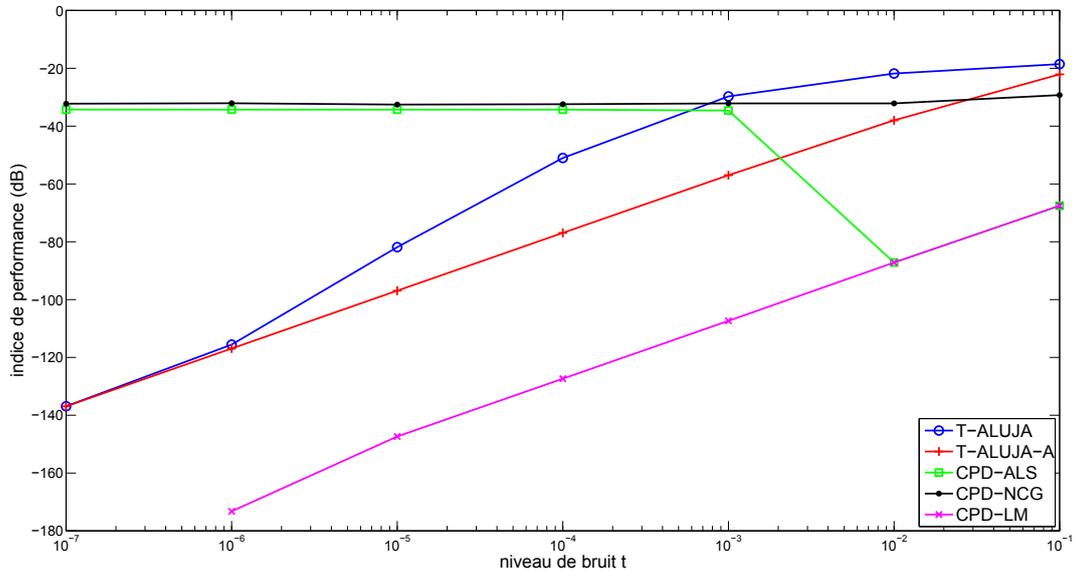


FIGURE 4.11 – Indice de performance moyen après 500 itérations en fonction du niveau de bruit t pour 10 tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ et pour 50 tirages indépendants.

Influence du nombre de tenseurs : Intéressons nous, maintenant, à l’impact du nombre K de tenseurs à conjointement diagonaliser sur les algorithmes considérés. Notons que, pour $K = 1$, on compare T-ALUJA et T-ALUJA-A aux trois mêmes algorithmes de décomposition CP mais, cette fois, adaptés pour un tenseur d’ordre trois. Les figures 4.12 et 4.13 montrent, respectivement, les performances médianes et moyennes des algorithmes en fonction de K après 500 itérations. Ce paramètre a globalement peu d’influence sur les performances médianes des algorithmes. Cependant, l’augmentation du nombre de tenseurs dans l’ensemble considéré améliore légèrement la valeur atteinte par l’indice de performance des différents algorithmes. Pour CPD-NCG, les cas de tenseurs présentant une quatrième dimension $K < 3$ semblent fortement dégrader son pourcentage de convergence. En effet, lorsque $K = 1$, pour plus de 50% des tirages, l’indice de performance de CPD-NCG ne descend pas en dessous de -33 dB. Sur la figure 4.13, on observe que, pour $K \leq 5$, CPD-NCG et CPD-ALS n’offrent pas un bon pourcentage de convergence contrairement à T-ALUJA, T-ALUJA-A et CPD-LM.

Les figures 4.14 et 4.15 mettent en lumière la diagonalisation d’un seul tenseur d’ordre trois ($K = 1$). Dans ce cas précis, la vitesse de convergence de tous les algorithmes est réduite par rapport à $K = 10$ (voir figure 4.6). Ces figures confirment que les pourcentages de convergence moyens de CPD-ALS et CPD-NCG sont mauvais. En effet, ces deux algorithmes ne convergent pas en moins de 500 itérations pour, respectivement, 46% et 66% des tirages.

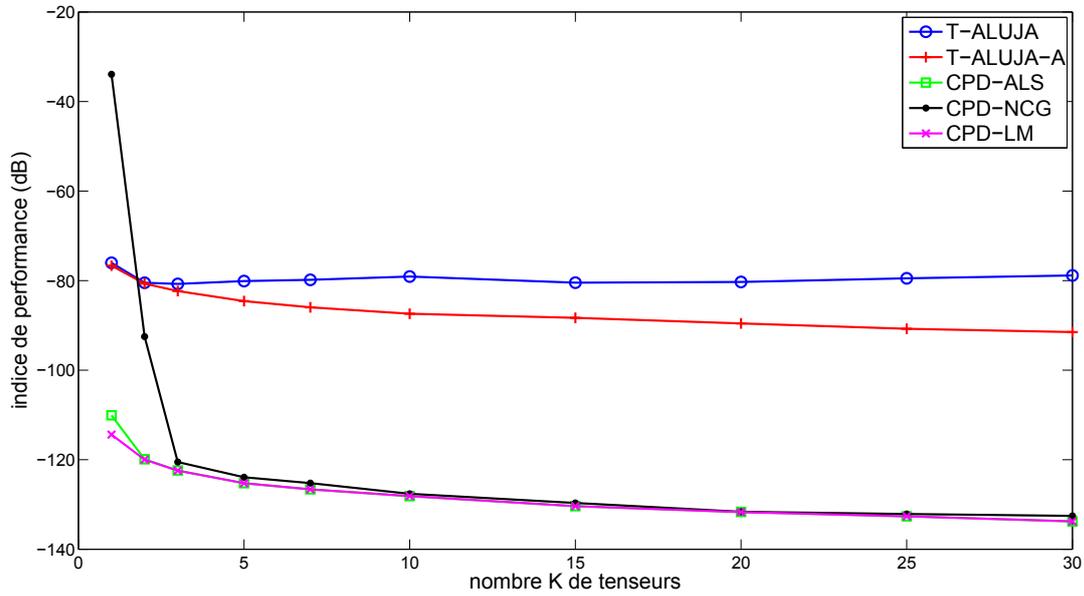


FIGURE 4.12 – Indice de performance médian après 500 itérations en fonction du nombre K de tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.

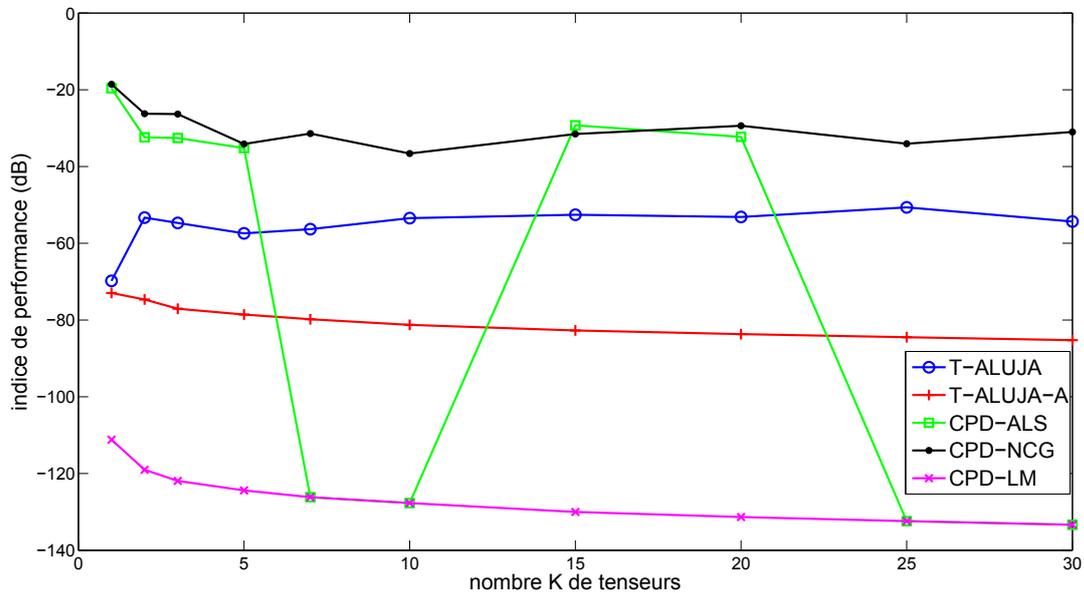


FIGURE 4.13 – Indice de performance moyen après 500 itérations en fonction du nombre K de tenseurs non-symétriques de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.

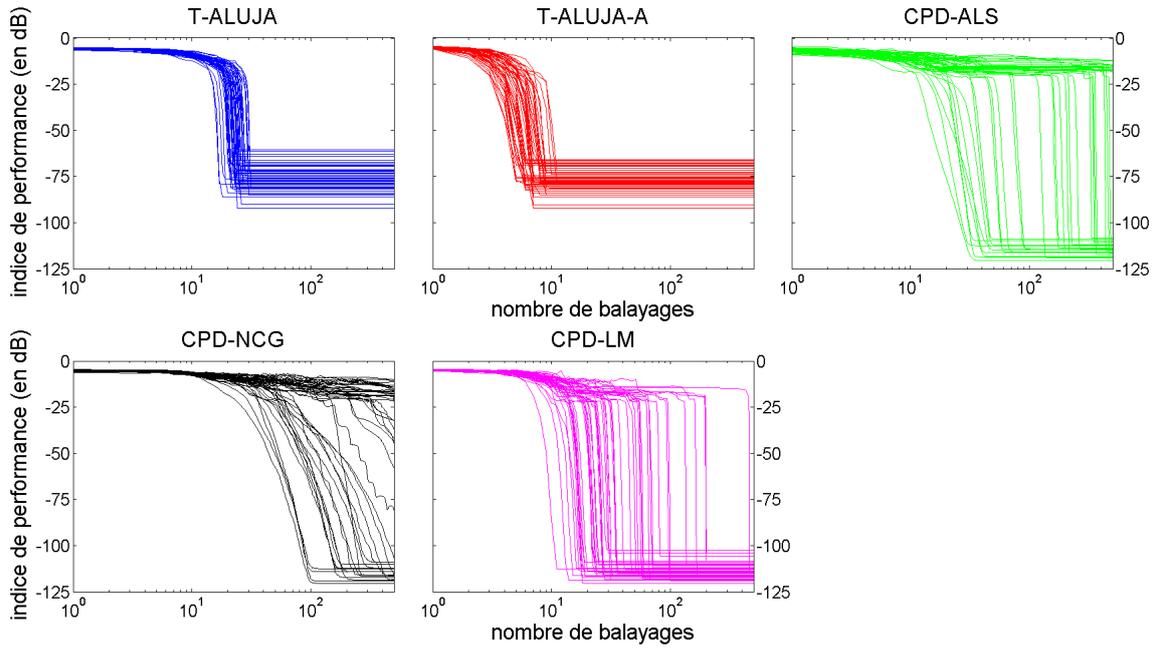


FIGURE 4.14 – Indices de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour 1 tenseur non-symétrique de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$.

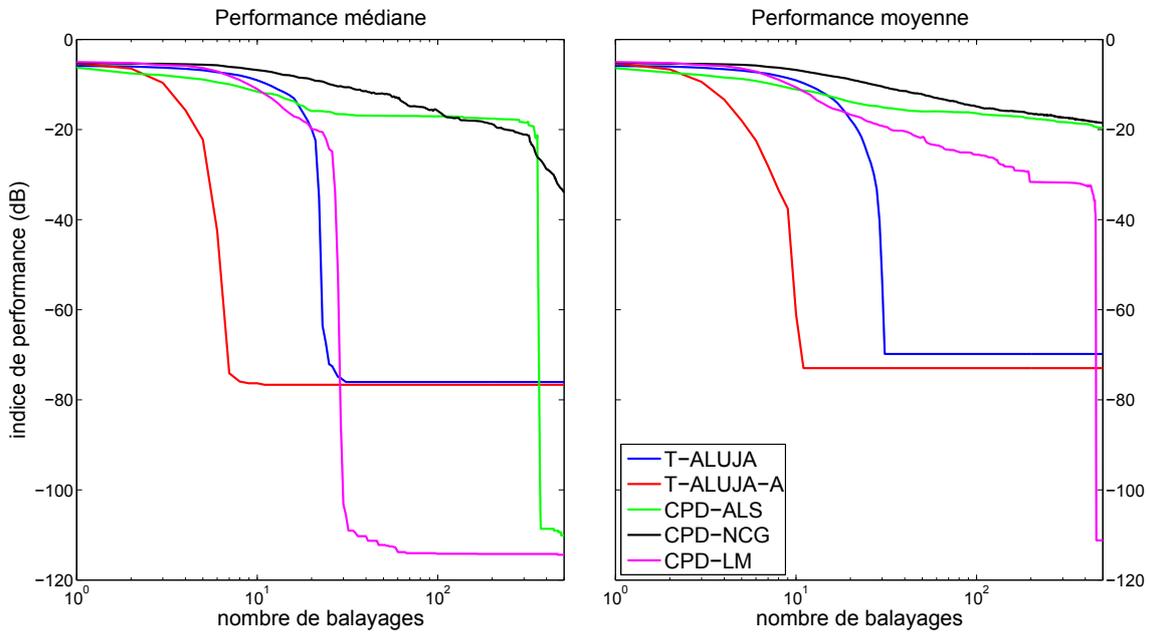


FIGURE 4.15 – Indices de performance médians et moyens en fonction du nombre de balayages pour 1 tenseur non-symétrique de dimensions $12 \times 15 \times 20$ avec $N_s = 10$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.

CPD-LM et les algorithmes proposés convergent, quant à eux, pour 100% des tirages, en moyenne en 461 itérations pour CPD-LM, en 31 itérations pour T-ALUJA et en 11 itérations pour T-ALUJA-A.

Influence de la dimension N_s des tenseurs : Abordons, enfin, l'influence du paramètre N_s sur l'indice de performance médian des algorithmes. Ceci est illustré par la figure 4.15. Elle montre que lorsque N_s augmente, alors le niveau de performance de tous les algorithmes se dégrade. Toutefois, cette remarque s'applique particulièrement à T-ALUJA dont les performances sont fortement impactées par N_s . Le comportement particulier de CPD-NCG révèle un pourcentage de convergence en moins de 500 itérations inférieur à 50% pour $N_s \in \{15, 20, 25\}$.

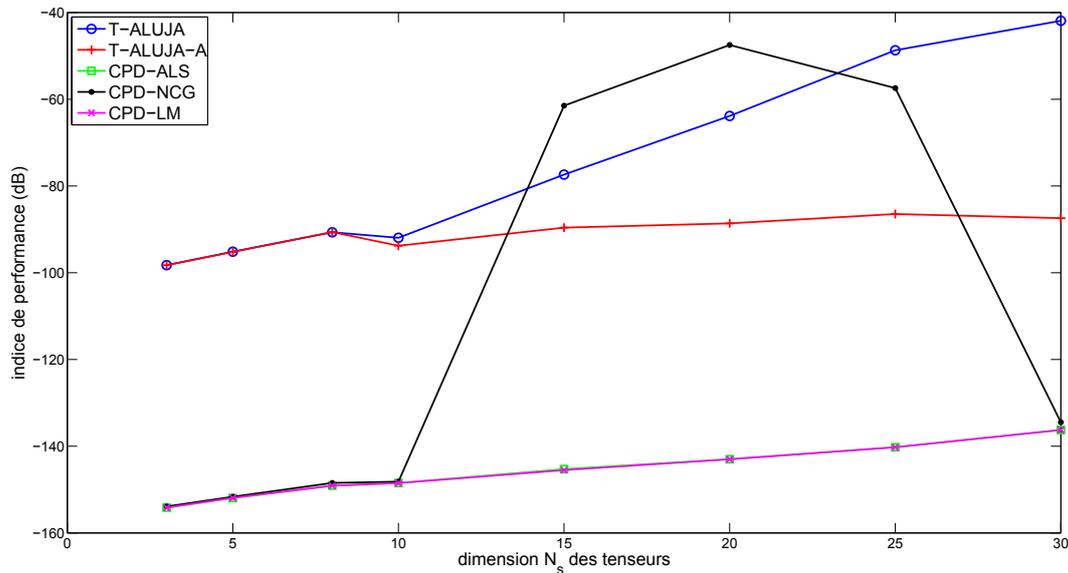


FIGURE 4.16 – Indice de performance médian après 500 itérations en fonction de la dimension réduite N_s des 15 tenseurs $32 \times 35 \times 40$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.

Avec les figures 4.17 et 4.18, on s'intéresse plus particulièrement au cas où $N_s = 30$. Ici, les algorithmes atteignent des niveaux de performance similaires à ceux des figures 4.6 et 4.7 où $N_s = 10$, sauf T-ALUJA, dont les performances sont fortement dégradées. Notons, enfin, que le paramètre N_s a aussi une influence sur la vitesse de convergence de chacun des algorithmes. Lorsque N_s augmente, le nombre d'itérations nécessaires aux algorithmes pour converger augmente aussi. Notons, aussi, que T-ALUJA-A fournit toujours la meilleure vitesse de convergence de ces cinq algorithmes.

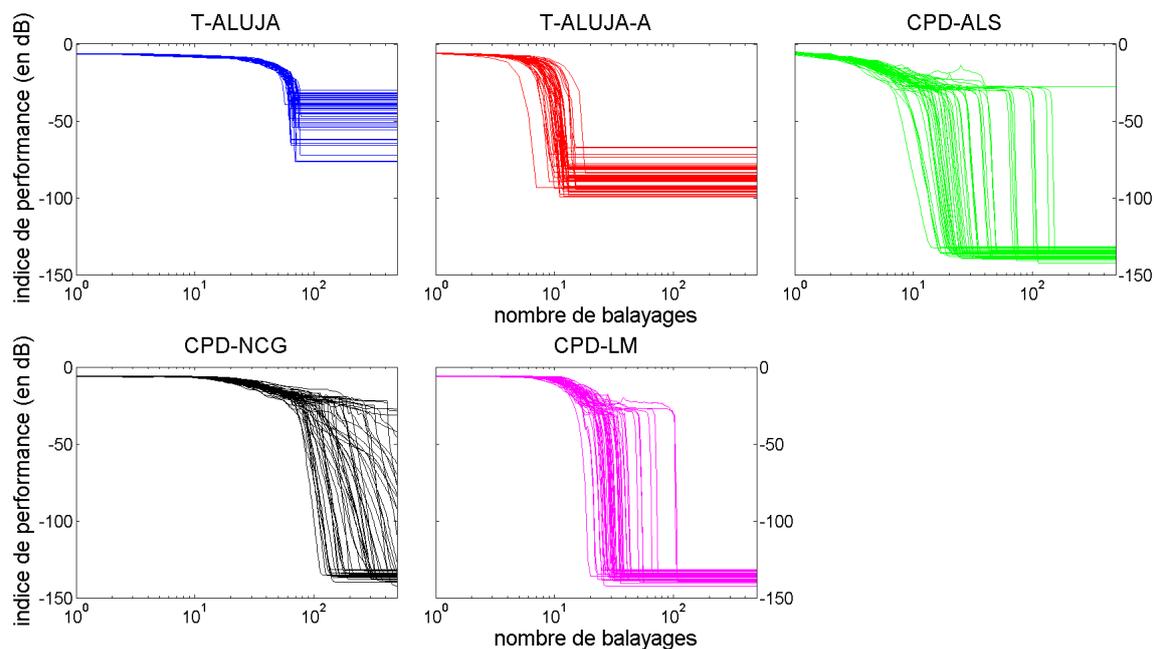


FIGURE 4.17 – Indices de performance de chacun des 50 tirages indépendants en fonction du nombre de balayages pour des ensembles de 10 tenseurs non-symétriques de dimensions $32 \times 35 \times 40$ avec $N_s = 30$ dans un contexte bruité $t = 10^{-4}$.

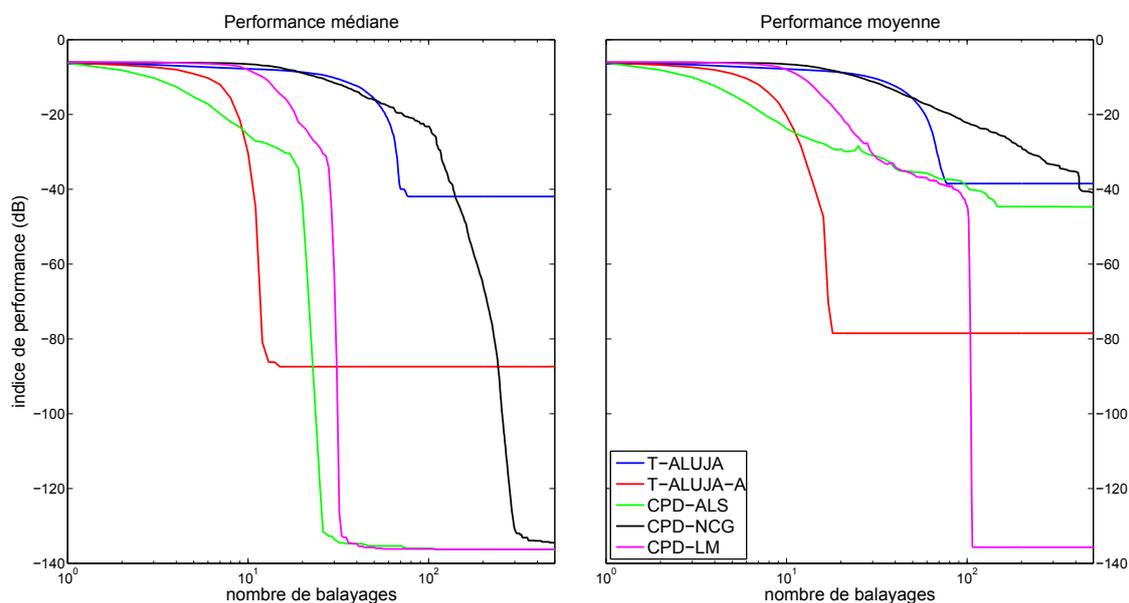


FIGURE 4.18 – Indices de performance médians et moyens en fonction du nombre de balayages pour 10 tenseurs non-symétriques de dimensions $32 \times 35 \times 40$ avec $N_s = 30$ dans un contexte bruité $t = 10^{-4}$ et pour 50 tirages indépendants.

