Algorithmes de diagonalisation conjointe par similitude pour la décomposition canonique polyadique de tenseurs : applications en séparation de sources

Rémi André

UNIVERSITÉ DE TOULON

ECOLE DOCTORALE 548
MER ET SCIENCES

Laboratoire d'Informatique et Systèmes (LIS), UMR CNRS 7020
Équipe 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 7 septembre 2018 par

Rémi ANDRÉ

ALGORITHMES DE DIAGONALISATION CONJOINTE PAR SIMILITUDE POUR LA DÉCOMPOSITION CANONIQUE POLYADIQUE DE TENSEURS : APPLICATIONS EN SÉPARATION DE SOURCES

Thèse dirigée par Éric MOREAU et encadrée par Xavier LUCIANI

Jury :

M. Olivier MICHEL	Professeur à l'INP Grenoble, GIPSA-lab	Président
M. David BRIE	Professeur à l'Université de Lorraine, CRAN	Rapporteur
M. Adel BELOUCHRANI	Professeur à l'ENP Alger, Electrical Engineering Department	Rapporteur
M. Martin HAARDT	Professeur à Ilmenau University of Technology, CRL	Examinateur
M. Laurent ALBERA	Maître de conférence HDR à l'Université de Rennes 1, LTSI	Examinateur
M. Éric MOREAU	Professeur à l'Université de Toulon, LIS	Directeur de thèse
M. Xavier LUCIANI	Maître de conférence à l'Université de Toulon, LIS	Encadrant de thèse

Remerciements

Surmonter des épreuves permet d'évoluer. Autant dire que je n'ai jamais autant évolué que pendant ces quatre années de doctorat, tant au niveau professionnel que personnel. Il est donc nécessaire que je remercie sincèrement l'ensemble des personnes qui m'ont permis de devenir le scientifique et l'être humain que je suis aujourd'hui.

Je tiens, tout d'abord, à remercier l'ensemble des membres de mon jury. Je remercie David Brie et Adel Belouchrani pour avoir accepté de rapporter ma thèse ainsi que pour leurs remarques constructives. Je remercie aussi mes examinateurs, Olivier Michel et Martin Haardt, pour la pertinence de leurs questions et pour être venus assister à ma soutenance. Je remercie Laurent Albera en tant qu'examinateur mais aussi pour m'avoir permis de collaborer avec lui pendant ma thèse. Cela fut vraiment très enrichissant.

Je souhaite remercier sincèrement mon directeur de thèse Éric Moreau, sans qui rien de tout cela n'aurait été possible. Cela fut un immense honneur de faire mes premiers pas en tant que jeune chercheur sous sa direction. Il a su me donner une bonne méthodologie de travail à tous les niveaux. Il a aussi fait en sorte que j'enseigne à l'IUT MMI. Enfin, il m'a permis d'échanger avec d'autres chercheurs à travers plusieurs conférences.

J'adresse de vifs remerciements à Xavier Luciani, mon encadrant, pour sa disponibilité. Être son premier doctorant a été un vrai plaisir, nous avons surmonté beaucoup de difficultés ensemble pour finalement aboutir à des résultats dont je suis fier.

Je remercie Nadège Thirion-Moreau pour m'avoir permis de profiter d'une bourse ministérielle durant trois ans. Je la remercie aussi pour nos discussions scientifiques extrêmement constructives. Ses compétences, sa bienveillance ainsi que son humilité forcent le respect.

Je tiens à remercier mes anciens enseignants d'école d'ingénieurs et de master dont certains sont devenus des collègues de travail : Sylvain Maire avec qui j'ai énormément rigolé de tout et de rien et qui a une ouverture d'esprit sans limite. Cyril Prissette avec son armée de robots. Jean-Pierre Rousselot, grand fan de Brassens. Hervé Galiano avec sa méthode de «jarisation» de hautes performances. Olivier Derrien dont le niveau d'exigence m'a permis de me dépasser. Audrey Roman et sa bonne humeur. Frédéric Bouchara et sa poigne musclée. Et enfin Ikhlef Bechar avec qui j'ai refait le monde un nombre incalculable de fois à travers des discussions philosophico-scientifiques.