Contexte 1 : 10 tenseurs $12 \times 15 \times 20$ avec $N_s = 10$ et $t = 10^{-4}$ (50 tirages indépendants)					
Algorithmes	T-ALUJA	T-ALUJA-A	CPD-ALS	CPD-NCG	CPD-LM
Tirages convergents en moins de 500 <i>ite</i>	100%	100%	98%	80%	100%
moyenne	24.5	3.96	N/A	N/A	23
écart-type	4.296	0.638	N/A	N/A	7.229
minimum	18	3	10	76	13
maximum	235	5	N/A	N/A	59
médiane	24	4	17	209	22
écart interquartile	6	0	9	203	7
Contexte 2 : 2 tenseurs $12 \times 15 \times 20$ avec $N_s = 10$ et $t = 10^{-4}$ (50 tirages indépendants)					
Algorithmes	T-ALUJA	T-ALUJA-A	CPD-ALS	CPD-NCG	CPD-LM
Tirages convergents en moins de 500 <i>ite</i>	100%	100%	98%	46%	100%
moyenne	23.78	5.28	N/A	N/A	30
écart-type	3.776	0.991	N/A	N/A	14.177
minimum	16	4	16	112	15
maximum	36	8	N/A	N/A	104
médiane	24	5	32	N/A	26.5
écart interquartile	3	1	26	N/A	15
Contexte 3 : 10 tenseurs $32 \times 35 \times 40$ avec $N_s = 30$ et $t = 10^{-4}$ (50 tirages indépendants)					
Algorithmes	T-ALUJA	T-ALUJA-A	CPD-ALS	CPD-NCG	CPD-LM
Tirages convergents en moins de 500 <i>ite</i>	100%	100%	98%	70%	100%
moyenne	73.78	12.1	N/A	N/A	37.04
écart-type	6.231	1.843	N/A	N/A	17.679
minimum	63	7	14	115	19
maximum	93	18	N/A	N/A	107
médiane	72.5	12	26	321	32
écart interquartile	6	2	24	N/A	9

TABLE 4.4 – Quelques statistiques sur le nombre d’itérations nécessaires aux cinq algorithmes pour converger pour différents contextes de simulation (toutes les statistiques sont exprimées en itérations).

4.3.5 Statistiques sur la vitesse de convergence

Le tableau 4.4 permet de quantifier, à l’aide de statistiques (moyenne, écart-type, minimum, maximum, médiane, écart interquartile), la vitesse de convergence de chacun des algorithmes. Ainsi pour chaque algorithme et pour chacun des tirages, on relève l’itération après laquelle l’indice de performance n’évolue plus. Par cela, on entend que la différence en valeur absolue des performances atteintes entre deux itérations consécutives ne dépasse pas un

seuil ϵ . Ici, ce seuil est calibré sur la meilleure performance atteinte, tout algorithme confondu (soit environ $\epsilon = 10^{-14}$). Lorsqu'on n'a pas 100% des tirages qui convergent en moins de 500 itérations pour un algorithme, certaines statistiques ne peuvent pas être renseignées (N/A). Les meilleures statistiques sont notées en gras.

Pour chacun des trois cas d'étude, T-ALUJA-A présente toutes les meilleures statistiques. La comparaison la plus pertinente est à faire avec CPD-LM qui est le seul algorithme de décomposition CP à présenter un pourcentage de convergence égal à 100%. Notons qu'en moyenne, T-ALUJA-A nécessite entre 19 et 25 itérations de moins que CPD-LM pour converger et entre 18 et 21 itérations en valeur médiane. Relevons aussi la très bonne dispersion concernant la vitesse de convergence de T-ALUJA-A avec un écart interquartile de 0 dans le contexte de simulation 1 et de maximum 2 dans le contexte de simulation 3. La diminution du nombre K de tenseurs ou l'augmentation de la dimension N_s des tenseurs de l'ensemble influe beaucoup plus sur la dispersion de CPD-LM. Enfin, relevons que dans les trois contextes étudiés, T-ALUJA-A converge pour l'intégralité des tirages avant que CPD-LM ne converge pour son « meilleur » tirage. En effet, l'itération maximum de convergence de T-ALUJA-A est toujours inférieure à l'itération minimum de convergence de CPD-LM. Ces très bonnes statistiques de T-ALUJA-A sont d'autant plus intéressantes que cet algorithme possède aussi la plus faible complexité (voir tableau 4.2).

Conclusion

Dans ce chapitre, nous avons proposé deux algorithmes de diagonalisation conjointe de tenseurs d'ordre trois. Parmi eux, T-ALUJA-A se compare favorablement aux algorithmes de décomposition CP, en termes de vitesse de convergence, de pourcentage de convergence et de complexité. T-ALUJA-A semble être une bonne alternative pour toute application dont le but est d'obtenir rapidement un résultat, sans avoir nécessairement besoin d'une bonne précision. On peut aussi envisager de combiner T-ALUJA-A avec des algorithmes présentant un meilleur niveau de performance mais nécessitant plus d'itérations pour converger comme CPD-LM. Ainsi, on profiterait de la bonne vitesse de convergence de T-ALUJA-A sur les premières itérations et de la précision de CPD-LM par la suite.

Application à la séparation de sources de télécommunications numériques

Dans ce chapitre, nous allons appliquer les algorithmes développés au cours de cette thèse à la séparation de sources de télécommunications numériques. Dans un premier temps, nous rappelons comment résoudre le problème de séparation de sources à l'aide de la diagonalisation conjointe. Puis nous exposons le cadre des simulations. Celles-ci sont effectuées, tout d'abord, sur des ensembles de matrices ou de tenseurs d'ordre trois symétriques, puis sur des ensembles de matrices hermitiennes ou des ensembles de tenseurs d'ordre trois respectant le modèle INDSCAL_h .

5.1 Séparation de sources

Comme énoncé au paragraphe 2.3, en séparation de sources, on considère le modèle classique d'ICA défini (2.41)

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}(t) = \mathbf{x}(t) + \mathbf{b}(t). \quad (5.1)$$

où les observations $\mathbf{y}(t)$ résultent du mélange linéaire instantané (modélisé par la matrice \mathbf{A}) des signaux sources $\mathbf{s}(t)$, supposés indépendants. Les signaux d'observations sont aussi bruités par un processus blanc gaussien additif $\mathbf{b}(t)$. A partir de ces seules observations, le but de la séparation de sources est d'estimer une matrice de séparation \mathbf{B} qui, appliquée à $\mathbf{y}(t)$, doit permettre de retrouver l'indépendance des sources, perdue lors du mélange. Pour cela, on a vu qu'une des méthodes envisageables est la diagonalisation conjointe. A l'aide d'opérateurs statistiques, on est en mesure de construire, à partir des observations, un ensemble de tenseurs cibles d'ordre supérieur ou égal à deux. La diagonalisation conjointe d'un tel ensemble permet l'estimation d'une matrice \mathbf{B} .

Ici, pour construire les tenseurs cibles, on propose d'utiliser les cumulants d'ordre quatre définis en (2.44) et (2.45). Cet outil statistique présente plusieurs avantages. Tout d'abord, comme les signaux sources et le bruit blanc gaussien sont statistiquement indépendants, le cumulants d'ordre quatre appliqué aux signaux d'observations (5.1) peut s'écrire de la manière suivante :

$$\begin{aligned} C_{\mathbf{y},1}^{a,b,c,d} &= \text{Cum}(y_a, y_b, y_c, y_d) \\ &= \text{Cum}(x_a, x_b, x_c, x_d) + \text{Cum}(b_a, b_b, b_c, b_d) \\ &= C_{\mathbf{x},1}^{a,b,c,d} + C_{\mathbf{b},1}^{a,b,c,d}. \end{aligned} \quad (5.2)$$

De plus, comme dans le cadre des télécommunications numériques, le bruit de canal est modélisé classiquement par un processus blanc gaussien, alors son cumulants est nul. Finalement, on obtient

$$C_{\mathbf{y},1}^{a,b,c,d} = C_{\mathbf{x},1}^{a,b,c,d}. \quad (5.3)$$

5.2 Cadre de simulations

Nous considérons une antenne de N_o capteurs recevant N_s sources avec $N_s \leq N_o$. La matrice de mélange de dimensions $N_o \times N_s$ est tirée selon une loi uniforme sur l'intervalle $[-1, 1]$ pour ses parties réelle et imaginaire, puis ses colonnes sont normées. Chacun des signaux numériques sources ou bruit est constitué de N_e échantillons. La matrice de bruit de dimensions $N_o \times N_e$ suit une loi normale centrée de variance σ_b^2 .

Concernant les sources, nous considérons les deux modulations suivantes :

- QPSK (pour *Quaternary Phase-Shift Keying*) qui module un signal selon les quatre états $\left\{ e^{-i\frac{3\pi}{4}}, e^{-i\frac{\pi}{4}}, e^{i\frac{\pi}{4}}, e^{i\frac{3\pi}{4}} \right\}$.
- 4-ASK (pour *4-Amplitude-Shift Keying*) qui module un signal selon les quatre états $\left\{ -\frac{3}{\sqrt{5}}, -\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}, \frac{3}{\sqrt{5}} \right\}$.

Les sources de N_e échantillons seront tirées uniformément sur l'ensemble des quatre états pris par la modulation choisie.

Comme la modulation QPSK fournit des signaux circulaires à l'ordre deux (2.47), alors le cumulants $C_{\mathbf{y},2}^{a,b,c,d}$ (2.45) est nul. Nous ne pouvons donc pas utiliser ce cumulants d'ordre quatre. Comme les signaux QPSK sont aussi non-circulaires à l'ordre quatre, nous avons opté pour $C_{\mathbf{y},1}^{a,b,c,d}$ (2.44) qui est associé à la diagonalisation conjointe de matrices symétriques et à celle de tenseurs d'ordre trois symétriques. La modulation 4-ASK donne des signaux non-circulaires permettant l'utilisation de n'importe quel cumulants d'ordre quatre. Toutefois, afin d'étudier la diagonalisation conjointe de matrices hermitiennes et de tenseurs d'ordre trois

INDSCAL_h, nous avons choisi d'utiliser le cumulante $C_{\mathbf{y},2}^{a,b,c,d}$ (2.45).

Pour les différentes simulations, nous introduisons aussi le rapport signal sur bruit, qui est défini comme le rapport entre la puissance du signal σ_s^2 et celle du bruit σ_b^2 et qui s'écrit

$$\text{RSB}_{\text{dB}} = 10 \log_{10} \left(\frac{\sigma_s^2}{\sigma_b^2} \right). \quad (5.4)$$

Dans les deux cadres de simulation, nous proposons de valider nos algorithmes dans un cas pratique et de les confronter aux mêmes algorithmes de la littérature qu'aux paragraphes 3.4 et 4.3. Seul l'algorithme CPD-NCG n'a pas été retenu pour ces simulations en raison de sa très faible vitesse de convergence. Pour les algorithmes de décomposition canonique polyadique, le tenseur d'ordre quatre résultant du calcul de cumulante sera directement utilisé. Pour les algorithmes de diagonalisation conjointe de tenseurs d'ordre trois, le cumulante d'ordre quatre sera rangé comme (2.59) dans le cas symétrique et comme (2.60) dans le cas INDSCAL_h. Enfin, pour les algorithmes de diagonalisation conjointe de matrices, les statistiques d'ordre supérieur seront rangées soit comme (2.48), soit comme (2.49) respectivement pour les cas des matrices symétriques et hermitiennes. Si N_o est supérieur à N_s , comme les algorithmes proposés nécessitent une étape de réduction de dimensions, alors, par soucis d'équité, nous effectuerons ce prétraitement pour tous les algorithmes. Finalement, les algorithmes de CPD décomposeront un tenseur de dimensions $N_s \times N_s \times N_s \times N_s$, tandis que les algorithmes de diagonalisation conjointe diagonaliseront soit un ensemble de N_s tenseurs de dimension $N_s \times N_s \times N_s$, soit un ensemble de N_s^2 matrices de dimensions $N_s \times N_s$.

Toutes les simulations sont effectuées pour 1000 balayages (ou itérations) de chacun des algorithmes et elles sont réalisées pour 50 tirages indépendants. Le niveau de performance est mesuré à l'aide de l'indice (3.116). Le comportement des algorithmes est étudié en fonction des différents paramètres du problème, à savoir, le nombre rapport signal sur bruit, le nombre de sources et le nombre d'échantillons considérés. On s'intéresse, en particulier, à trois types de performances pour chacun des algorithmes.

- L'indice de performance médian (3.116) après 1000 itérations : celui-ci est considéré comme bon s'il est inférieur à -30dB , moyen s'il est compris entre -30dB et -20dB et mauvais quand il est supérieur à -20dB .
- Le nombre d'itérations nécessaires à un algorithme pour converger (dans le sens où son indice de performance n'évolue plus) que l'on nomme, ici, vitesse de convergence. Elle sera considérée comme bonne si elle est inférieure à 30 itérations, moyenne si elle est comprise entre 30 et 100 itérations et mauvaise lorsqu'elle prend une valeur supérieure à 100 itérations.
- Le pourcentage de réalisations pour lequel un algorithme converge en moins de 1000

itérations que l'on nomme simplement pourcentage de convergence. Il est considéré comme bon à 100%, moyen entre 90% et 100% et mauvais lorsqu'il est inférieur à 90%.

Enfin, on observera un des signaux sources obtenus, après diagonalisation de l'ensemble cible, par trois des algorithmes étudiés.

Le tableau 5.1 résume, pour chacune des simulations et pour les deux modulations étudiées, le numéro de la figure associée, les valeurs utilisées pour les différents paramètres et le type de courbe tracée.

Modulation QPSK et cas symétrique					
Figure	N_o	N_s	N_e	RSB _{dB}	affichage
5.1	7	5	2^{12}	20	toutes réalisations
5.2	7	5	2^{12}	20	méd. et moy.
5.3	7	5	2^{12}	$\{-10, \dots, 50\}$	niv. perf. méd.
5.4	7	5	2^{12}	$\{-10, \dots, 50\}$	vit. conv. méd.
5.5	10	3	2^{12}	$\{20, 30, 40\}$	source
5.6	18	$\{2, \dots, 15\}$	2^{12}	20	niv. perf. méd.
5.7	18	$\{2, \dots, 15\}$	2^{12}	20	vit. conv. méd.
5.8	10	$\{2, 3, 10\}$	2^{12}	30	source
5.9	6	5	$\{2^8, \dots, 2^{18}\}$	20	niv. perf. méd.
5.10	6	5	$\{2^8, \dots, 2^{18}\}$	20	vit. conv. méd.
5.11	10	3	$\{2^8, 2^{12}, 2^{14}\}$	30	source
Modulation 4-ASK et cas hermitien/INDSCAL _h					
Figure	N_o	N_s	N_e	RSB _{dB}	affichage
5.12	8	7	2^{14}	30	toutes réalisations
5.13	8	7	2^{14}	30	méd. et moy.
5.14	8	7	2^{14}	$\{-10, \dots, 50\}$	niv. perf. méd.
5.15	8	7	2^{14}	$\{-10, \dots, 50\}$	vit. conv. méd.
5.16	5	3	2^{12}	$\{20, 30, 40\}$	source
5.17	15	$\{2, \dots, 15\}$	2^{14}	20	niv. perf. méd.
5.18	15	$\{2, \dots, 15\}$	2^{14}	20	vit. conv. méd.
5.19	5	$\{2, 3, 5\}$	2^{12}	30	source
5.20	6	5	$\{2^6, \dots, 2^{16}\}$	20	niv. perf. méd.
5.21	6	5	$\{2^6, \dots, 2^{16}\}$	20	vit. conv. méd.
5.22	5	3	$\{2^8, 2^{12}, 2^{14}\}$	30	source

TABLE 5.1 – Numéro de figure, paramètres utilisés et courbes affichées pour l'ensemble des simulations liées à l'application à la séparation de sources de signaux de télécommunications numériques.

5.3 Modulation QPSK et cas symétrique

Pour cette première modulation, on compare les algorithmes proposés dans le cadre de la diagonalisation conjointe de matrices (G-CLUJA, A-CLUJA, C-CLUJA et ALUJA) et de tenseurs d'ordre trois (T-ALUJA et T-ALUJA-A) à CS-LUJ1D (présenté au paragraphe 3.2.1.1 comme généralisation au cas complexe symétrique de LUJ1D [1]) et aux algorithmes de décomposition canonique polyadique CPD-ALS [105] et CPD-LM [106].

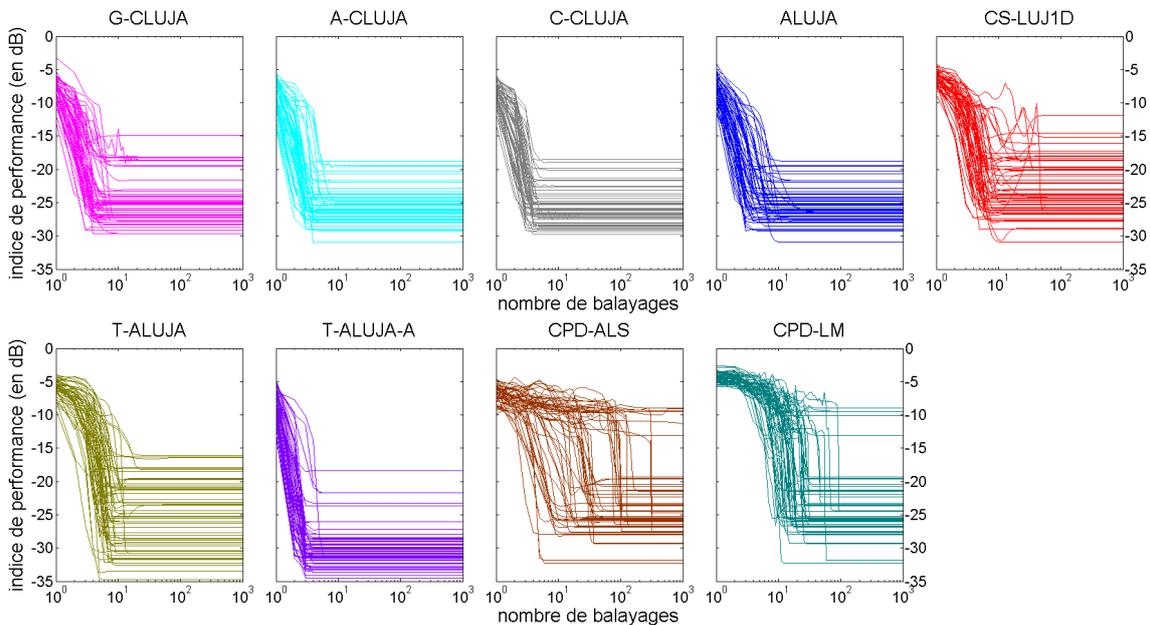


FIGURE 5.1 – Indice de performance de chacun des algorithmes pour toutes les réalisations en fonction du nombre de balayages. Contexte : 7 signaux d'observations de 4096 échantillons, constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK et avec un RSB_{dB} de 20dB.

Pour commencer, la figure 5.1 montre l'évolution de l'indice de performance au cours des balayages pour l'intégralité des réalisations de chacun des algorithmes et la figure 5.2 donne leurs performances médianes et moyennes. L'algorithme T-ALUJA-A donne les meilleurs résultats tant en termes de vitesse de convergence qu'en niveau de performance atteint. Les algorithmes C-CLUJA et A-CLUJA affichent la même vitesse de convergence que T-ALUJA-A, mais avec des performances environ 4dB plus élevées. ALUJA présente un indice de performance moyen similaire aux deux algorithmes précédents mais il nécessite, cependant, 10 itérations de plus pour converger. Notons que les algorithmes CPD possèdent les moins bonnes vitesses de convergence qui, de plus, varient beaucoup d'une réalisation à l'autre. CPD-ALS est le seul algorithme qui ne présente pas un pourcentage de convergence en moins de 1000

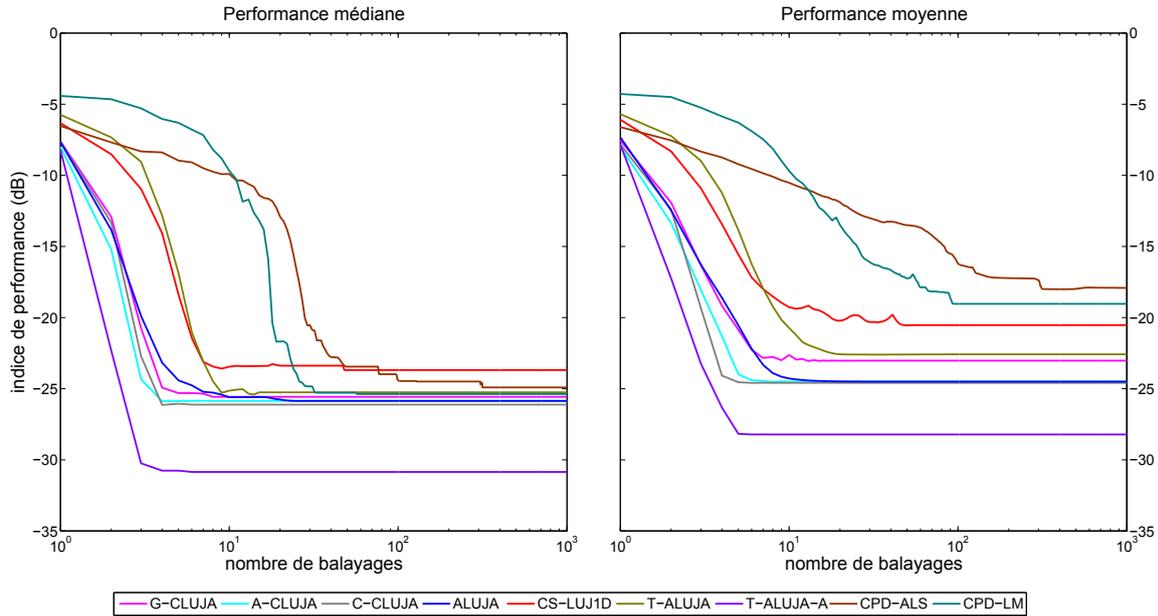


FIGURE 5.2 – Indices de performance médians et moyens des algorithmes (50 réalisations) en fonction du nombre de balayages pour 7 signaux d’observations de 4096 échantillons, constitués d’un mélange linéaire de 5 sources respectant une modulation QPSK et avec un RSB_{dB} de 20dB.

itérations de 100%. Nous allons, maintenant, étudier l’influence respective des paramètres RSB_{dB} , N_s et N_e sur ces algorithmes.

5.3.1 Influence du rapport signal sur bruit

Tout d’abord, regardons l’impact du rapport signal sur bruit. La figure 5.3 expose l’indice de performance médian obtenu par les différents algorithmes en fonction du RSB_{dB} . Pour un RSB_{dB} inférieur ou égal à 0dB, les résultats obtenus sont mauvais. En effet tous les algorithmes affichent des performances supérieures à $-10dB$, ce qui ne permet pas la séparation des sources. Pour $RSB_{dB} = 10dB$, CS-LUJ1D, T-ALUJA et G-CLUJA présentent encore un indice de performance supérieur à $-20dB$. Pour ce même cas, T-ALUJA-A, CPD-ALS et CPD-LM atteignent un niveau de performance similaire à environ $-22dB$. Finalement, lorsque le rapport signal sur bruit augmente encore ($RSB_{dB} > 10dB$), l’indice de performance de T-ALUJA-A devient meilleur que ceux de tous les autres algorithmes, convergeant notamment 4dB plus bas que celui de T-ALUJA.

Le tableau 5.2 donne une idée de la robustesse des algorithmes vis-à-vis du bruit. En effet, on retrouve dans ce tableau, le pourcentage de convergence en moins de 1000 balayages des algorithmes sur les 50 réalisations. On ne répertorie que les algorithmes ne présentant

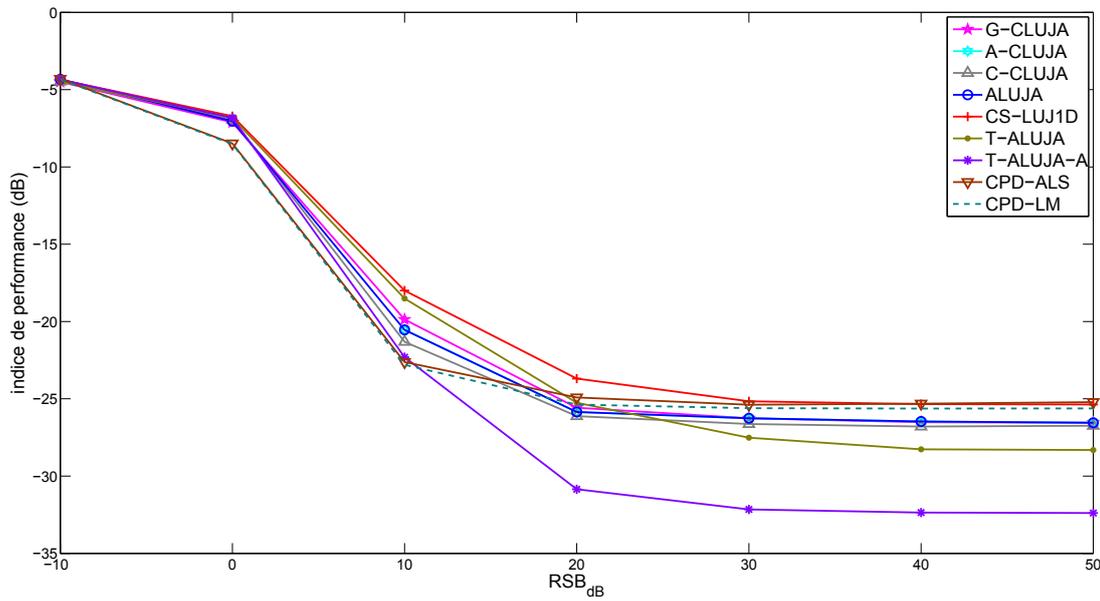


FIGURE 5.3 – Indice de performance médian (50 réalisations) après 1000 balayages en fonction du RSB_{dB} pour 7 signaux d’observations de 4096 échantillons, constitués d’un mélange linéaire de 5 sources respectant une modulation QPSK.

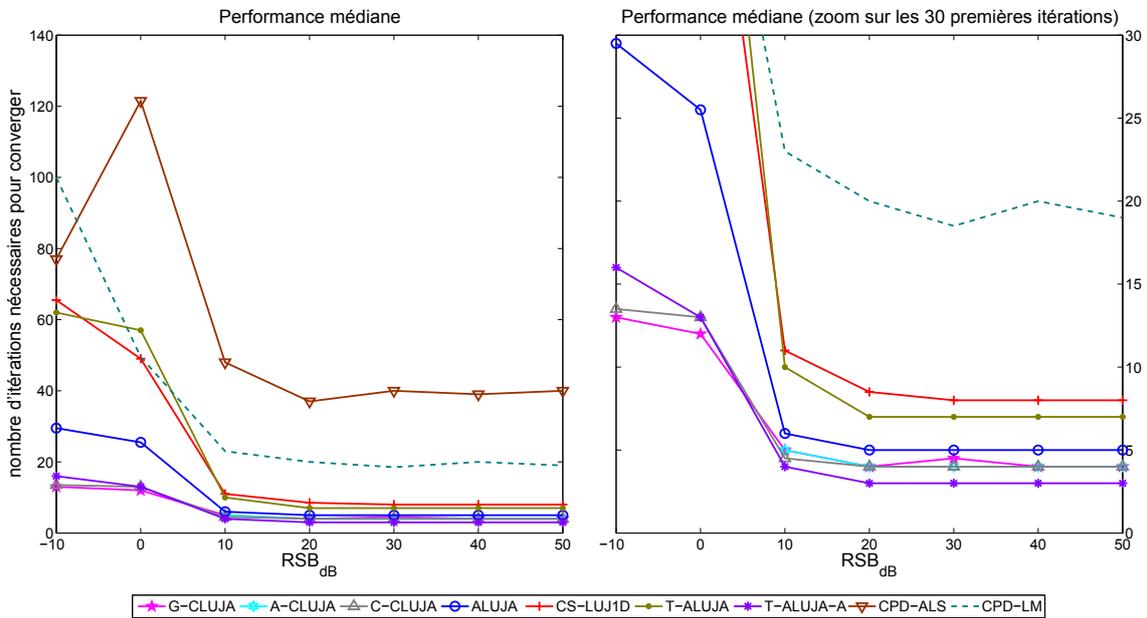


FIGURE 5.4 – Vitesse de convergence médiane (50 réalisations) en fonction du RSB_{dB} pour 7 signaux d’observations de 4096 échantillons, constitués d’un mélange linéaire de 5 sources respectant une modulation QPSK.

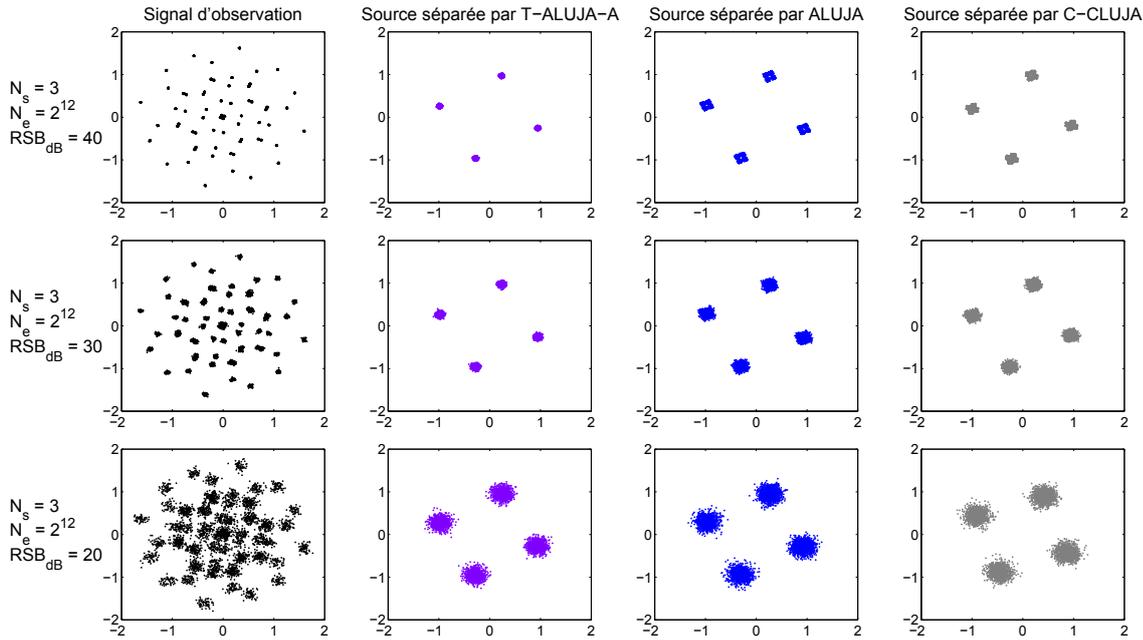


FIGURE 5.5 – Exemple explicite de séparation de sources en fonction du RSB_{dB} pour 10 signaux d’observations de 4096 échantillons, constitués d’un mélange linéaire de 3 sources respectant une modulation QPSK. De gauche à droite : un des signaux d’observations puis une des sources séparée respectivement par T-ALUJA-A, ALUJA et C-CLUJA

RSB_{dB}	-10dB	0dB	10dB	20dB	30dB	40dB	50dB
A-CLUJA	24%	52%	100%	100%	100%	100%	100%
CPD-ALS	98%	100%	100%	100%	100%	100%	98%

TABLE 5.2 – Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du RSB_{dB} pour 7 signaux d’observations de 4096 échantillons, constitués d’un mélange linéaire de 5 sources respectant une modulation QPSK (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les RSB_{dB}).

pas 100% de convergence. Ainsi, en présence d’un très fort bruit ($RSB_{dB} < 10dB$), A-CLUJA diverge, tandis que les autres algorithmes convergent vers une solution. Notons que CPD-ALS ne converge pas en moins de 1000 itérations pour un tirage lorsque $RSB_{dB} = -10dB$ mais aussi, de façon plus surprenante, pour un tirage lorsque $RSB_{dB} = 50dB$.

La figure 5.4 concerne l’évolution de la vitesse de convergence en fonction du rapport signal sur bruit. Les algorithmes CPD affichent, globalement, les plus mauvaises vitesses de convergence. Pour tout rapport signal sur bruit supérieur ou égal à 10dB, T-ALUJA-A est l’algorithme nécessitant le moins d’itérations pour converger. En effet, lorsque $RSB_{dB} \geq 20dB$, T-ALUJA-A converge en seulement trois itérations, contre environ 4 itérations pour les trois algorithmes CLUJA et 5 itérations pour ALUJA.

La figure 5.5, donne un exemple de séparation de sources effectuée par T-ALUJA-A, ALUJA et C-CLUJA pour trois RSB_{dB} différents. Idéalement, un signal QPSK est représenté par une constellation de quatre états. Chacun de ces états est précisément localisé en un point dans le plan complexe. Un signal QPSK bruité verra ses points dispersés autour de chacun des quatre états de la constellation associée à cette modulation. Ainsi, sur la figure 5.5, pour tout RSB_{dB} , chacun des trois algorithmes fournit bien une source constituée d'une constellation de quatre états distincts représentant un signal QPSK. On constate que plus le bruit augmente, plus les ensembles de points correspondant aux quatre états de la constellation QPSK sont dispersés. Notons, enfin, que pour chaque RSB_{dB} , les points constituant les constellations sont moins dispersées avec T-ALUJA-A qu'avec les deux autres algorithmes.

5.3.2 Influence du nombre de sources

Intéressons nous, maintenant, à l'influence du nombre de sources sur les performances des algorithmes. Les figures 5.6 et 5.7 illustrent, respectivement, l'évolution de leurs indices de performance médian et celle de leurs vitesses de convergence en fonction du nombre de signaux sources. En présence de moins de cinq sources, les algorithmes CPD atteignent les meilleurs niveaux de performance et CPD-ALS présente une bonne vitesse de convergence. Lorsque N_s est supérieur à 5, T-ALUJA-A obtient les meilleurs résultats, tant en termes de niveau de performance atteint qu'en termes de vitesse de convergence. A plus de dix sources, les performances des algorithmes CPD se dégradent fortement et A-CLUJA diverge pour plus de 50% des réalisations (voir tableau 5.3). On note aussi que l'augmentation du nombre de sources impacte très fortement la vitesse de convergence de CPD-ALS avec pas moins de 970 itérations nécessaires pour converger, celle de CPD-LM (285 itérations) et dans une moindre mesure celles de CS-LUJ1D (une centaine d'itérations) et de T-ALUJA (environ 90 itérations). Enfin, T-ALUJA-A est l'algorithme le moins influencé par le paramètre N_s en termes de vitesse de convergence. Lorsque l'on considère 15 sources, T-ALUJA-A converge en seulement 7 itérations, soit respectivement 8, 9 et 14 itérations de moins que C-CLUJA, G-CLUJA et ALUJA.

La figure 5.8 confirme que plus le nombre de sources augmente, plus les niveaux de performance des algorithmes se dégradent. Ceci se traduit par des signaux QPSK présentant des constellations dont les points associés à chacun des quatre état sont de plus en plus dispersés. On note, aussi, que pour chaque N_s , les constellations obtenues avec T-ALUJA-A présentent le moins de dispersion, ce qui confirme encore l'étude précédente.

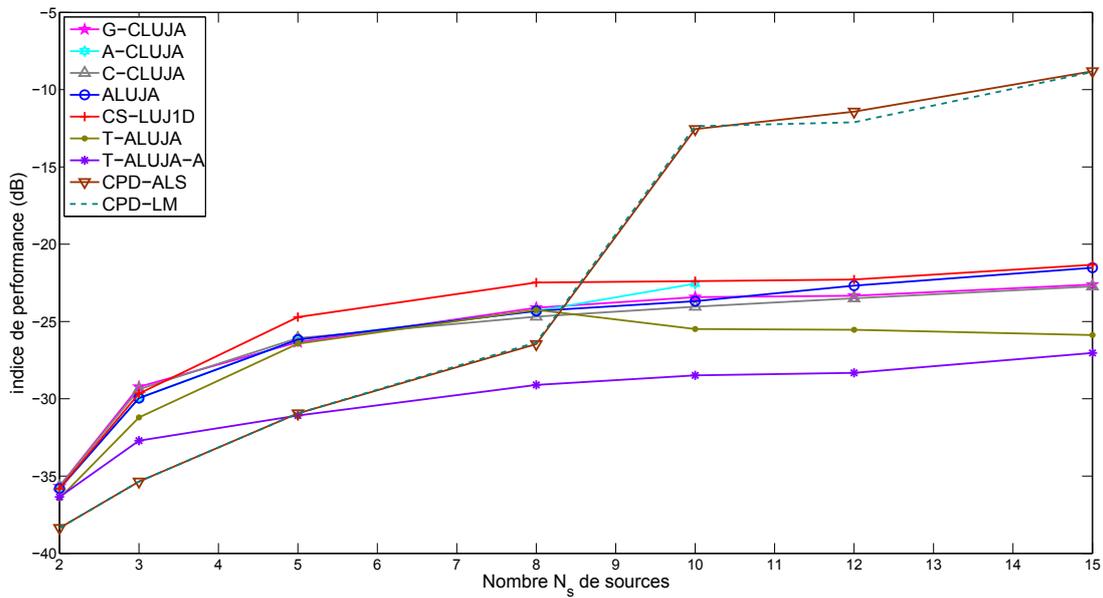


FIGURE 5.6 – Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_s de sources de modulation QPSK. Contexte : 7 signaux d’observations de 4096 échantillons avec un RSB_{dB} de 20dB.

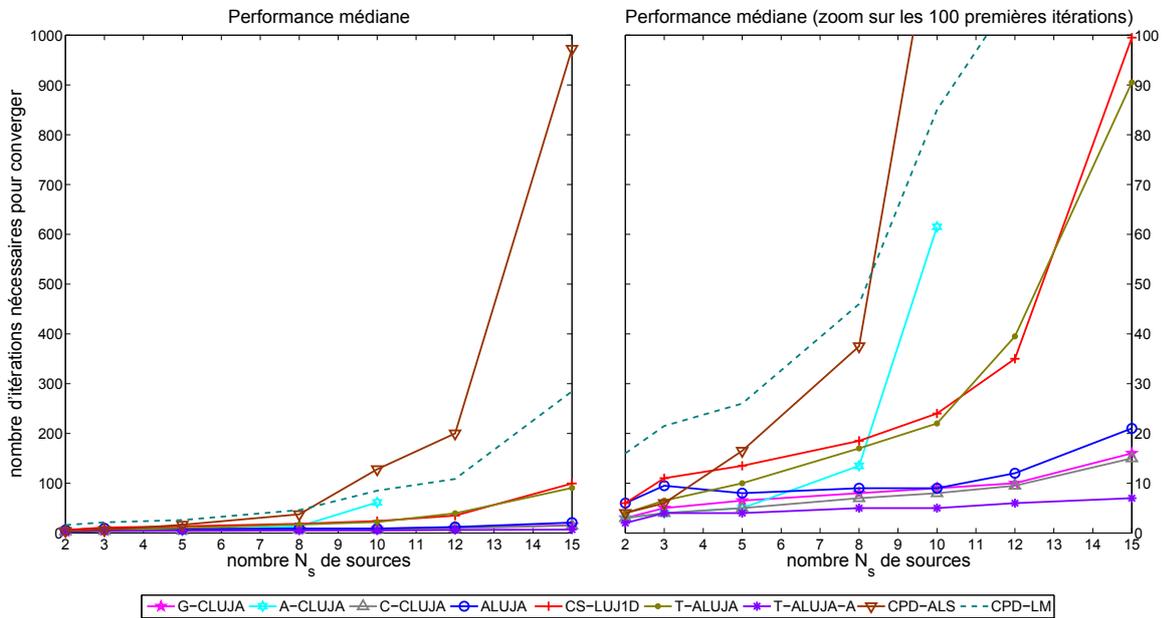


FIGURE 5.7 – Vitesse de convergence médian (50 réalisations) en fonction du nombre N_s de sources de modulation QPSK pour 18 signaux d’observations de 4096 échantillons avec un RSB_{dB} de 20dB.

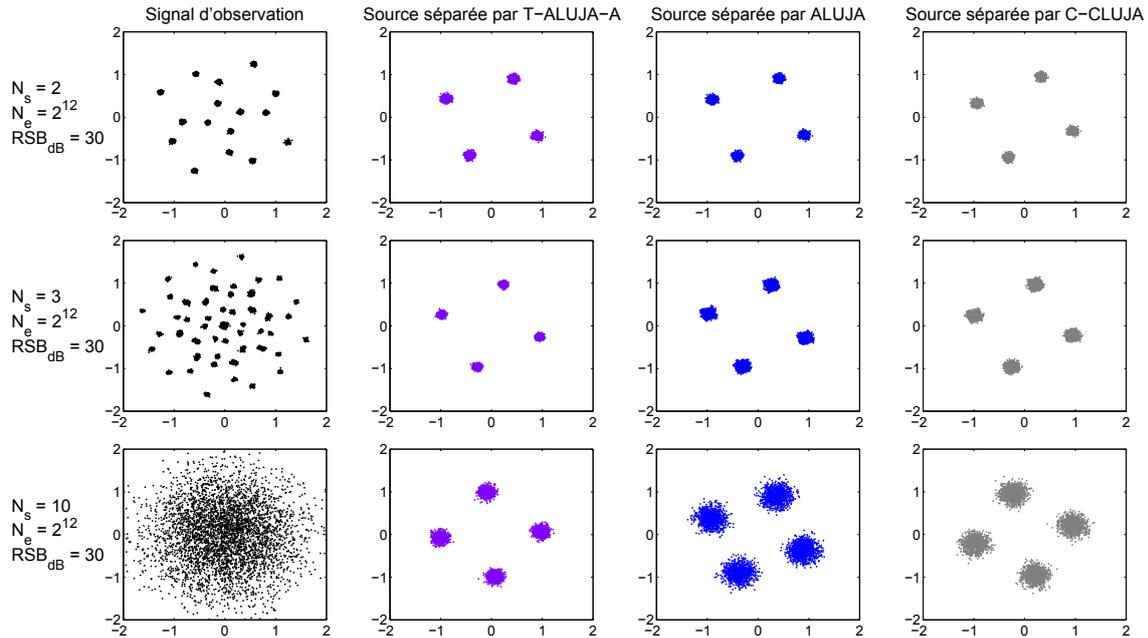


FIGURE 5.8 – Exemple explicite de séparation de sources en fonction du nombre N_s de sources respectant une modulation QPSK pour 10 signaux d’observations de 4096 échantillons avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d’observations puis une des sources séparée respectivement par T-ALUJA-A, ALUJA et C-CLUJA

N_s	2	3	5	8	10	12	15
A-CLUJA	24%	52%	100%	94%	76%	4%	8%
CPD-ALS	100%	100%	100%	98%	98%	96%	74%
CPD-LM	100%	100%	100%	100%	100%	100%	94%

TABLE 5.3 – Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du nombre N_s de sources respectant une modulation QPSK. Contexte : 18 signaux d’observations de 4096 échantillons avec un RSB_{dB} de 20dB (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les N_s).

5.3.3 Influence du nombre d’échantillons considérés

Etudions, enfin, l’influence du nombre d’échantillons utilisés pour le calcul des cumulants cibles d’ordre quatre (2.44). La figure 5.9 montre que ce paramètre a un impact similaire sur tous les algorithmes. Plus le nombre d’échantillons considérés est important, meilleur deviennent les indices de performance des différents algorithmes. Ainsi, les niveaux de performance de tous les algorithmes sont mauvais lorsque $N_e = 256$ et bons lorsque $N_e \geq 16384$. On relève que quel que soit le nombre d’échantillons considérés, l’indice de performance de T-ALUJA-A est toujours le meilleur et converge entre 3dB et 5dB plus bas que T-ALUJA.