Évoluer dans l'ancien laboratoire LSIS, devenu LIS, m'a convaincu que le milieu de la recherche est fait pour moi. J'y ai facilement trouvé ma place grâce à la sympathie du personnel de ce laboratoire. Je tiens donc à remercier Vincente Guis avec qui j'ai abordé à peu près tous les sujets de conversation possibles. Je remercie aussi l'équipe «Gauthier» constituée de Jean-Paul Gauthier, Éric Busvelle, Nicolas Boizot et Francesca Chittaro pour leur passion des sciences et leur vocation à la transmettre. Tous mes remerciements

à Adoration Di Santi, la gestionnaire du laboratoire pour son efficacité et sa jovialité. Je remercie Ricard Marxer pour les discussions sur la séparation de sources et pour ses conseils en matière de poursuite de carrière. Merci à Elisabeth Murisasco, la responsable du laboratoire, pour son ouverture au dialogue et sa volonté à débloquer les problèmes. Merci à Aïda Feddaoui d'avoir supporté mes propos absurdes. Je voudrais aussi remercier Jérôme Spagnoli, mon voisin de bureau turbulent ainsi que l'équipe des sauveurs de cachalots Marion Poupard, Julien Ricard, Vincent Roger, Maxence Ferrari et Cosmine. Merci à Ouazna Oukacha pour les petits gâteaux de Kabylie. Un grand merci à Émilien Royer pour m'avoir démontré de manière empirique que les limites de la capacité volumique de l'estomac d'un homme peuvent être inversement proportionnelles à sa corpulence. Et un gigantesque merci à Vincent Marié avec qui j'ai passé d'innombrables soirées à regarder des vidéos de mécanique quantique et qui va se dépêcher de rédiger sa thèse.

Certaines personnes étaient là au début de ma thèse, mais ont dû nous quitter suite un grave choc doctoral. Ils ont été parmi les premiers à m'accueillir dans le laboratoire, je leur en suis très reconnaissant. J'ai nommé Giang Nguyen jeune maman, Amin Bohi fan du Barca, Omar Cherrak toujours armé de son sourire, Cécile Peschoud toujours partante pour faire la fête, Xuan Vu et Jean-Philip Royer notre petit couple de jeunes scientifiques, Diogone Sylla le docteur le plus baraqué du monde. Et pour finir la série, un immense merci à Victor Maurandi, ancien doctorant d'Éric Moreau, qui a été pour moi comme un grand frère au niveau scientifique et qui m'aura enseigné l'art du MMORPG.

Je remercie Gwen Arnaud du laboratoire MIO pour ses conseils avisés tout au long de ma thèse.

Un grand merci à mes amis de l'entretien, Sandro et Mika, leurs menaces quotidiennes et leur second degré vont me manquer.

Merci à ma copine Cécile Réau pour s'être occupée de moi lorsque j'avais la jambe cassée et pour m'avoir conduit au laboratoire pendant ma convalescence. Un grand merci à ses parents pour m'avoir hébergé la semaine de ma soutenance.

Merci à mes amis, Alex, Killian, Ahmed, Pierre, Antoine, Quentin, Romain, Mathilde, Thomas, Amin, Adil, Fanny, Jérôme, Clément, Nina, Nico, Delphy, Laurent G, Laurent F, Stef, Olly, Louie, Stecy, Aaron, Olivier et Nanou ainsi qu'à ma sœur et mon frère pour avoir su être là dans les périodes gaies et sombres de ma vie.

Enfin, je suis infiniment reconnaissant à tous les membres de ma famille pour m'avoir soutenu sans relâche tout au long de mon cursus. Merci également de m'avoir parlé des sciences avec autant de passion depuis mon plus jeune âge. Je remercie plus particulièrement ma mère et mon père de m'avoir poussé à faire des études, j'y ai tellement pris goût que je ne voulais plus m'arrêter, maintenant je suis arrivé au bout. Je vous dois beaucoup.