Encore une fois, sur les neuf algorithmes considérés, sept convergent pour 100% des

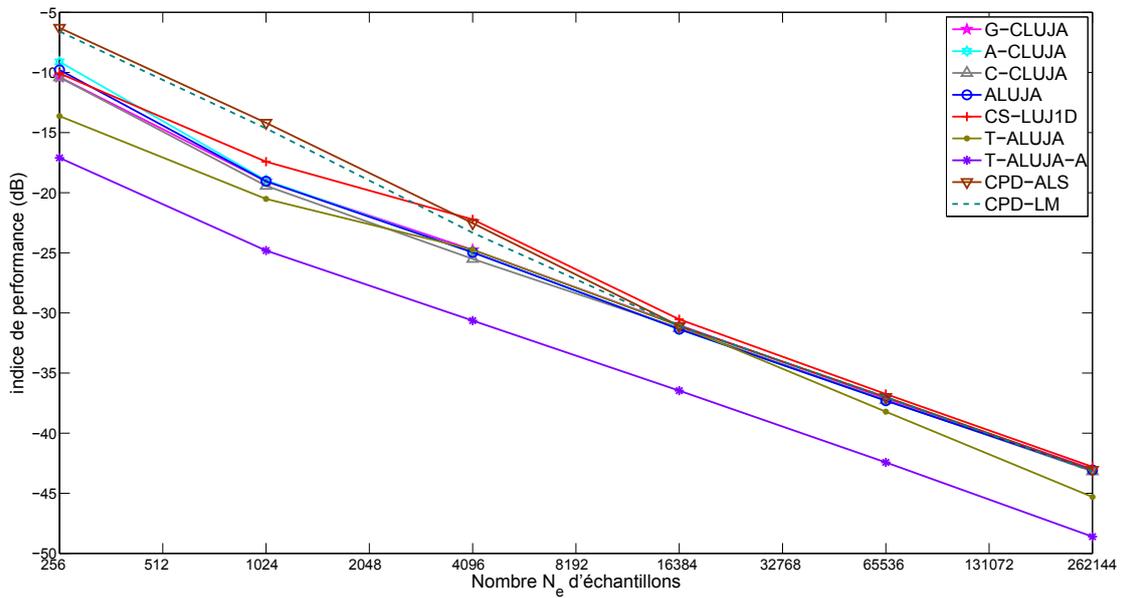


FIGURE 5.9 – Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK avec un RSB_{dB} de 20dB.

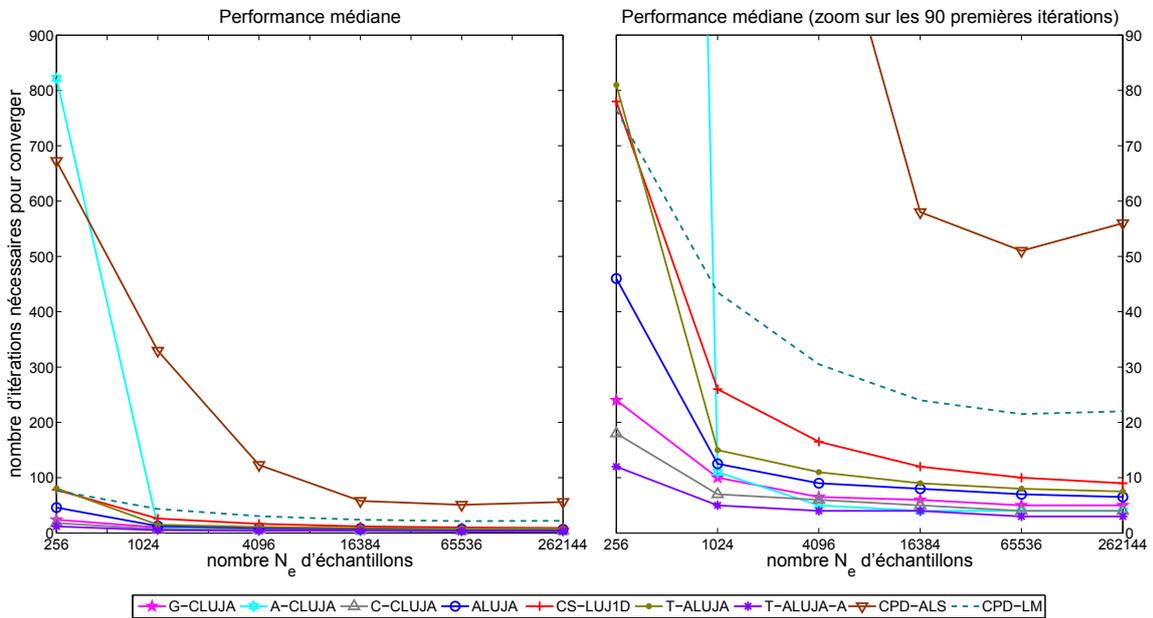


FIGURE 5.10 – Vitesse de convergence médiane (50 réalisations) en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK avec un RSB_{dB} de 20dB.

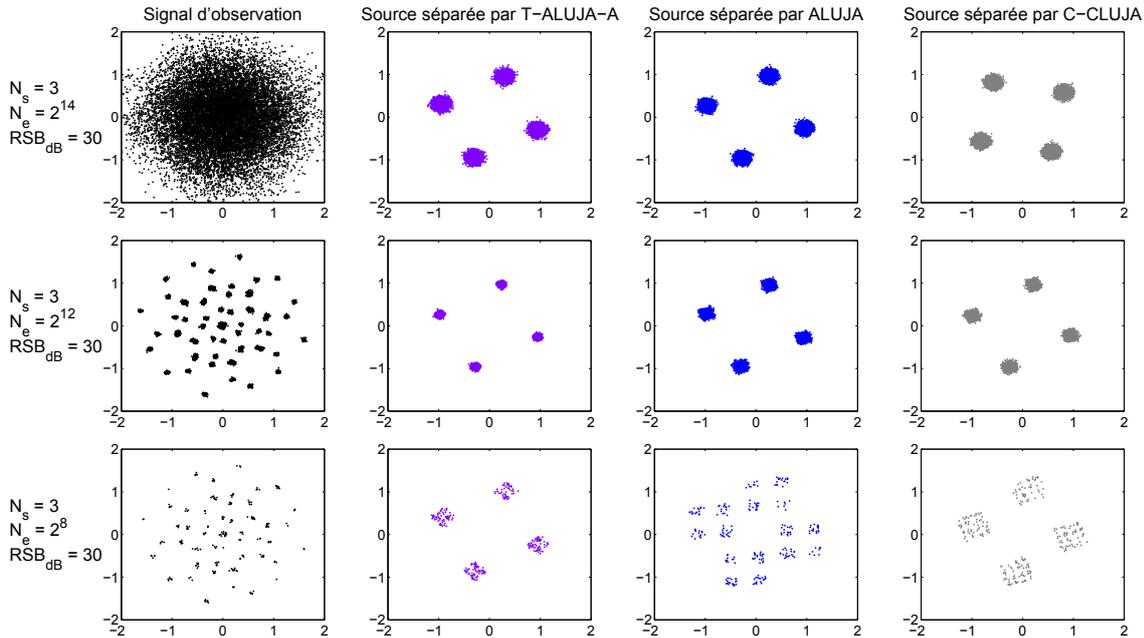


FIGURE 5.11 – Exemple explicite de séparation de sources en fonction du nombre N_e d'échantillons pour 10 signaux d'observations constitués d'un mélange linéaire de 3 sources respectant une modulation QPSK avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, ALUJA et C-CLUJA

N_e	256	1024	4096	16384	65536	262144
A-CLUJA	68%	92%	100%	100%	100%	100%
CPD-ALS	82%	90%	94%	98%	96%	96%

TABLE 5.4 – Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation QPSK avec un RSB_{dB} de 20dB (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les N_e).

réalisations quel que soit le nombre d'échantillons pris en compte. Comme on peut le voir dans le tableau 5.4, CPD-ALS ne présente pas un bon pourcentage de convergence pour tout N_e et A-CLUJA ne parvient plus à converger en moins de 1000 itérations pour 100% des réalisations lorsque $N_e \leq 1024$. On remarque que pour $N_e = 256$, le pourcentage de convergence, ainsi que la vitesse de convergence (voir figure 5.10) de ces deux algorithmes sont mauvais. T-ALUJA-A, C-CLUJA, G-CLUJA et ALUJA sont toujours, quant à eux, les algorithmes nécessitant le moins d'itérations pour converger, quel que soit le nombre d'échantillons considérés.

La figure 5.11 illustre la dégradation des performances des algorithmes lorsque les ensembles cibles sont calculés à partir d'un faible nombre d'échantillons. En effet, lorsque

$N_e = 256$, le signal obtenu avec ALUJA ne permet pas d'identifier une seule et unique constellation QPSK, contrairement aux signaux obtenus avec T-ALUJA-A et C-CLUJA.

5.4 Modulation 4-ASK et cas hermitien/INDSCAL_h

Pour cette modulation, on utilise les cumulants d'ordre quatre (2.45) afin de travailler sur la diagonalisation conjointe d'ensemble de matrices hermitiennes et d'ensemble de tenseurs d'ordre trois INDSCAL_h. Dans ce paragraphe, on compare les algorithmes proposés (G-CLUJA, A-CLUJA, C-CLUJA, ALUJA, T-ALUJA et T-ALUJA-A) aux algorithmes de diagonalisation conjointe de matrices LUCJD [127], JTJD [56], CJD_i [78] et U-WEDGE [114] et aux algorithmes de décomposition canonique polyadique CPD-ALS [105] et CPD-LM [106].

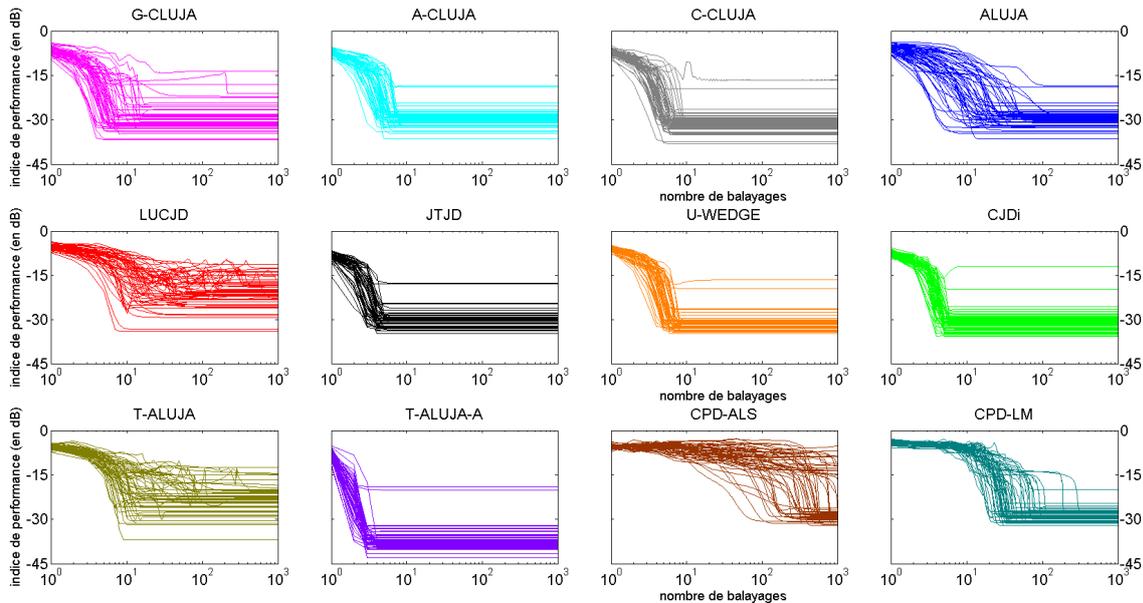


FIGURE 5.12 – Indice de performance de chacun des algorithmes pour toutes les réalisations en fonction du nombre de balayages. Contexte : 8 signaux d'observations de 16384 échantillons, constitués d'un mélange linéaire de 7 sources respectant une modulation 4-ASK et avec un RSB_{dB} de 30dB.

Dans un premier temps, les figures 5.12 et 5.13 exposent le comportement général des algorithmes pour 8 signaux d'observations de 16384 échantillons, constitués d'un mélange linéaire de 7 sources et avec un rapport signal sur bruit de 30dB. T-ALUJA-A fournit, comme pour le modèle symétrique, les meilleurs résultats, aussi bien en termes de performances atteintes qu'en termes de vitesse de convergence. Parmi les algorithmes matriciels, U-WEDGE et C-CLUJA, dont les indices de performance moyens après convergence valent $-29dB$, af-

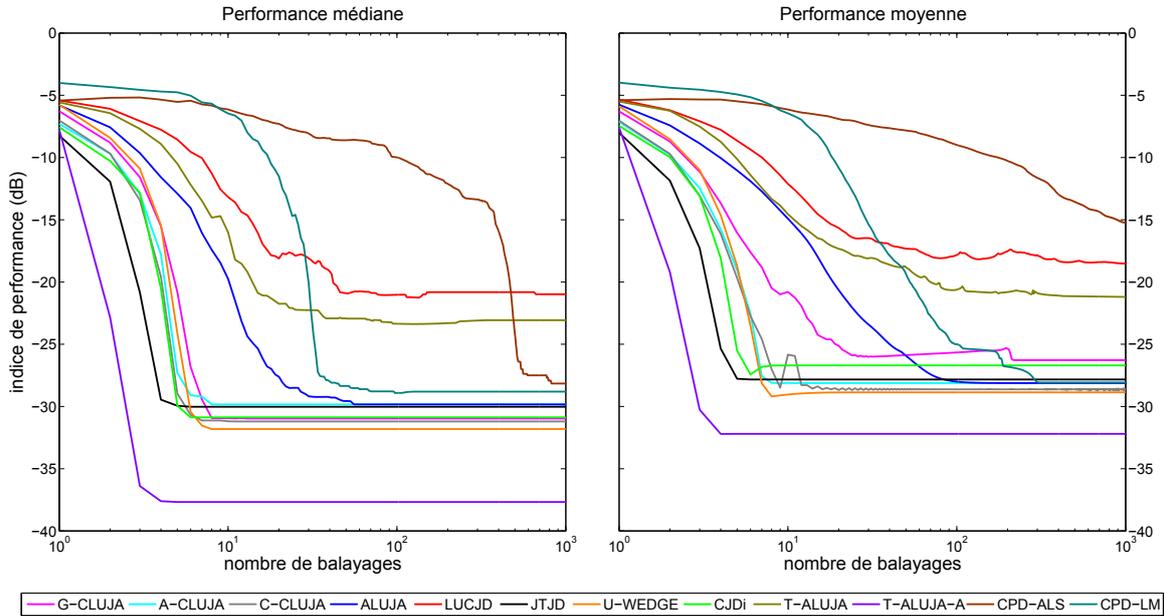


FIGURE 5.13 – Indices de performance médians et moyens des algorithmes (50 réalisations) en fonction du nombre de balayages pour 8 signaux d’observations de 16384 échantillons, constitués d’un mélange linéaire de 7 sources respectant une modulation 4-ASK et avec un RSB_{dB} de 30dB.

fichent les meilleurs niveaux de performance. JTJD est l’algorithme matriciel présentant la meilleure vitesse de convergence. En effet, cet algorithme converge en 5 itérations, soit seulement une itération de plus que T-ALUJA-A. Concernant les algorithmes de décomposition CP, les indices de performances médian de CPD-ALS et CPD-LM sont supérieurs à -30 dB. La vitesse de convergence de CPD-LM est moyenne tandis que celle de CPD-ALS est mauvaise. Etudions, maintenant, l’impact des différents paramètres sur ces algorithmes.

5.4.1 Influence du rapport signal sur bruit

La figure 5.14 montre les performances médianes atteintes par les algorithmes en fonction du rapport signal sur bruit. Comme précédemment, pour un RSB_{dB} supérieur ou égal à 0dB, les indices de performance des différents algorithmes sont mauvais. De plus, neuf des douze algorithmes étudiés n’obtiennent pas un bon pourcentage de convergence (voir tableau 5.5). Avec un RSB_{dB} de 10dB, les CPD-LM et CPD-ALS atteignent les meilleurs niveaux de performance avec -21 dB, devant U-WEDGE et T-ALUJA-A avec -18 dB. Enfin lorsque le rapport signal sur bruit est inférieur à 10dB, T-ALUJA-A devient environ 5dB meilleur que respectivement U-WEDGE, C-CLUJA, CJDj et G-CLUJA.

La figure 5.15 permet d’évaluer la vitesse de convergence médiane des algorithmes en

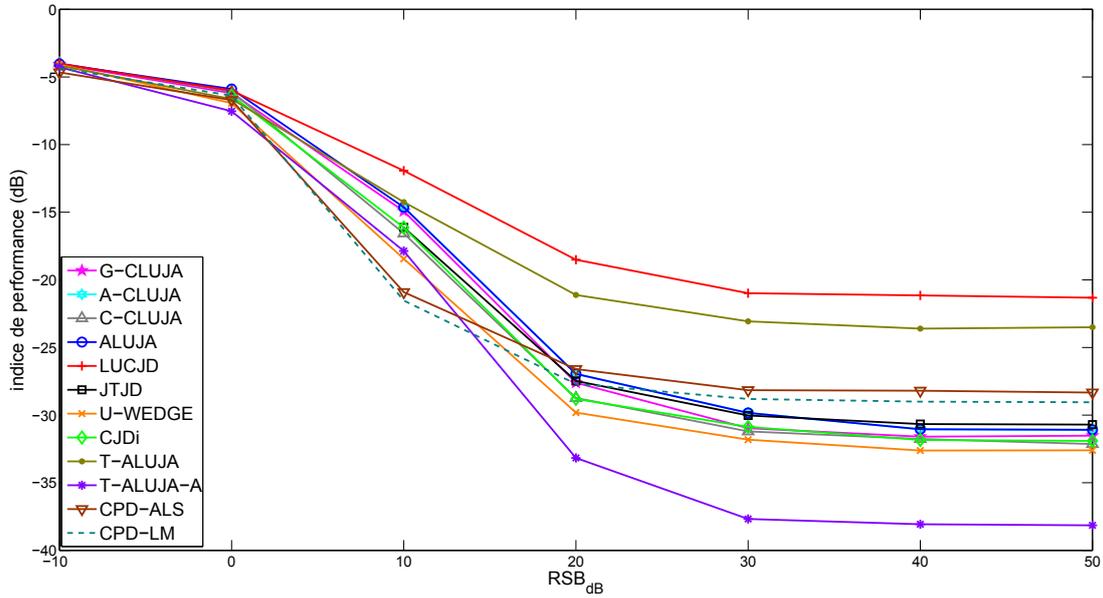


FIGURE 5.14 – Indice de performance médian (50 réalisations) après 1000 balayages en fonction du RSB_{dB} pour 8 signaux d’observations de 16384 échantillons, constitués d’un mélange linéaire de 7 sources respectant une modulation 4-ASK.

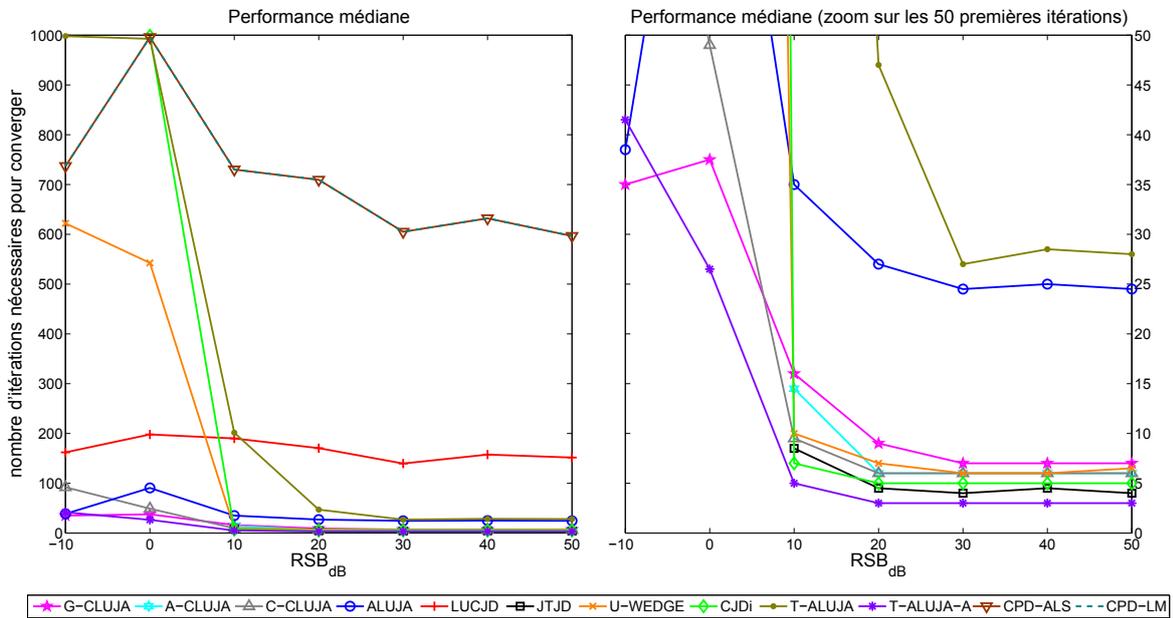


FIGURE 5.15 – Vitesse de convergence médiane (50 réalisations) en fonction du RSB_{dB} pour 8 signaux d’observations de 16384 échantillons, constitués d’un mélange linéaire de 7 sources respectant une modulation 4-ASK.

RSB _{dB}	-10dB	0dB	10dB	20dB	30dB	40dB	50dB
A-CLUJA	2%	24%	92%	96%	100%	100%	100%
C-CLUJA	92%	98%	98%	100%	100%	100%	100%
JTJD	0%	0%	80%	92%	100%	100%	100%
U-WEDGE	100%	96%	98%	100%	100%	100%	100%
CJDi	30%	58%	96%	98%	100%	100%	100%
T-ALUJA	76%	90%	100%	98%	100%	100%	100%
T-ALUJA-A	98%	100%	100%	100%	100%	100%	100%
CPD-ALS	96%	92%	90%	96%	90%	90%	92%
CPD-LM	100%	98%	100%	100%	100%	100%	100%

TABLE 5.5 – Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du RSB_{dB} pour 8 signaux d’observations de 16384 échantillons, constitués d’un mélange linéaire de 7 sources respectant une modulation 4-ASK (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les RSB_{dB}).