Résumé

Dans cette thèse, nous nous focalisons sur le développement et la conception d'algorithmes de diagonalisation conjointe par similitude. Le but de ces algorithmes est d'estimer la matrice diagonalisant conjointement un ensemble de matrices diagonalisables dans la même base de vecteurs propres. Les algorithmes de diagonalisation conjointe par similitude permettent, entre autres, de résoudre le problème de décomposition canonique polyadique de tenseurs. Le but de ce problème est d'estimer les matrices facteurs d'un tableau de données multidimensionnel appelé tenseur. La décomposition canonique polyadique est particulièrement utilisée dans les problèmes de séparation de sources et de dé-mélange. L'utilisation de la diagonalisation conjointe par similitude permet de palier certains problèmes dont les autres types de méthode de décomposition canonique polyadique souffrent, tels que le taux de convergence, la sensibilité à la surestimation du nombre de facteurs et la sensibilité aux facteurs corrélés.

Les algorithmes de diagonalisation conjointe par similitude existants sont déclinés en deux versions : l'une pour traiter des données réelles et l'autre pour traiter des données complexes. Dans le cas complexe, une partie des algorithmes donne de bons résultats lorsque le niveau de bruit est faible alors que l'autre partie des algorithmes est plus robuste au bruit mais a un coût de calcul élevé. Nous proposons donc en premier lieu des algorithmes de diagonalisation conjointe par similitude fonctionnant à la fois sur des données réelles et complexes sans modifications et estimant les paramètres inconnus de manière analytique.

Par ailleurs, dans plusieurs applications, les matrices facteurs de la décomposition canonique polyadique contiennent des éléments exclusivement non-négatifs. Prendre en compte cette contrainte de non-négativité permet de rendre les algorithmes de décomposition canonique polyadique plus robustes à la surestimation du nombre de facteurs ou lorsque ces derniers ont un haut degré de corrélation. Nous proposons donc aussi des algorithmes de diagonalisation conjointe par similitude exploitant cette contrainte de non-négativité.

Les simulations numériques proposées montrent que le premier type d'algorithmes développés améliore la précision et le coût de calcul pour des matrices de petite et moyenne taille. Cependant, ces algorithmes sont sensibles à la taille des matrices à diagonaliser. Les simulations numériques montrent aussi que les algorithmes avec contrainte de non-négativité améliorent l'estimation des matrices facteurs lorsque leurs colonnes ont un haut degré de corrélation. Enfin, nos résultats sont validés à travers deux applications de séparation de sources en télécommunications numériques et en spectroscopie de fluorescence.

Mots-clefs :

Traitement du signal ; Diagonalisation conjointe de matrices ; Décomposition matricielle ; Décomposition canonique polyadique ; PARAFAC ; Tenseur ; Analyse de données multidimensionnelles ; Non-négativité ; Optimisation ; Séparation de sources ; Télécommunications numériques ; Spectroscopie de fluorescence.

Abstract

This thesis focuses on the development and the design of several joint eigenvalue decomposition algorithms. These algorithms consist in finding a matrix diagonalizing a set of matrices jointly diagonalizable in the same basis of eigenvectors. These algorithms allow amongst others to solve the canonical polyadic decomposition problem. This problem consists in finding the factor matrices of a multiway array called tensor. The canonical polyadic decomposition is widely used for blind source separation and unmixing problems. Using the joint eigenvalue decomposition to solve the canonical polyadic decomposition problem allows to avoid some problems whose the others canonical polyadic decomposition algorithms generally suffer, such as the convergence rate, the overfactoring sensibility and the correlated factors sensibility.

The existing joint eigenvalue decomposition algorithms are declined in two versions : the first one deals with real data and the second one with complex data. In the complex case, some algorithms give good results when the noise power is low, while the others are more robust to the noise power but have a high numerical cost. Therefore, we first propose algorithms dealing with real and complex data without modifications and analytically estimating the unknown parameters.