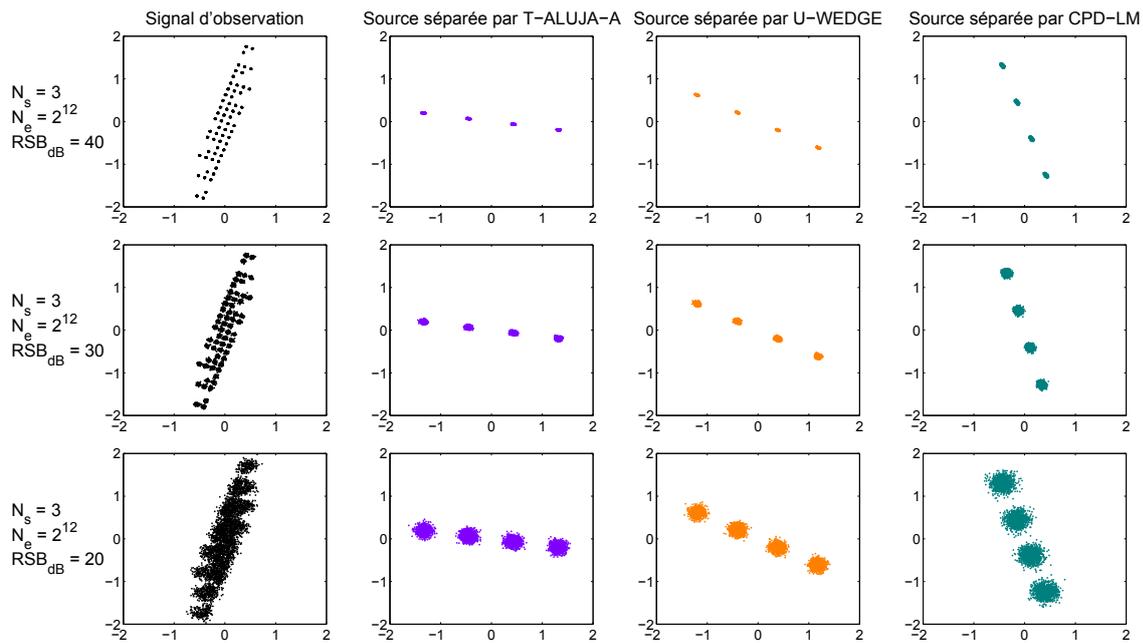


FIGURE 5.16 – Exemple explicite de séparation de sources en fonction du RSB_{dB} pour 5 signaux d’observations de 4096 échantillons, constitués d’un mélange linéaire de 3 sources respectant une modulation 4-ASK. De gauche à droite : un des signaux d’observations puis une des sources séparée respectivement par T-ALUJA-A, U-WEDGE et CPD-LM

fonction du rapport signal sur bruit. La présence d’un bruit important ($\text{RSB}_{\text{dB}} < 10\text{dB}$) dégrade, sans surprise, les vitesses de convergence de la plupart des algorithmes. Pour des rapports signal sur bruit inférieurs ou égaux à 10dB, T-ALUJA-A, qui converge en 3 itérations, est légèrement plus rapide que JTJD, CJDi, C-CLUJA, A-CLUJA, U-WEDGE et G-CLUJA.

Pour cette plage de bruit, le tableau 5.5 révèle que CPD-ALS, A-CLUJA mais aussi JTJD ont des pourcentages de convergence qui seront qualifiés de moyens.

La figure 5.16 donne un exemple de séparation de sources effectuée par T-ALUJA-A, U-WEDGE et CPD-LM pour trois RSB_{dB} . Dans chacun des cas, les trois algorithmes fournissent bien une source constituée d'une constellation de quatre états distincts représentant un signal 4-ASK. Evidemment, plus le RSB_{dB} diminue, plus les ensembles de points correspondant chacun des quatre états de la constellation 4-ASK sont dispersés. Notons que cette dispersion est plus importante pour CPD-LM que pour T-ALUJA-A et U-WEDGE. Ceci confirme le fait que, pour un rapport signal sur bruit compris entre 20dB et 40dB, les indices de performance de ces deux derniers algorithmes sont toujours meilleurs que celui de CPD-LM (voir figure 5.14).

5.4.2 Influence du nombre de sources

La figure 5.17 illustre les niveaux de performance médian des algorithmes en fonction du nombre de sources. Comme dans le cas de la modulation QPSK, pour moins de 8 sources, les algorithmes CPD présentent les meilleurs niveaux de performance, mais pas les meilleures vitesses de convergence, comme l'expose la figure 5.18. Puis, lorsque N_s augmente, leurs performances se dégradent. Ainsi, pour 15 sources, CPD-LM converge en 78 itérations et son indice de performance vaut $-17dB$. Pour un nombre de sources supérieur ou égal à 8, T-ALUJA-A et U-WEDGE surpassent les autres algorithmes en termes de niveau de performance atteint. Cependant, T-ALUJA-A nécessite moins d'itérations pour converger. Notons que pour deux sources, JTJD et CJDi convergent le plus rapidement car ils sont basés, dans ce cas précis, sur une GEVD.

Sur la figure 5.19, on constate que pour 4096 échantillons et un RSB_{dB} de 30dB, dès cinq sources, le signal 4-ASK obtenu avec CPD-LM présente une dispersion importante des points constituant constellation. Même si les quatre états de cette constellation 4-ASK se distinguent à l'œil nu, la localisation de certains points du signal ne permet pas de déterminer de façon exacte auxquels des quatre états ils appartiennent.

5.4.3 Influence du nombre d'échantillons considérés

Intéressons nous, enfin, à l'effet du nombre d'échantillons utilisés pour calculer les cumulants (2.45). Les figures 5.20 et 5.21, ainsi que le tableau 5.6, illustrent, respectivement, le niveau de performance médian, la vitesse de convergence et le pourcentage de convergence en moins de 1000 itérations en fonction du nombre d'échantillons. Ce paramètre influe globalement de la même façon sur tous les algorithmes en termes de niveau de performance : plus on

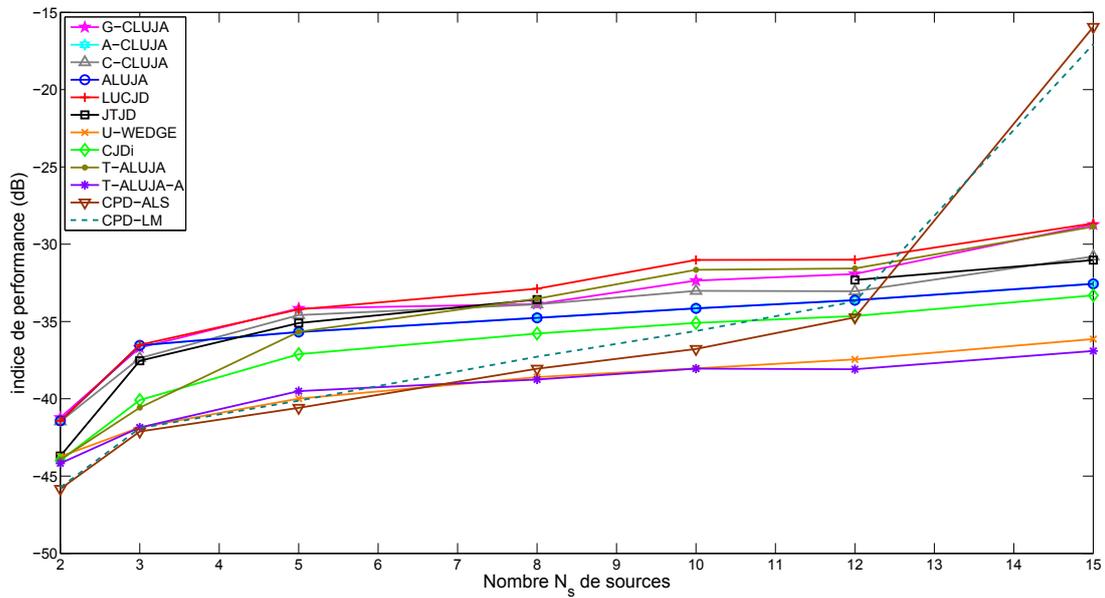


FIGURE 5.17 – Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_s de sources de modulation 4-ASK. Contexte : 15 signaux d’observations de 16384 échantillons avec un RSB_{dB} de 20dB.

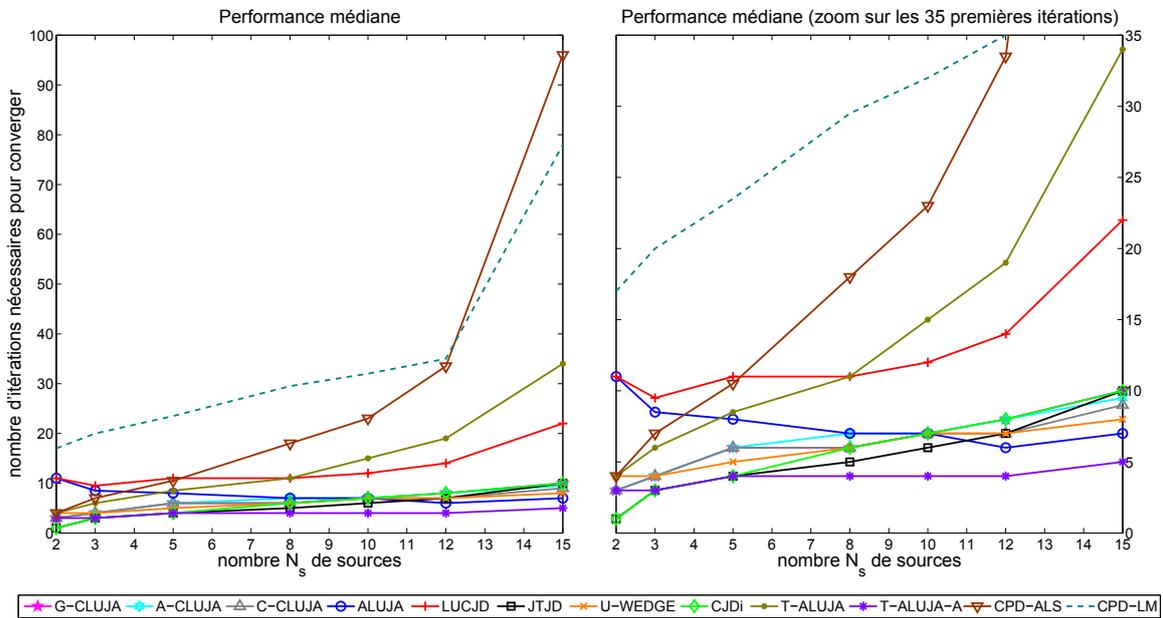


FIGURE 5.18 – Vitesse de convergence médiane (50 réalisations) en fonction du nombre N_s de sources de modulation 4-ASK pour 15 signaux d’observations de 16384 échantillons avec un RSB_{dB} de 20dB.

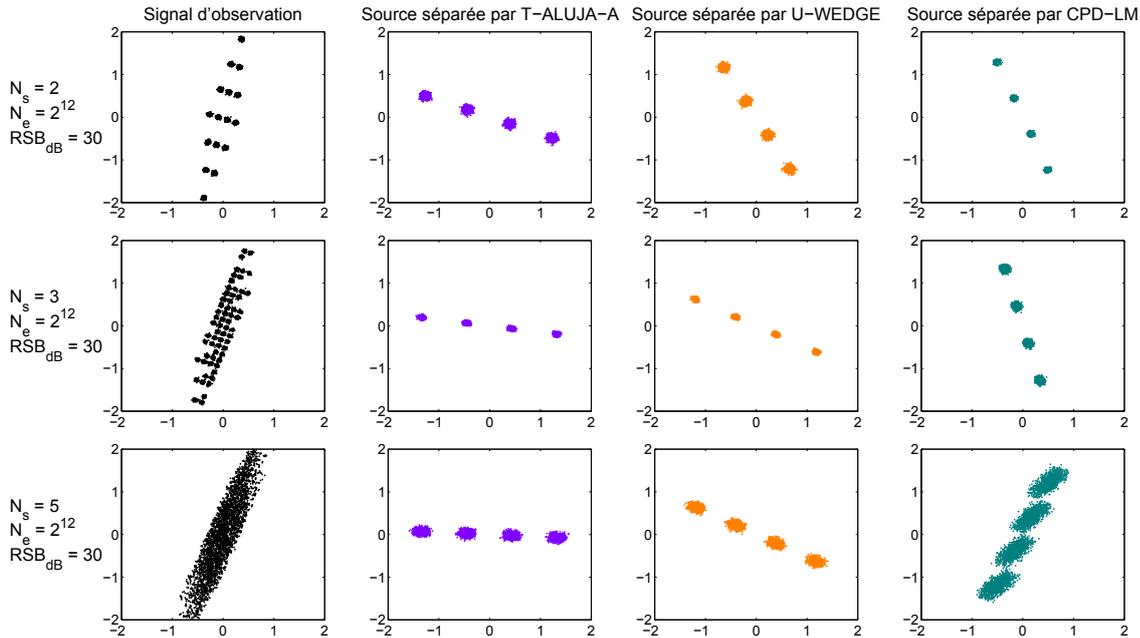


FIGURE 5.19 – Exemple explicite de séparation de sources en fonction du nombre N_s de sources respectant une modulation 4-ASK pour 5 signaux d’observations de 4096 échantillons avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d’observations puis une des sources séparée respectivement par T-ALUJA-A, U-WEDGE et CPD-LM

prend d’échantillons, plus faibles seront les valeurs atteintes par l’indice de performance. Encore une fois, T-ALUJA-A et U-WEDGE sont les algorithmes affichant les meilleurs niveaux de performance, pour tout $N_e \geq 256$.

Concernant la vitesse de convergence, comme pour la modulation QPSK, moins on considère d’échantillons, plus le nombre d’itérations nécessaires pour que les algorithmes convergent augmente. Par exemple, pour $N_e = 64$, CPD-ALS converge en plus de 500 itérations, CPD-LM en 77 itérations, T-ALUJA en 74 itérations, U-WEDGE en 58 itérations et LUCJD en 54 itérations. Dans ce même cas, A-CLUJA et JTJD divergent. Les autres algorithmes sont nettement moins influencés par N_e . Enfin, quel que soit le nombre d’échantillons considérés, T-ALUJA-A possède toujours la meilleure vitesse de convergence.

La figure 5.22 montre la dégradation des performances de CPD-LM lorsque $N_e = 256$. En effet, on constate une dispersion particulière (dans une direction précise) des points du signal obtenu avec CPD-LM. Ceci atteste d’une dépendance entre plusieurs signaux 4-ASK et confirme le mauvais indice de performance de l’algorithme pour ce nombre d’échantillons considérés.

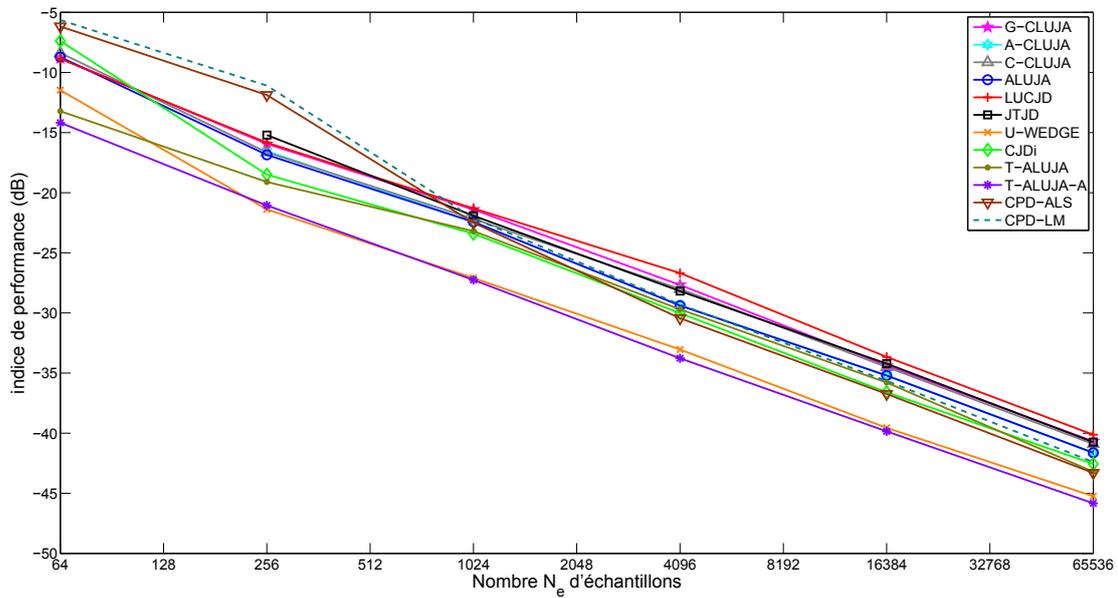


FIGURE 5.20 – Indice de performance médian (50 réalisations) après 1000 balayages en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation 4-ASK avec un RSB_{dB} de 20dB.

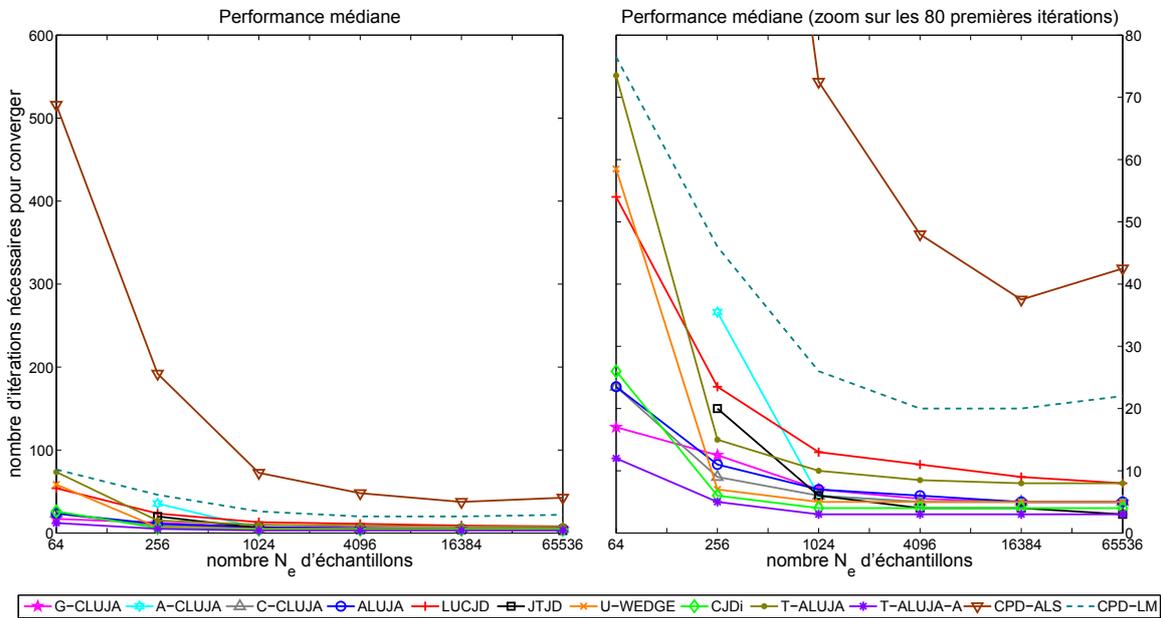


FIGURE 5.21 – Vitesse de convergence médiane (50 réalisations) en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation 4-ASK avec un RSB_{dB} de 20dB.

N_e	64	256	1024	4096	16384	65536
A-CLUJA	38%	92%	100%	100%	100%	100%
JTJD	18%	74%	96%	100%	100%	100%
CJDi	64%	94%	100%	100%	100%	100%
CPD-ALS	86%	92%	96%	96%	100%	98%

TABLE 5.6 – Pourcentage de réalisations convergeant en moins de 1000 itérations en fonction du nombre N_e d'échantillons pour 6 signaux d'observations constitués d'un mélange linéaire de 5 sources respectant une modulation 4-ASK avec un RSB_{dB} de 20dB (ne sont répertoriés, ici, que les algorithmes ne présentant pas 100% de convergence pour toutes les réalisations et tous les N_e).

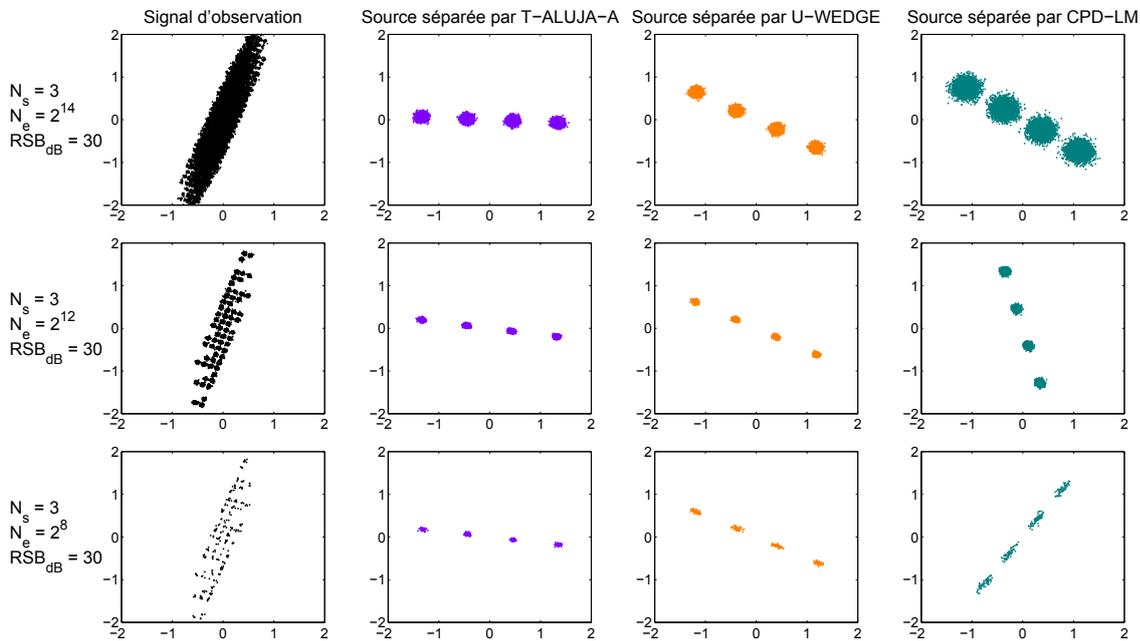


FIGURE 5.22 – Exemple explicite de séparation de sources en fonction du nombre N_e d'échantillons pour 5 signaux d'observations constitués d'un mélange linéaire de 3 sources respectant une modulation 4-ASK avec un RSB_{dB} de 30dB. De gauche à droite : un des signaux d'observations puis une des sources séparée respectivement par T-ALUJA-A, U-WEDGE et CPD-LM

5.5 Conclusion

Dans ce chapitre, nous avons pu voir que les algorithmes développés au cours de cette thèse peuvent être appliqués à la séparation de sources. L'algorithme de diagonalisation conjointe de tenseurs d'ordre trois T-ALUJA-A obtient des résultats très intéressants, que ce soit en termes de niveau de performance, de vitesse de convergence ou encore de pourcentage de convergence. De plus, T-ALUJA-A se compare favorablement aux algorithmes de diagonalisation conjointe de matrices et de décomposition CP de la littérature.

Conclusion et perspectives

Conclusion

Au cours de cette thèse, nous avons développé de nouveaux algorithmes de diagonalisation conjointe non-unitaire de matrices et de tenseurs d'ordre trois. Le but principal était de traiter des signaux complexes à partir de leurs informations statistiques. Dans ces cas précis, les tenseurs rencontrés présentent des symétries particulières. En effet, à l'ordre deux, on rencontre des matrices symétriques ou hermitiennes, tandis qu'à l'ordre trois on dispose de tenseurs symétriques ou admettant une décomposition INDSCAL_h .

Les algorithmes proposés sont de type Jacobi et reposent sur l'optimisation d'un critère inverse classique et sur une paramétrisation LU de la matrice diagonalisante à estimer. Nous avons envisagé l'optimisation de ce critère selon deux approches : la première, de manière directe, et la seconde, en supposant être proche d'une solution diagonalisante. Nous avons aussi proposé deux stratégies d'estimation des paramètres de la matrice diagonalisante : la première consiste à évaluer chacun des paramètres du problème indépendamment, tandis que pour la seconde, on se focalise sur l'estimation de couples de paramètres judicieusement choisis.

Ainsi, dans le cadre matriciel, nous avons développé trois algorithmes de diagonalisation conjointe non-unitaire fonctionnant pour des décompositions symétriques ou hermitiennes. Ces algorithmes reposent sur trois méthodes de résolution distinctes. Pour ALUJA, on recherche simplement la racine de la dérivée du critère permettant de le minimiser. Pour A-CLUJA, le critère s'écrit comme une forme quadratique et sa minimisation est assurée par la recherche du vecteur propre associé à la plus petite valeur propre d'une matrice définie non-négative. Enfin, pour G-CLUJA, on met en place une méthode de gradient avec calcul d'un pas optimal. Les deux derniers algorithmes ont été combinés pour donner C-CLUJA.

Dans le cadre tensoriel d'ordre trois, nous avons développé deux algorithmes de diagonalisation conjointe non-unitaire, fonctionnant pour les modèles symétrique, INDSCAL_h et non-symétrique. Les algorithmes proposés appliqués à ce dernier modèle ont été comparés

avec des algorithmes de décomposition canonique polyadique d'un tenseur d'ordre quatre. T-ALUJA et T-ALUJA-A reposent sur une stratégie d'estimation découplée des paramètres qui permet une résolution analytique du problème. L'optimisation du critère se fait par la recherche de la racine de sa dérivée.

Les algorithmes proposés ont été validés pour des ensembles synthétiques satisfaisant des modèles de décomposition exact bruités. Dans le cadre matriciel, C-CLUJA se compare favorablement à l'état de l'art par son niveau de performance et par son pourcentage de convergence. Il nécessite aussi très peu d'itérations pour converger. Dans le cas tensoriel, T-ALUJA-A se compare favorablement aux algorithmes de décomposition canonique polyadique, tant par sa vitesse de convergence que par sa complexité ou son pourcentage de convergence.

Ces algorithmes de diagonalisation conjointe non-unitaire trouvent une application directe à la séparation de sources lorsque celles-ci sont statistiquement indépendantes. Nous avons ainsi pu confronter les algorithmes proposés à des signaux synthétiques de télécommunications numériques. Dans ce cas pratique, T-ALUJA-A fournit des résultats particulièrement intéressants en termes de niveau de performance, de vitesse de convergence et de pourcentage de convergence.

Perspectives

Au cours de ces trois années, de nombreuses simulations ont été effectuées, dans des contextes variés où les différents paramètres des problèmes ont été étudiés. Parmi les algorithmes développés selon l'approche adaptée du critère d'optimisation, ALUJA et T-ALUJA-A ont, globalement, toujours obtenu de très bonnes performances et notamment en termes de convergence. Il serait toutefois intéressant d'étudier les questions de convergence de ces algorithmes et, si possible, de faire le lien entre la décroissance des critères d'optimisation classique et adapté.

Pour tous les algorithmes proposés, nous avons considéré une paramétrisation LU de la matrice diagonalisante. Il pourrait être intéressant d'étudier d'autres types de décompositions, comme nous avons commencé à le faire dans l'article de conférence MLSP'2014. Dans ce dernier, nous avons proposé un algorithme de diagonalisation conjointe non-unitaire de matrices réelles reposant sur une paramétrisation QR.

Les algorithmes proposés permettent de traiter les cas où le nombre de sources est inférieur ou égal au nombre d'observations. Il pourrait être intéressant de développer les mêmes types de décompositions pour le critère direct afin de pouvoir diagonaliser conjointement des problèmes sous-déterminés.

En ce qui concerne l'application à la séparation de sources, nous nous sommes intéressés

à des mélanges linéaires instantanés. Il pourrait être intéressant d'aborder le cas de mélanges convolutifs et ainsi généraliser les algorithmes proposés à la bloc-diagonalisation conjointe.

Enfin, les algorithmes de diagonalisation conjointe non-unitaire de tenseurs d'ordre trois proposés reposent sur une stratégie d'estimation découplée des paramètres de la matrice diagonalisante. Il serait intéressant d'étudier une stratégie couplée pour ce type de problème.

Bibliographie

- [1] B. Afsari, “Simple LU and QR based non-orthogonal matrix joint diagonalization,” in *Independent Component Analysis and Blind Signal Separation*. Springer, 2006, pp. 1–7.
- [2] B. Afsari, “Sensitivity analysis for the problem of matrix joint diagonalization,” in *SIAM Journal on Matrix Analysis and Applications*, Vol. 30, No. 3, pp. 1148–1171, 2008.
- [3] B. Afsari and P. S. Krishnaprasad, “Some gradient based joint diagonalization methods for ICA,” in *Independent Component Analysis and Blind Signal Separation*. Springer, 2004, pp. 437–444.
- [4] L. Albera, A. Ferréol, P. Comon, and P. Chevalier, “Sixth order blind identification of underdetermined mixtures (BIRTH) of sources,” in *ICA*, Vol. 3, pp. 1–4, 2003.
- [5] L. Albera, A. Kachenoura, P. Comon, A. Karfoul, F. Wendling, L. Senhadji, and I. Merlet, “ICA-based EEG denoising : a comparative analysis of fifteen methods,” in *Bulletin of the Polish Academy of Sciences : Technical Sciences*, Vol. 60, No. 3, pp. 407–418, 2012.
- [6] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, “A blind source separation technique using second-order statistics,” in *Signal Processing, IEEE Transactions on*, Vol. 45, No. 2, pp. 434–444, 1997.
- [7] A. Belouchrani and M. G. Amin, “Blind source separation based on time-frequency signal representations,” in *Signal Processing, IEEE Transactions on*, Vol. 46, No. 11, pp. 2888–2897, 1998.
- [8] A. Belouchrani, M. G. Amin, N. Thirion-Moreau, and Y. D. Zhang, “Source Separation and Localization Using Time-Frequency Distributions : An Overview,” in *Signal Processing Magazine, IEEE*, Vol. 30, No. 6, pp. 97–107, 2013.

- [9] J.-F. Cardoso, "Source separation using higher order moments," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, IEEE, pp. 2109–2112, 1989.
- [10] J.-F. Cardoso, "Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, IEEE, pp. 3109–3112, 1991.
- [11] J.-F. Cardoso, "Fourth-order cumulant structure forcing : application to blind array processing," in *Statistical Signal and Array Processing, 1992. Conference Proceedings., IEEE Sixth SP Workshop on*, IEEE, pp. 136–139, 1992.
- [12] J.-F. Cardoso, "Iterative techniques for blind source separation using only fourth-order cumulants," in *Proc. EuSIPCO*, Vol. 92, pp. 739–742, 1992.
- [13] J.-F. Cardoso *et al.*, "On the performance of orthogonal source separation algorithms," in *Proc. EUSIPCO*, Vol. 94, Edinburgh, UK, pp. 776–779, 1994.
- [14] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," in *IEE Proceedings F (Radar and Signal Processing)*, Vol. 140, No. 6, IET, pp. 362–370, 1993.
- [15] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," in *SIAM journal on matrix analysis and applications*, Vol. 17, No. 1, pp. 161–164, 1996.
- [16] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Youn" decomposition," in *Psychometrika*, Vol. 35, No. 3, pp. 283–319, 1970.
- [17] J. D. Carroll and P. E. Green, "An INDSCAL-based approach to multiple correspondence analysis," in *Journal of Marketing Research*, pp. 193–203, 1988.
- [18] M. Castella, P. Bianchi, A. Chevreuil, and J.-C. Pesquet, "A blind source separation framework for detecting CPM sources mixed by a convolutive MIMO filter," in *Signal Processing*, Vol. 86, No. 8, pp. 1950–1967, 2006.
- [19] M. Castella and E. Moreau, "Generalized identifiability conditions for blind convolutive MIMO separation," in *Signal Processing, IEEE Transactions on*, Vol. 57, No. 7, pp. 2846–2852, 2009.
- [20] G. Chabriel and J. Barrère, "An instantaneous formulation of mixtures for blind separation of propagating waves," in *Signal Processing, IEEE Transactions on*, Vol. 54, No. 1, pp. 49–58, 2006.

- [21] G. Chabriel and J. Barrere, “Non-symmetrical joint zero-diagonalization and mimo zero-division multiple access,” *in* Signal Processing, IEEE Transactions on, Vol. 59, No. 5, pp. 2296–2307, 2011.
- [22] G. Chabriel and J. Barrere, “A direct algorithm for nonorthogonal approximate joint diagonalization,” *in* Signal Processing, IEEE Transactions on, Vol. 60, No. 1, pp. 39–47, 2012.
- [23] G. Chabriel, J. Barrère, N. Thirion-Moreau, and E. Moreau, “Algebraic joint zero-diagonalization and blind sources separation,” *in* Signal Processing, IEEE Transactions on, Vol. 56, No. 3, pp. 980–989, 2008.
- [24] G. Chabriel, M. Kleinstuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor, “Joint matrices decompositions and blind source separation : A survey of methods, identification, and applications,” *in* Signal Processing Magazine, IEEE, Vol. 31, No. 3, pp. 34–43, 2014.
- [25] P. Chevalier, L. Albera, P. Comon, and A. Ferréol, “Comparative performance analysis of eight blind source separation methods on radiocommunications signals,” *in* *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, Vol. 1, IEEE, 2004.
- [26] P. CHEVALIER and P. COMON, “Séparation aveugle de sources,” *in* Techniques de l’ingénieur. Télécoms, No. TE5250, pp. TE5250–1, 2002.
- [27] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications : From two-way to multiway component analysis,” *in* Signal Processing Magazine, IEEE, Vol. 32, No. 2, pp. 145–163, 2015.
- [28] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.
- [29] P. Comon, “Independent component analysis, a new concept ?” *in* Signal processing, Vol. 36, No. 3, pp. 287–314, 1994.
- [30] P. Comon, “Tensors : a brief introduction,” *in* IEEE Signal Processing Magazine, Vol. 31, No. 3, pp. 44–53, 2014.
- [31] P. Comon and C. Jutten, *Handbook of Blind Source Separation : Independent component analysis and applications*. Academic press, 2010.
- [32] P. Comon, E. Moreau, and L. Rota, “BLIND SEPARATION OF CONVOLUTIVE MIXTURES A CONTRAST-BASED JOINT DIAGONALIZATION APPROACH,” *in* notes, Vol. 9, p. 6, 2001.

- [33] M. Congedo, C. Gouy-Pailler, and C. Jutten, “On the blind source separation of human electroencephalogram by approximate joint diagonalization of second order statistics,” *in* *Clinical Neurophysiology*, Vol. 119, No. 12, pp. 2677–2686, 2008.
- [34] M. Congedo, R. Phlypo, and J. Chatel-Goldman, “Orthogonal and non-orthogonal joint blind source separation in the least-squares sense,” *in* *20th European Signal Processing Conference (EUSIPCO-2012)*, pp. 1885–1889, 2012.
- [35] M. Congedo, R. Phlypo, and D.-T. Pham, “Approximate joint singular value decomposition of an asymmetric rectangular matrix set,” *in* *Signal Processing, IEEE Transactions on*, Vol. 59, No. 1, pp. 415–424, 2011.
- [36] L. De Lathauwer and J. Castaing, “Blind identification of underdetermined mixtures by simultaneous matrix diagonalization,” *in* *Signal Processing, IEEE Transactions on*, Vol. 56, No. 3, pp. 1096–1105, 2008.
- [37] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, “Fourth-order cumulant-based blind identification of underdetermined mixtures,” *in* *Signal Processing, IEEE Transactions on*, Vol. 55, No. 6, pp. 2965–2973, 2007.
- [38] L. De Lathauwer and B. De Moor, “On the blind separation of non-circular sources,” *in* *Signal Processing Conference, 2002 11th European*, IEEE, pp. 1–4, 2002.
- [39] L. De Lathauwer, B. De Moor, and J. Vandewalle, “ICA techniques for more sources than sensors,” *in* *Higher-Order Statistics, 1999. Proceedings of the IEEE Signal Processing Workshop on*, IEEE, pp. 121–124, 1999.
- [40] L. De Lathauwer, B. De Moor, and J. Vandewalle, “Fetal electrocardiogram extraction by blind source subspace separation,” *in* *IEEE transactions on biomedical engineering*, Vol. 47, No. 5, pp. 567–572, 2000.
- [41] L. De Lathauwer, B. De Moor, and J. Vandewalle, “Independent component analysis and (simultaneous) third-order tensor diagonalization,” *in* *Signal Processing, IEEE Transactions on*, Vol. 49, No. 10, pp. 2262–2271, 2001.
- [42] S. Dégerine and E. Kane, “A comparative study of approximate joint diagonalization algorithms for blind source separation in presence of additive noise,” *in* *Signal Processing, IEEE Transactions on*, Vol. 55, No. 6, pp. 3022–3031, 2007.
- [43] J. Delabrouille, J.-F. Cardoso, and G. Patanchon, “Multidetector multicomponent spectral matching and applications for cosmic microwave background data analysis,” *in* *Monthly Notices of the Royal Astronomical Society*, Vol. 346, No. 4, pp. 1089–1102, 2003.
- [44] N. Delfosse and P. Loubaton, “Adaptive blind separation of independent sources : a deflation approach,” *in* *Signal processing*, Vol. 45, No. 1, pp. 59–83, 1995.

- [45] R. Dubroca, C. De Luigi, M. Castella, and E. Moreau, "A general algebraic algorithm for blind extraction of one source in a MIMO convolutive mixture," *in* *Signal Processing*, IEEE Transactions on, Vol. 58, No. 5, pp. 2484–2493, 2010.
- [46] R. Dubroca, C. De Luigi, and E. Moreau, "Adapted deflation approach for referenced contrast optimization in blind MIMO convolutive source separation," *in* *Independent Component Analysis and Signal Separation*. Springer, 2009, pp. 243–250.
- [47] E. M. Fadaili, N. Thirion-Moreau, and E. Moreau, "Nonorthogonal joint diagonalization/zero diagonalization for source separation based on time-frequency distributions," *in* *Signal Processing*, IEEE Transactions on, Vol. 55, No. 5, pp. 1673–1687, 2007.
- [48] A. Ferréol, L. Albera, and P. Chevalier, "Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM)," *in* *Signal Processing*, IEEE Transactions on, Vol. 53, No. 5, pp. 1640–1653, 2005.
- [49] A. Ferréol, P. Chevalier, and L. Albera, "Second-order blind separation of first- and second-order cyclostationary sources-application to AM, FSK, CPFSK, and deterministic sources," *in* *Signal Processing*, IEEE Transactions on, Vol. 52, No. 4, pp. 845–861, 2004.
- [50] B. N. Flury, "Common principal components in k groups," *in* *Journal of the American Statistical Association*, Vol. 79, No. 388, pp. 892–898, 1984.
- [51] H. Ghennioui, E. M. Fadaili, N. Thirion-Moreau, A. Adib, and E. Moreau, "A nonunitary joint block diagonalization algorithm for blind separation of convolutive mixtures of sources," *in* *Signal Processing Letters*, IEEE, Vol. 14, No. 11, pp. 860–863, 2007.
- [52] H. Ghennioui, N. Thirion-Moreau, E. Moreau, and D. Aboutajdine, "Gradient-based joint block diagonalization algorithms : Application to blind separation of FIR convolutive mixtures," *in* *Signal Processing*, Vol. 90, No. 6, pp. 1836–1849, 2010.
- [53] L. Giulieri, H. Ghennioui, N. Thirion-Moreau, and E. Moreau, "Nonorthogonal joint diagonalization of spatial quadratic time-frequency matrices for source separation," *in* *Signal Processing Letters*, IEEE, Vol. 12, No. 5, pp. 415–418, 2005.
- [54] L. Giulieri, N. Thirion-Moreau, and P.-Y. Arques, "Blind sources separation based on bilinear time-frequency representations : a performance analysis," *in* *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, Vol. 2, IEEE, pp. II–1649, 2002.
- [55] X.-F. Gong, X.-L. Wang, and Q.-H. Lin, "Generalized Non-orthogonal Joint Diagonalization with LU Decomposition and Successive Rotations," *in* *Signal Processing*, IEEE Transactions on, Vol. 63, No. 5, pp. 1322–1334, 2015.

- [56] X. Guo, S. Zhu, S. Miron, and D. Brie, "Approximate joint diagonalization by nonorthogonal nonparametric Jacobi transformations," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, IEEE, pp. 3774–3777, 2010.
- [57] R. A. Harshman, "Foundations of the PARAFAC procedure : Models and conditions for an explanatory multi-modal factor analysis," Vol. 16, pp. 1–84, 1970.
- [58] G. Hellboug, T. Trainini, R. Weber, E. Moreau, C. Capdessus, and A. Boonstrd, "RFI subspace estimation techniques for new generation radio telescopes," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, IEEE, pp. 200–204, 2012.
- [59] J. Héroult, C. Jutten, and B. Ans, "Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé," in *10^e Colloque sur le traitement du signal et des images, FRA, 1985*, GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 1985.
- [60] X. Huang, H.-C. Wu, and J. C. Principe, "Robust blind beamforming algorithm using joint multiple matrix diagonalization," in *Sensors Journal*, IEEE, Vol. 7, No. 1, pp. 130–136, 2007.
- [61] R. Iferroudjene, K. Abed-Meraim, and A. Belouchrani, "Joint diagonalization of non defective matrices using generalized Jacobi rotations," in *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, IEEE, pp. 345–348, 2010.
- [62] M. Joho, "Newton method for joint approximate diagonalization of positive definite hermitian matrices," in *SIAM Journal on Matrix Analysis and Applications*, Vol. 30, No. 3, pp. 1205–1218, 2008.
- [63] C. Jutten and J. Héroult, "Blind separation of sources, part I : An adaptive algorithm based on neuromimetic architecture," in *Signal processing*, Vol. 24, No. 1, pp. 1–10, 1991.
- [64] A. Karfoul, L. Albera, and G. Birot, "Blind underdetermined mixture identification by joint canonical decomposition of HO cumulants," in *Signal Processing, IEEE Transactions on*, Vol. 58, No. 2, pp. 638–649, 2010.
- [65] Z. Koldovský and P. Tichavský, "Time-domain blind separation of audio sources on the basis of a complete ICA decomposition of an observation space," in *Audio, Speech, and Language Processing, IEEE Transactions on*, Vol. 19, No. 2, pp. 406–416, 2011.
- [66] D. Lahat, J.-F. Cardoso, and H. Messer, "Second-order multidimensional ICA : Performance analysis," in *Signal Processing, IEEE Transactions on*, Vol. 60, No. 9, pp. 4598–4610, 2012.

- [67] B. Laheld and J.-F. Cardoso, "Adaptive source separation without prewhitening," in *Proc. EUSIPCO, (Edinburgh)*, pp. 183–186, 1994.
- [68] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," 1944.
- [69] X.-L. Li and T. Adali, "Blind separation of noncircular correlated sources using Gaussian entropy rate," in *Signal Processing, IEEE Transactions on*, Vol. 59, No. 6, pp. 2969–2975, 2011.
- [70] X.-L. Li and X.-D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solution," in *Signal Processing, IEEE Transactions on*, Vol. 55, No. 5, pp. 1803–1814, 2007.
- [71] X. Li, X. Zhang, and P. Li, "Gradient Search non-orthogonal approximate joint diagonalization algorithm," in *Tsinghua Science & Technology*, Vol. 12, No. 6, pp. 669–673, 2007.
- [72] X. Liu, "Joint estimation of angle and Doppler frequency in MIMO radar," in *Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on*, IEEE, pp. 231–235, 2013.
- [73] X. Luciani and L. Albera, "Joint eigenvalue decomposition using polar matrix factorization," in *Latent Variable Analysis and Signal Separation*. Springer, 2010, pp. 555–562.
- [74] X. Luciani and L. Albera, "Canonical polyadic decomposition based on joint eigenvalue decomposition," in *Chemometrics and Intelligent Laboratory Systems*, Vol. 132, pp. 152–167, 2014.
- [75] X. Luciani and L. Albera, "Joint eigenvalue decomposition of non-defective matrices based on the LU factorization with application to ICA," in *Signal Processing, IEEE Transactions on*, Vol. 63, No. 17, pp. 4594–4608, 2015.
- [76] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," in *Journal of the Society for Industrial & Applied Mathematics*, Vol. 11, No. 2, pp. 431–441, 1963.
- [77] K. A. Meraim, A. Belouchrani, J.-F. Cardoso, and E. Moulines, "Asymptotic performance of second order blind source separation," in *Proc. ICASSP*, Vol. 4, pp. 277–280, 1994.
- [78] A. Mesloub, K. Abed-Meraim, and A. Belouchrani, "A new algorithm for complex non-orthogonal joint diagonalization based on Shear and Givens rotations," in *Signal Processing, IEEE Transactions on*, Vol. 62, No. 8, pp. 1913–1925, 2014.

- [79] F. Miwakeichi, E. Martinez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, "Decomposing EEG data into space–time–frequency components using parallel factor analysis," *in* *NeuroImage*, Vol. 22, No. 3, pp. 1035–1045, 2004.
- [80] E. Moreau, "A generalization of joint-diagonalization criteria for source separation," *in* *Signal Processing*, IEEE Transactions on, Vol. 49, No. 3, pp. 530–541, 2001.
- [81] E. Moreau, "Comments on" CuBICA : Independent Component Analysis by Simultaneous third-and Fourth-Order Cumulant Diagonalization," *in* *Signal Processing*, IEEE Transactions on, Vol. 54, No. 12, pp. 4826–4828, 2006.
- [82] E. Moreau, "Joint diagonalization of third order complex symmetric tensors and application to blind separation of non-circular sources," *in* *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*, IEEE, pp. 417–421, 2007.
- [83] E. Moreau and T. Adali, *Blind Identification and Separation of Complex-valued Signals*. Wiley-ISTE, 2013.
- [84] E. Moreau and O. Macchi, "High-order contrasts for self-adaptive source separation," *in* *International Journal of Adaptive Control and Signal Processing*, Vol. 10, No. 1, pp. 19–46, 1996.
- [85] E. Moreau and O. Macchi, "A one stage self-adaptive algorithm for source separation," *in* *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, Vol. 3, IEEE, pp. III–49, 1994.
- [86] M. Mørup, "Applications of tensor (multiway array) factorizations and decompositions in data mining," *in* *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, Vol. 1, No. 1, pp. 24–40, 2011.
- [87] S. Moussaoui, D. Brie, C. Carteret, and A. Mohammad-Djafari, "Application of Bayesian non-negative source separation to mixture analysis in spectroscopy," *in* *24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, MAXENT 2004*, pp. 237–244, 2004.
- [88] D. Muti and S. Bourennane, "Multiway filtering based on fourth-order cumulants," *in* *EURASIP Journal on Advances in Signal Processing*, Vol. 2005, No. 7, p. 708571, 2005.
- [89] C. L. Nikias and A. P. Petropulu, "Higher-Order Spectra Analysis : A Nonlinear Signal Processing Framework," *in* PTR Prentice Hall, Englewood Cliffs, NJ, 1993.
- [90] D. Nion, "A tensor framework for nonunitary joint block diagonalization," *in* *Signal Processing*, IEEE Transactions on, Vol. 59, No. 10, pp. 4585–4594, 2011.

- [91] E. Ollila and V. Koivunen, “Complex ICA using generalized uncorrelating transform,” *in* *Signal Processing*, Vol. 89, No. 4, pp. 365–377, 2009.
- [92] C. Paulus, J. Mars *et al.*, “New multicomponent filters for geophysical data processing,” *in* *Geoscience and Remote Sensing, IEEE Transactions on*, Vol. 44, No. 8, pp. 2260–2270, 2006.
- [93] D. T. Pham, “Joint approximate diagonalization of positive definite Hermitian matrices,” *in* *SIAM Journal on Matrix Analysis and Applications*, Vol. 22, No. 4, pp. 1136–1152, 2001.
- [94] D. T. Pham and P. Garat, “Blind separation of mixture of independent sources through a maximum likelihood approach,” *in* *In Proc. EUSIPCO*, Citeseer, 1997.
- [95] A.-H. Phan, A. Cichocki, and P. Tichavsky, “On fast algorithms for orthogonal Tucker decomposition,” *in* *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, IEEE, pp. 6766–6770, 2014.
- [96] A.-H. Phan, P. Tichavsky, and A. Cichocki, “Candecomp/Parafac decomposition of high-order tensors through tensor reshaping,” *in* *Signal Processing, IEEE Transactions on*, Vol. 61, No. 19, pp. 4847–4860, 2013.
- [97] B. Picinbono, “On circularity,” *in* *Signal Processing, IEEE Transactions on*, Vol. 42, No. 12, pp. 3473–3482, 1994.
- [98] M. Rajih, P. Comon, and R. A. Harshman, “Enhanced line search : A novel method to accelerate PARAFAC,” *in* *SIAM Journal on Matrix Analysis and Applications*, Vol. 30, No. 3, pp. 1128–1147, 2008.
- [99] J.-P. Royer, N. Thirion-Moreau, P. Comon, R. Redon, and S. Mounier, “A regularized nonnegative canonical polyadic decomposition algorithm with preprocessing for 3D fluorescence spectroscopy,” *in* *Journal of Chemometrics*, Vol. 29, No. 4, pp. 253–265, 2015.
- [100] M. Seaton, “Diagonalisation of complex symmetric matrices using a modified Jacobi method,” *in* *The Computer Journal*, Vol. 12, No. 2, pp. 156–157, 1969.
- [101] H. Shen and M. Kleinstueber, “Complex blind source separation via simultaneous strong uncorrelating transform,” *in* *Latent Variable Analysis and Signal Separation*. Springer, 2010, pp. 287–294.
- [102] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, “Blind PARAFAC receivers for DS-CDMA systems,” *in* *Signal Processing, IEEE Transactions on*, Vol. 48, No. 3, pp. 810–823, 2000.

- [103] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis : applications in the chemical sciences*. John Wiley & Sons, 2005.
- [104] L. Sorber, M. V. Barel, and L. D. Lathauwer, “Unconstrained optimization of real functions in complex variables,” in *SIAM Journal on Optimization*, Vol. 22, No. 3, pp. 879–898, 2012.
- [105] L. Sorber, I. Domanov, M. Van Barel, and L. De Lathauwer, “Exact line and plane search for tensor optimization,” in *Computational Optimization and Applications*, pp. 1–22, 2014.
- [106] L. Sorber, M. Van Barel, and L. De Lathauwer, “Optimization-Based Algorithms for Tensor Decompositions : Canonical Polyadic Decomposition, Decomposition in Rank- $(L_r, L_r, 1)$ Terms, and a New Generalization,” in *SIAM Journal on Optimization*, Vol. 23, No. 2, pp. 695–720, 2013.
- [107] L. Sorber, M. Van Barel, and L. De Lathauwer, “Tensorlab v2. 0,” in Available online, URL : www.tensorlab.net, 2014.
- [108] M. Sørensen, P. Comon, S. Icart, and L. Deneire, “Approximate tensor diagonalization by invertible transforms,” in *Signal Processing Conference, 2009 17th European*, IEEE, pp. 500–504, 2009.
- [109] A. Souloumiac, “Nonorthogonal joint diagonalization by combining givens and hyperbolic rotations,” in *Signal Processing, IEEE Transactions on*, Vol. 57, No. 6, pp. 2222–2231, 2009.
- [110] N. Thirion, “Separation d’ondes en prospection sismique,” Ph.D. dissertation, 1995.
- [111] N. Thirion, J. Mars, and J.-L. Lacoume, “Analytical links between steering vectors and eigenvectors,” in *Proceedings of the European Signal Processing Conference (EUSIPCO’96)*, pp. 81–84, 1996.
- [112] P. Tichavsky and Z. Koldovsky, “Algorithms for nonorthogonal approximate joint block-diagonalization,” in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, IEEE, pp. 2094–2098, 2012.
- [113] P. Tichavský and Z. Koldovský, “Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources,” in *Signal Processing, IEEE Transactions on*, Vol. 59, No. 3, pp. 1037–1047, 2011.
- [114] P. Tichavský and A. Yeredor, “Fast approximate joint diagonalization incorporating weight matrices,” in *Signal Processing, IEEE Transactions on*, Vol. 57, No. 3, pp. 878–891, 2009.

- [115] K. Todros and J. Tabrikian, "QML-based joint diagonalization of positive-definite hermitian matrices," *in* Signal Processing, IEEE Transactions on, Vol. 58, No. 9, pp. 4656–4673, 2010.
- [116] L. Tong, R.-w. Liu, V. C. Soon, and Y.-F. Huang, "Indeterminacy and identifiability of blind identification," *in* Circuits and Systems, IEEE Transactions on, Vol. 38, No. 5, pp. 499–509, 1991.
- [117] T. Trainini, X.-L. Li, E. Moreau, and T. Adali, "A relative gradient algorithm for joint decompositions of complex matrices," *in* Proc. Eur. Signal Process. Conf.(EUSIPCO), Vol. 1, 2010.
- [118] T. Trainini and E. Moreau, "A Coordinate Descent Algorithm for Complex Joint Diagonalization Under Hermitian and Transpose Congruences," *in* Signal Processing, IEEE Transactions on, Vol. 62, No. 19, pp. 4974–4983, 2014.
- [119] E. Trefftz, "Overlapping metastable levels in dielectronic recombination," *in* Zeitschrift fur Astrophysik, Vol. 65, p. 299, 1967.
- [120] L. R. Tucker, "The extension of factor analysis to three-dimensional matrices," *in* Contributions to mathematical psychology, pp. 109–127, 1964.
- [121] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *in* Psychometrika, Vol. 31, No. 3, pp. 279–311, 1966.
- [122] A.-J. Van der Veen, "Joint diagonalization via subspace fitting techniques," *in* Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on, Vol. 5, IEEE, pp. 2773–2776, 2001.
- [123] E. Vincent and Y. Deville, "Audio applications," *in* Handbook of Blind Source Separation, Independent Component Analysis and Applications, pp. 779–819, 2010.
- [124] R. Vollgraf and K. Obermayer, "Quadratic optimization for simultaneous matrix diagonalization," *in* Signal Processing, IEEE Transactions on, Vol. 54, No. 9, pp. 3270–3278, 2006.
- [125] V. Vrabie, J. I. Mars, and J.-L. Lacoume, "Modified singular value decomposition by means of independent component analysis," *in* Signal Processing, Vol. 84, No. 3, pp. 645–652, 2004.
- [126] F. Wang, Z. Liu, and J. Zhang, "Nonorthogonal joint diagonalization algorithm based on trigonometric parameterization," *in* Signal Processing, IEEE Transactions on, Vol. 55, No. 11, pp. 5299–5308, 2007.
- [127] K. Wang, X.-F. Gong, and Q.-H. Lin, "Complex non-orthogonal joint diagonalization based on LU and LQ decompositions," *in* Latent Variable Analysis and Signal Separation. Springer, 2012, pp. 50–57.

- [128] W. Wang, S. Sanei, J. Chambers *et al.*, “Penalty function-based joint diagonalization approach for convolutive blind separation of nonstationary sources,” *in* Signal Processing, IEEE Transactions on, Vol. 53, No. 5, pp. 1654–1669, 2005.
- [129] M. Wax and J. Sheinvald, “A least-squares approach to joint diagonalization,” *in* Signal Processing Letters, IEEE, Vol. 4, No. 2, pp. 52–53, 1997.
- [130] G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar, “Tensor displays : compressive light field synthesis using multilayer displays with directional backlighting.” *in* ACM Trans. Graph., Vol. 31, No. 4, p. 80, 2012.
- [131] X.-F. Xu, D.-Z. Feng, and W. X. Zheng, “A fast algorithm for nonunitary joint diagonalization and its application to blind source separation,” *in* Signal Processing, IEEE Transactions on, Vol. 59, No. 7, pp. 3457–3463, 2011.
- [132] A. Yeredor, “Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation,” *in* Signal Processing, IEEE Transactions on, Vol. 50, No. 7, pp. 1545–1553, 2002.
- [133] A. Yeredor, “On using exact joint diagonalization for noniterative approximate joint diagonalization,” *in* Signal Processing Letters, IEEE, Vol. 12, No. 9, pp. 645–648, 2005.
- [134] T. J. Zeng and Q. Y. Feng, “Non-orthogonal joint diagonalization algorithm based on hybrid trust region method and its application to blind source separation,” *in* Neurocomputing, Vol. 133, pp. 280–294, 2014.
- [135] 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,” *in* The Journal of Machine Learning Research, Vol. 5, pp. 777–800, 2004.

Annexe 1 : Article

A Decoupled Jacobi-Like Algorithm for Non-Unitary Joint Diagonalization of Complex-Valued Matrices

Victor Maurandi and Eric Moreau, *Senior Member, IEEE*

Abstract—We consider the problem of non-orthogonal joint diagonalization of a set of complex matrices. This appears in many signal processing problems and is instrumental in source separation. We propose a new Jacobi-like algorithm based both on a special parameterization of the diagonalizing matrix and on an adapted local criterion. The optimization scheme is based on an alternate estimation of the useful parameters. Numerical simulations illustrate the overall very good performances of the proposed algorithm in comparison to two other Jacobi-like algorithms and to a global algorithm existing in the literature.

Index Terms—Blind source separation, independent component analysis, Jacobi algorithm, joint diagonalization.

I. INTRODUCTION

JOINT DIAGONALIZATION of sets of matrices is an important issue in blind signal processing and more particularly in source separation, see e.g. the overview paper [1] and more specifically [2]–[20]. The aim is to estimate a matrix which jointly diagonalizes the given matrix set in using a particular decomposition that depends on the considered matrix properties. For this, different approaches have been proposed in the literature. Classically, they are based on the direct estimation of the diagonalizing matrix [2]–[4]. Recently, some algorithms used a Jacobi iterative scheme to estimate the diagonalizing matrix as e.g. [5]–[8]. Jacobi procedures have the main advantage to be simple to implement, especially when one is able to derive an analytical solution in the basic two dimension case. Moreover these methods often allow a potential computational parallelism.

The unitary case for Hermitian matrices has been first considered yielding the popular JADE [9] algorithm using fourth-order cumulants. This approach was generalized to statistics of any order greater than two in [10]. The other interesting unitary complex case, for symmetric matrices, was considered in [11]. Nowadays the non-unitary diagonalizing matrix case has become important mainly because it allows to skip a first processing step (whitening of the observations in source separation) which limits the performances in practice. Recently, non-unitary Jacobi-like algorithms have been proposed for the real case in [5], [6] and for the complex Hermitian case in [7], [8].

Manuscript received June 11, 2014; accepted July 07, 2014. Date of publication July 17, 2014; date of current version July 29, 2014. This work was supported by the French Direction Générale de l'Armement (DGA). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Paris Smaragdakis.

The authors are with the Aix Marseille Université, CNRS, ENSAM, LSIS, UMR 7296, 13397 Marseille, France, and also with the Université de Toulon, CNRS, LSIS, UMR 7296, 83957 La Garde, France (e-mail: moreau@univ-tln.fr).

Digital Object Identifier 10.1109/LSP.2014.2339891

In this paper, we propose a new Jacobi-like algorithm for non-unitary joint diagonalization of complex-valued matrices. We both consider the Hermitian and the symmetric cases within the same framework. The proposed algorithm follows the work proposed in [5] in the real case. However, we consider a novel approach based on a well-chosen adapted criterion. The analytical optimal solutions are derived. In using computer simulations, we illustrate the performances of the proposed algorithm in comparison to the ones suggested in [5] (generalized to the two complex cases), [4] and [8]. The proposed algorithm shows an improved convergence speed and an improved performance with respect to its competitors. Finally, we mention that part of this work concerning the only complex symmetric case has been briefly presented in [12].

II. PROBLEM FORMULATION

We consider K ($K \geq 2$) complex matrices \mathbf{M}_k , $k = 1, \dots, K$, defined as

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

where $(\cdot)^\dagger$ corresponds to the transpose matrix operator, denoted specifically as $(\cdot)^T$, or to the conjugate transpose matrix operator, denoted specifically as $(\cdot)^H$. We consider the square case where all matrices are of size $N \times N$. In (1), the matrices \mathbf{D}_k all are complex diagonal and the matrix \mathbf{A} is complex invertible. In source separation, the matrix \mathbf{A} is the so-called mixing matrix [13]. When $\dagger \equiv H$ (resp. $\dagger \equiv T$) this is referred to as the Hermitian (resp. the complex symmetric) case.

In using only the matrices \mathbf{M}_k , the aim of joint diagonalization consists of estimating a diagonalizing matrix \mathbf{B} such that all the transformed matrices

$$\mathbf{M}'_k = \mathbf{B} \mathbf{M}_k \mathbf{B}^\dagger \quad (2)$$

are jointly diagonal. Hence, ideally, \mathbf{B} has to be equal to the inverse of the mixing matrix \mathbf{A} up to the left product by an invertible diagonal matrix and by a permutation matrix. In practice, since matrices \mathbf{M}_k are “only” estimated from statistics or other analysis operators, it is impossible to make all matrices \mathbf{M}'_k exactly diagonal but only in an approximate sense. The approximation is measured in using a criterion. One of the most useful and used criterion, see e.g. [1], is

$$J(\mathbf{B}) = \sum_{k=1}^K \|\text{Zdiag}\{\mathbf{M}'_k\}\|^2 \quad (3)$$

where $\|\cdot\|$ is the matrix Frobenius norm and $\text{Zdiag}\{\mathbf{X}\}$ is the matrix defined as the matrix \mathbf{X} with all zeros on its main diagonal. The joint diagonalization problem now consists of searching for an invertible matrix \mathbf{B} that minimizes J . Since $\mathbf{B} = \mathbf{0}$ is a trivial (non interesting) matrix minimizing J , it is clearly important to fix this problem. A norm normalization

could be done but it does not guaranty an invertible matrix. It is the reason why we choose to constrain \mathbf{B} to have a unit determinant, *i.e.* $\det \mathbf{B} = 1$. Even if, in the following, J is not directly concerned, all the discussion still holds.

III. THE PROPOSED JACOBI PROCEDURE

The basic Jacobi matrix estimation procedure consists of decomposing the overall $N \times N$ matrix as the product of $\frac{N(N-1)}{2}$ elementary matrices \mathbf{B}^{ij} as

$$\mathbf{B} = \prod_{i=1}^{N-1} \prod_{j=i+1}^N \mathbf{B}^{ij}. \quad (4)$$

Each elementary matrix \mathbf{B}^{ij} corresponds to the $N \times N$ identity matrix where the four components at the intersection of the i -th and j -th rows and columns are characterized by a 2×2 sub-matrix denoted as

$$\tilde{\mathbf{B}}^{ij} = \begin{pmatrix} B_{ii}^{ij} & B_{ij}^{ij} \\ B_{ji}^{ij} & B_{jj}^{ij} \end{pmatrix}. \quad (5)$$

Since we are looking for a matrix \mathbf{B} only up to the left product by a diagonal matrix, two of the four parameters of $\tilde{\mathbf{B}}^{ij}$ are of no interest and thus only two are of interest. On the other hand, the unit determinant constraint is directly reported as $\det \tilde{\mathbf{B}}^{ij} = 1$ for all i, j . Triangular matrices are very useful since these are always of unit determinant when their diagonal components all are equal to 1. In that case, there are only two possibilities for $\tilde{\mathbf{B}}^{ij}$. Either

$$\tilde{\mathbf{B}}^{ij} = \tilde{\mathbf{L}}^{ij} = \begin{pmatrix} 1 & 0 \\ \ell_{ij} & 1 \end{pmatrix} \quad (6)$$

or

$$\tilde{\mathbf{B}}^{ij} = \tilde{\mathbf{U}}^{ij} = \begin{pmatrix} 1 & u_{ij} \\ 0 & 1 \end{pmatrix}. \quad (7)$$

These two possibilities corresponding to the two searched parameters are used one after the other in a recursive scheme. More precisely, the proposed Jacobi procedure is described as

- For each indexes $1 \leq i \leq j \leq N$
 - 1) $\tilde{\mathbf{B}}^{ij} = \tilde{\mathbf{U}}^{ij}$ is calculated and $\mathbf{U}^{ij} = \mathbf{B}^{ij}$ is built
 - 2) the matrix set is updated as $\mathbf{M}_k \leftarrow \mathbf{U}^{ij} \mathbf{M}_k \mathbf{U}^{ij\dagger}$
- For each indexes $1 \leq i \leq j \leq N$
 - 1) $\tilde{\mathbf{B}}^{ij} = \tilde{\mathbf{L}}^{ij}$ is calculated and $\mathbf{L}^{ij} = \mathbf{B}^{ij}$ is built
 - 2) the matrix set is updated as $\mathbf{M}_k \leftarrow \mathbf{L}^{ij} \mathbf{M}_k \mathbf{L}^{ij\dagger}$

The above procedure corresponds to a so-called sweep. It has to be iterated until convergence.

We propose to determine the optimal parameters u_{ij} (resp. ℓ_{ij}) through the use of the 2×2 matrix $\tilde{\mathbf{U}}^{ij}$ (resp. $\tilde{\mathbf{L}}^{ij}$) and not from the $N \times N$ matrix \mathbf{U}^{ij} (resp. \mathbf{L}^{ij}). It is an important difference with the approach in [5] that is not only just a question of implementation (otherwise the obtained performances would be nearly the same). This is developed in the next section.

IV. CONSIDERED CRITERION

In [5], the optimal parameters are derived by considering independently all matrices \mathbf{U}^{ij} and \mathbf{L}^{ij} in place of \mathbf{B} in criterion. Hence all off-diagonal components of the target matrices \mathbf{M}_k for the rows and columns i and j are considered altogether in order to globally minimize them (within the iterations) when building \mathbf{M}'_k . More importantly the same off-diagonal components are considered several times whenever i and j vary. Moreover this directly depends on the size N of the matrices. It is

clearly not a good strategy leading to a loss of performance for each iteration.

Here, for given i and j , we propose a very different strategy by focusing on the only two off-diagonal components at the (i, j) and (j, i) locations. Hence, we consider the extracted 2×2 sub-matrix $\tilde{\mathbf{M}}_k^{ij}$ defined as

$$\tilde{\mathbf{M}}_k^{ij} = \begin{pmatrix} M_{k,ii} & M_{k,ij} \\ M_{k,ji} & M_{k,jj} \end{pmatrix} \quad (8)$$

and the associated criterion

$$J_2(\tilde{\mathbf{B}}^{ij}) = \sum_{k=1}^K \|\text{Zdiag}\{\tilde{\mathbf{B}}^{ij} \tilde{\mathbf{M}}_k^{ij} \tilde{\mathbf{B}}^{ij\dagger}\}\|^2 \quad (9)$$

which thus directly focus on two off-diagonal components.

The two criteria J and J_2 are different. To clearly emphasize this point, we now derive the link between them. We only consider the Hermitian case when $\tilde{\mathbf{B}}^{ij} = \tilde{\mathbf{U}}^{ij}$. The other cases are similar. Straightforward algebraic derivations lead to

$$J(\mathbf{U}^{ij}) = J_2(\tilde{\mathbf{U}}^{ij}) + J'(\mathbf{U}^{ij}) + C \quad (10)$$

where

$$\begin{aligned} J'(\mathbf{U}^{ij}) &= \sum_{k=1}^K \sum_{q \in \mathcal{Q}} [|M_{k,iq} + M_{k,qj} u_{ij}|^2 \\ &\quad + |M_{k,qi} + M_{k,qj} u_{ij}^*|^2] \\ C &= \sum_{k=1}^K \sum_{\substack{(m,n) \in \mathcal{P}^2 \\ m \neq n}} |M_{k,mn}|^2 \end{aligned}$$

with $|\cdot|$ is the complex modulus, $(\cdot)^*$ is the complex conjugate operator, $\mathcal{P} = \{1, \dots, N\} \setminus \{i\}$ and $\mathcal{Q} = \mathcal{P} \setminus \{j\}$. Because J' in (10) is not constant, the two criteria J and J_2 are not equivalent. However, we have the following very interesting property. When matrices \mathbf{M}_k (that evolve with the iterations) are close diagonal (in the sense that the off-diagonal components of each \mathbf{M}_k have a very small magnitude in comparison to the diagonal component with the smallest magnitude), then it is rather easy to show that $|J'(\mathbf{U}^{ij})| \ll |J_2(\tilde{\mathbf{U}}^{ij})|$ leading to

$$J(\mathbf{U}^{ij}) \approx J_2(\tilde{\mathbf{U}}^{ij}) + C. \quad (11)$$

Thus, both criteria are almost equivalent close to a diagonalizing solution. We can conclude that the criteria (3) and (9) are different, nevertheless when we are close to a diagonalizing solution they are almost equivalent.

V. OPTIMAL SOLUTIONS

We now derive the optimal solutions for u_{ij} and ℓ_{ij} by minimizing $J_2(\tilde{\mathbf{B}}^{ij})$. For a given (i, j) , we first consider the case $\tilde{\mathbf{B}}^{ij} = \tilde{\mathbf{U}}^{ij}$. In the complex symmetric case, J_2 denoted by $J_2^{(T)}$, writes as

$$J_2^{(T)}(\tilde{\mathbf{U}}^{ij}) = 2 \sum_{k=1}^K |M_{k,ij} + M_{k,jj} u_{ij}|^2 \quad (12)$$

and its derivative *w.r.t.* u_{ij}^* reads

$$\frac{\partial J_2^{(T)}(\tilde{\mathbf{U}}^{ij})}{\partial u_{ij}^*} = 2 \sum_{k=1}^K M_{k,ij} M_{k,jj}^* + |M_{k,jj}|^2 u_{ij}. \quad (13)$$

TABLE I
 ALUJA ALGORITHM

Initialize \mathbf{B} , e.g. the identity matrix \mathbf{I} of size $N \times N$. Repeat N_{ite} times the following process
1) Upper triangular part, for $1 \leq i < j \leq N$ <ul style="list-style-type: none"> • Extract matrices $\widetilde{\mathbf{M}}_k^{ij}$ • Compute u_{ij} by using (14) or (17) • Build \mathbf{U}^{ij} • $\mathbf{M}_k \leftarrow \mathbf{U}^{ij} \mathbf{M}_k \mathbf{U}^{ij \dagger}$ and $\mathbf{B} \leftarrow \mathbf{U}^{ij} \mathbf{B}$
2) Lower triangular part, for $1 \leq i < j \leq N$ <ul style="list-style-type: none"> • Extract matrices $\widetilde{\mathbf{M}}_k^{ij}$ • Compute ℓ_{ij} by using (14) or (17) permuting i and j • Build \mathbf{L}^{ij} • $\mathbf{M}_k \leftarrow \mathbf{L}^{ij} \mathbf{M}_k \mathbf{L}^{ij \dagger}$ and $\mathbf{B} \leftarrow \mathbf{L}^{ij} \mathbf{B}$

By setting it to zero, the optimal value of u_{ij} follows as

$$u_{ij} = -\frac{\sum_{k=1}^K M_{k,ij} M_{k,jj}^*}{\sum_{k=1}^K |M_{k,jj}|^2}. \quad (14)$$

In the Hermitian case, J_2 denoted by $J_2^{(H)}$, writes as

$$J_2^{(H)}(\widetilde{\mathbf{U}}^{ij}) = \sum_{k=1}^K |M_{k,ij} + M_{k,jj} u_{ij}|^2 + |M_{k,ji} + M_{k,jj} u_{ij}^*|^2 \quad (15)$$

and its derivative *w.r.t.* u_{ij}^* reads

$$\frac{\partial J_2^{(H)}(\widetilde{\mathbf{U}}^{ij})}{\partial u_{ij}^*} = \sum_{k=1}^K M_{k,ij} M_{k,jj}^* + M_{k,ji}^* M_{k,jj} + 2|M_{k,jj}|^2 u_{ij}. \quad (16)$$

By setting it to zero, the optimal value of u_{ij} follows as

$$u_{ij} = -\frac{\sum_{k=1}^K M_{k,ij} M_{k,jj}^* + M_{k,ji}^* M_{k,jj}}{2 \sum_{k=1}^K |M_{k,jj}|^2}. \quad (17)$$

It is interesting to notice that if \mathbf{D}_k in (1) are real in the Hermitian case, then matrices \mathbf{M}_k are Hermitian and it is then straightforward to see that both expressions (14) and (17) are identical.

We now consider $\widetilde{\mathbf{B}}^{ij} = \widetilde{\mathbf{L}}^{ij}$. To simplify the exposition, it is clear that the transpose of such a lower triangular matrix is an upper triangular matrix and we come back to the above case. Thus the two optimal values of ℓ_{ij} directly follows from (14) and (17) by permuting i and j respectively in each right part.

An outline of the complete proposed algorithm is presented in Table I. We call it ‘Alternate LU Jacobi Algorithm’ (ALUJA).

VI. SIMULATIONS

We now illustrate the performances of the proposed ALUJA algorithm in comparison to two other Jacobi-like algorithms namely the JTJD algorithm suggested in [8] (Hermitian case) and the LUJID algorithm suggested in [5] in the real case that we generalized to the complex case (we call it C-LUJID). We also consider the U-WEDGE algorithm suggested in [4] in the real case but directly generalized to the Hermitian one.

In order to evaluate the algorithm performances, we use the performance index suggested in [10], [14], [15]. It is defined by $I(\mathbf{S}) = \frac{1}{2N(N-1)} I'(\mathbf{S})$ where

$$I'(\mathbf{S}) = \sum_{i=1}^N \left(\sum_{j=1}^N \frac{|S_{ij}|^2}{S_r^2} - 1 \right) + \sum_{j=1}^N \left(\sum_{i=1}^N \frac{|S_{ij}|^2}{S_c^2} - 1 \right) \quad (18)$$

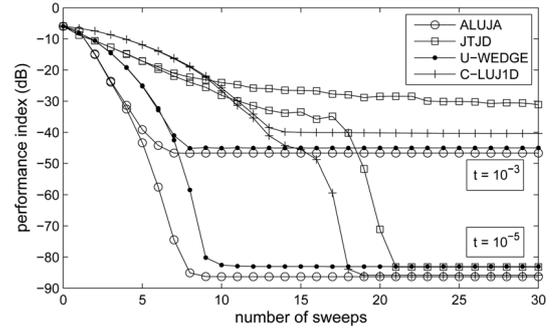


Fig. 1. Mean level of performance index versus the number of iterations in the Hermitian case (100 independent draws of 25 matrices of size 25×25 with $t = 10^{-3}$ and $t = 10^{-5}$).

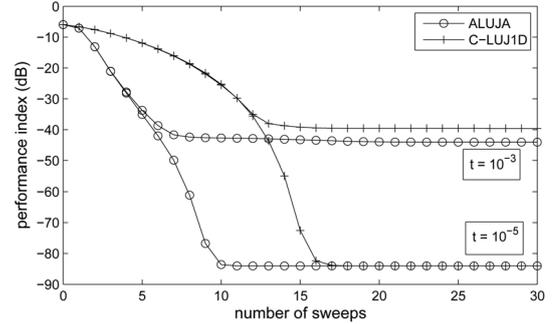


Fig. 2. Mean level of performance index versus the number of iterations in the complex symmetric case (100 independent draws of 25 matrices of size 25×25 with $t = 10^{-3}$ and $t = 10^{-5}$).

with $\mathbf{S} = \mathbf{B}\mathbf{A}$, $S_r^2 = \max_m |S_{im}|^2$ and $S_c^2 = \max_m |S_{mj}|^2$. This index is always positive and is equal to zero when $\mathbf{S} = \mathbf{D}\mathbf{P}$ where \mathbf{D} and \mathbf{P} are respectively a diagonal matrix and a permutation matrix corresponding to the classical acceptable indeterminacies.

We consider 25 noisy complex matrices of size 25×25 defined as $\mathbf{M}_k + t\mathbf{N}_k$ where the noise matrices \mathbf{N}_k follow the same characterizing properties of \mathbf{M}_k . We consider two noise levels, $t = 10^{-5}$ and $t = 10^{-3}$. The diagonal matrices \mathbf{D}_k , the mixing matrix \mathbf{A} and the noise matrices \mathbf{N}_k are all drawn following a zero mean unit variance normal distribution for both the real and imaginary parts. The matrices \mathbf{N}_k are then forced by projection to follow the same symmetry properties as \mathbf{M}_k . The diagonalizing matrix \mathbf{B} is initialized as the identity matrix. Finally, we display the mean value of the performance index over one hundred independent draws.

Fig. 1 displays the mean index value *w.r.t.* iterations (the sweeps for the Jacobi-like algorithms) in the Hermitian case and for both values of t . It exhibits that the proposed ALUJA algorithm outperforms the three other ones for both the convergence speed and more surprisingly for the attained level of performance after convergence.

Fig. 2 displays the mean index value *w.r.t.* iterations (the sweeps for the Jacobi-like algorithms) in the complex symmetric case and with both values of t . Because U-WEDGE and JTJD algorithms do not work in this case, we only compare with C-LUJID. As previously, ALUJA algorithm outperforms C-LUJID, presenting a similar behavior to the one in the above Hermitian case. Let recall that the convergence speed of ALUJA is much better than the C-LUJID one.

In Table II, we give the mean, the standard deviation, the median and the interquartile range for all algorithms after convergence with the new value $t = 10^{-2}$. For other values of t the

TABLE II
SOME STATISTICS AFTER CONVERGENCE (100 INDEPENDENT DRAWS OF
25 MATRICES OF SIZE 25×25 WITH $t=10^{-2}$)

Hermitian	mean	std	median	iqr
ALUJA	1.058e-3	2.122e-3	1.854e-4	5.596e-4
C-LUJID	1.273e-3	1.771e-3	7.847e-4	7.372e-4
U-WEDGE	1.378e-3	2.980e-3	3.208e-4	8.432e-4
JTJD	1.372e-3	2.552e-3	3.088e-4	7.733e-4
Symmetric	mean	std	median	iqr
ALUJA	1.399e-3	1.895e-3	4.510e-4	1.142e-3
C-LUJID	1.474e-3	1.445e-3	1.066e-3	1.106e-3

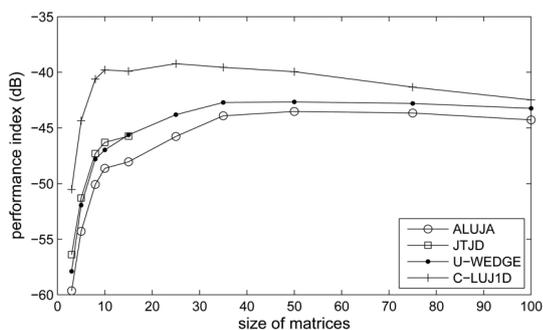


Fig. 3. Mean level of performance index obtained after convergence versus the size of matrices in the Hermitian case (100 independent draws of 25 matrices with $t=10^{-3}$ for each matrix size).

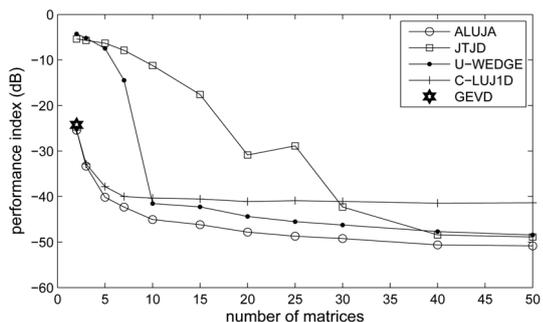


Fig. 4. Mean level of performance index obtained after convergence versus the number of considered matrices in the Hermitian case (100 independent draws of 25×25 matrices with $t=10^{-3}$ for each number of matrices).

results are similar. In all cases, ALUJA has the best mean and median. C-LUJID has the best standard deviation and the best dispersion in the symmetric case.

Now let show the influence of the number of matrices K and of the size of matrices N on the algorithms performances (after convergence). For that we fix the non varying parameter to the value given above. We only display the curves in the Hermitian case for $t=10^{-3}$ but the results are similar for $t=10^{-5}$ (of course with better performances). That is the same for the symmetric case.

Fig. 3 displays the level of performance after convergence w.r.t. the size N . It results that ALUJA algorithm is more accurate than the other ones. Notice that in this simulation, after $N=15$, JTJD diverges for at least one draw. That is why the corresponding curve is only defined for $N \leq 15$.

Finally Fig. 4 displays the level of performance after convergence w.r.t. the number K of considered matrices. It shows that the proposed ALUJA algorithm is again more accurate than the other ones especially when the number of matrices is small. For a given matrix size, JTJD needs a number of matrices to be efficient. To a lesser extent, it is also the case for U-WEDGE. We also note that for $K=2$, a true diagonalization (here GEVD) provides a result similar to ALUJA namely $-24dB$.

VII. CONCLUSION

For non-unitary joint diagonalization of complex-valued matrices, we proposed a new decoupled Jacobi-like algorithm. It is based on the optimization of a local criterion that focuses on two symmetric off-diagonal matrix components. A link is shown with the classical criterion and the optimal parameters are derived analytically. The computer simulations illustrate the overall very interesting behavior of the proposed ALUJA algorithm and show that it compares favorably to U-WEDGE, JTJD and C-LUJID algorithms in terms of convergence speed and achieved performance level.

REFERENCES

- [1] G. Chabriel, M. Kleinstuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor, "Joint Matrices Decompositions and Blind Source Separation," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 34–43, May 2014.
- [2] 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, Jul. 2002.
- [3] X.-L. Li and X.-D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solution," *IEEE Trans. Signal Process.*, vol. 55, no. 5, May 2007.
- [4] P. Tichavsky and A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 878–891, Mar. 2009.
- [5] B. Afsari, "Simple LU and QR based non-orthogonal matrix joint diagonalization," in *Proc. ICA'2006, Springer LNCS series*, 2006, pp. 1–7.
- [6] A. Souloumiac, "Nonorthogonal joint diagonalization by combining Givens and hyperbolic rotations," *IEEE Trans. Signal Processing*, vol. 57, no. 6, pp. 2222–2231, Jun. 2009.
- [7] V. Maurandi, E. Moreau, and C. De Luigi, "Jacobi like algorithms for non-orthogonal joint diagonalization of hermitian matrices," in *Proc. ICASSP'2014*, Florence, Italy, May 2014.
- [8] X. Guo, S. Zhu, S. Miron, and D. Brie, "Approximate joint diagonalization by nonorthogonal nonparametric Jacobi transformations," in *Proc. ICASSP'2010*, Dallas, USA, Mar. 2010.
- [9] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non Gaussian signals," *Proc. Inst. Elect. Eng. F*, vol. 40, pp. 362–370, 1993.
- [10] E. Moreau, "A generalization of joint-diagonalization criteria for source separation," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 530–541, Mar. 2001.
- [11] L. De Lathauwer, B. De Moor, and J. Vandewalle, "ICA techniques for more sources than sensors," in *Proc. IEEE Workshop on Higher-Order Statistics (HOS'99)*, Caesarea, Israel, Jun. 1999, pp. 121–124.
- [12] V. Maurandi, C. De Luigi, and E. Moreau, "Fast Jacobi like algorithms for joint diagonalization of complex symmetric matrices," in *Proc. EU-SIPCO'2013*, Marrakech, Morocco, Sep. 2013.
- [13] E. Moreau and T. Adali, "Blind Identification and Separation of Complex-valued Signals," in *FOCUS Digital Signal and Image Processing Series* Wiley-ISTE, 2013.
- [14] E. Moreau and O. Macchi, "A one stage self-adaptive algorithm for source separation," in *Proc. ICASSP*, Apr. 1994, vol. 3, pp. 49–52.
- [15] E. Moreau and O. Macchi, "High order contrasts for self-adaptive source separation," *International J. Adaptive Control and Signal Processing*, vol. 10, no. 1, pp. 19–46, Jan. 1996.
- [16] 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. 777–863, Jul. 2004.
- [17] D. T. Pham, "Joint approximate diagonalization of positive definite hermitian matrices," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 4, pp. 1136–1152, 2001.
- [18] F. Wang, Z. Liu, and J. Zhang, "Nonorthogonal joint diagonalization algorithm based on trigonometric parameterization," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5299–5308, Nov. 2007.
- [19] 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. Lett.*, vol. 14, no. 11, pp. 860–863, Nov. 2007.
- [20] T. Trainini and E. Moreau, "Variations around gradient like algorithms for joint diagonalization of Hermitian matrices," in *Proc. EUSIPCO'2012*, Bucharest, Romania, Aug. 2012.

Résumé

Cette thèse développe des méthodes de diagonalisation conjointe de matrices et de tenseurs d'ordre trois, et son application à la séparation MIMO de sources de télécommunications numériques. Après un état, les motivations et objectifs de la thèse sont présentés. Les problèmes de la diagonalisation conjointe et de la séparation de sources sont définis et un lien entre ces deux domaines est établi. Par la suite, plusieurs algorithmes itératifs de type Jacobi reposant sur une paramétrisation LU sont développés. Pour chacun des algorithmes, on propose de déterminer les matrices permettant de diagonaliser l'ensemble considéré par l'optimisation d'un critère inverse. On envisage la minimisation du critère selon deux approches : la première, de manière directe, et la seconde, en supposant que les éléments de l'ensemble considéré sont quasiment diagonaux. En ce qui concerne l'estimation des différents paramètres du problème, deux stratégies sont mises en œuvre : l'une consistant à estimer tous les paramètres indépendamment et l'autre reposant sur l'estimation indépendante de couples de paramètres spécifiquement choisis. Ainsi, nous proposons trois algorithmes pour la diagonalisation conjointe de matrices complexes symétriques ou hermitiennes et deux algorithmes pour la diagonalisation conjointe d'ensembles de tenseurs symétriques ou non-symétriques ou admettant une décomposition INDSCAL. Nous montrons aussi le lien existant entre la diagonalisation conjointe de tenseurs d'ordre trois et la décomposition canonique polyadique d'un tenseur d'ordre quatre, puis nous comparons les algorithmes développés à différentes méthodes de la littérature. Le bon comportement des algorithmes proposés est illustré au moyen de simulations numériques. Puis, ils sont validés dans le cadre de la séparation de sources de télécommunications numériques.

Abstract

This thesis develops joint diagonalization of matrices and third-order tensors methods for MIMO source separation in the field of digital telecommunications. After a state of the art, the motivations and the objectives are presented. Then the joint diagonalisation and the blind source separation issues are defined and a link between both fields is established. Thereafter, five Jacobi-like iterative algorithms based on an LU parameterization are developed. For each of them, we propose to derive the diagonalization matrix by optimizing an inverse criterion. Two ways are investigated : minimizing the criterion in a direct way or assuming that the elements from the considered set are almost diagonal. Regarding the parameters derivation, two strategies are implemented : one consists in estimating each parameter independently, the other consists in the independent derivation of couple of well-chosen parameters. Hence, we propose three algorithms for the joint diagonalization of symmetric complex matrices or hermitian ones. The first one relies on searching for the roots of the criterion derivative, the second one relies on a minor eigenvector research and the last one relies on a gradient descent method enhanced by computation of the optimal adaptation step. In the framework of joint diagonalization of symmetric, INDSCAL or non symmetric third-order tensors, we have developed two algorithms. For each of them, the parameters derivation is done by computing the roots of the considered criterion derivative. We also show the link between the joint diagonalization of a third-order tensor set and the canonical polyadic decomposition of a fourth-order tensor. We confront both methods through numerical simulations. The good behavior of the proposed algorithms is illustrated by means of computing simulations. Finally, they are applied to the source separation of digital telecommunication signals.