Moreover, in some applications, factor matrices of the canonical polyadic decomposition contain only nonnegative values. Taking this constraint into account makes the algorithms more robust to the overfactoring and to the correlated factors. Therefore, we also offer joint eigenvalue decomposition algorithms taking advantage of this nonnegativity constraint.

Suggested numerical simulations show that the first developed algorithms improve the estimation accuracy for small and medium size matrices and reduce the numerical cost in the case of complex data. However they are sensitive to big-size matrices. Our numerical simulations also highlight the fact that our nonnegative joint eigenvalue decomposition algorithms improve the factor matrices estimation when their columns have a high correlation degree. Eventually, we successfully applied our algorithms to two blind source separation problems : one concerning numerical telecommunications and the other concerning fluorescence spectroscopy.

Key words :

Signal processing ; Joint diagonalization ; Matrix decomposition ; Canonical polyadic decomposition ; PARAFAC ; Tensor ; Multiway data analysis ; Nonnegativity ; Optimization ; Blind source separation ; Digital telecommunications ; Fluorescence spectroscopy.

Table des figures

Liste des algorithmes

Table des matières

Chapitre 1

Introduction générale

1.1 Cadre de la thèse

L'analyse de signaux multidimensionnels est devenue un outils important dans de nombreux problèmes de traitement du signal. Ce type d'analyse permet de profiter de différentes diversités d'un signal afin d'en extraire au mieux des informations utiles. Ces signaux peuvent être stockés dans des tableaux à plusieurs dimensions directement après la mesure lorsque le modèle physique le permet. Ces tableaux peuvent aussi être construits en appliquant différents opérateurs d'analyse tels que les statistiques d'ordre supérieur ou les représentations temps-fréquence.

Ainsi, les travaux présentés dans cette thèse portent sur la conception et le développement d'algorithmes de décomposition de tableaux de données multidimensionnels. Le but de ces algorithmes est d'estimer un ensemble de matrices à partir d'un tableau de données multidimensionnel en minimisant une fonction coût judicieusement choisie. Nous nous focalisons sur deux types de décomposition : la décomposition Canonique Polyadique (CP) de tableaux multidimensionnels particuliers appelés tenseurs et la Diagonalisation Conjointe de matrices par Similitude (DCS).

Ces algorithmes peuvent être utilisés pour la séparation aveugle de sources issues d'un mélange linéaire. Le but est alors de retrouver les signaux sources à partir d'observations caractérisant un mélange inconnu de sources. Le problème de séparation de sources modélise de nombreux systèmes multi-capteurs comme des réseaux d'antennes, de microphones ou de capteurs chimiques. Il peut donc être appliqué aux domaines de l'astronomie [1, 2], de l'audio [3, 4], du biomédical [5–9], de la sismique [10, 11], des télécommunications numériques [12–17] ou encore de la spectroscopie de fluorescence [18–22].

1.1.1 Décompositions tensorielles

Les décompositions tensorielles jouent un rôle important dans plusieurs problèmes de traitement du signal comme la psychométrie [23], la chimiométrie [24], la vision par ordinateur [25], la fouille de données [26], les neurosciences [27] ou les télécommunications numériques [17]. Un résumé des différentes décompositions tensorielles et de leurs domaines d'application peut être trouvé dans [28, 29].

Les premiers travaux sur les décompositions tensorielles sont attribués à Hitchcock [30] en 1927. Par la suite, ces travaux ont été popularisés par Tucker [31, 32] en 1963, puis par

Caroll et Chang [33] et Harshman [23] en 1970.

La décomposition tensorielle multilinéaire la plus générale est la décomposition de Tucker. Cette dernière considère qu'un tenseur peut se décomposer à l'aide d'un tenseur cœur et de plusieurs matrices facteurs de manière multilinéaire. Dans le cas où les matrices facteurs sont orthogonales, cette décomposition prend le nom de HOSVD [34]. Lorsque le tenseur cœur est diagonal, il s'agit alors de la décomposition CP. Celle-ci est aussi connue sous de le nom de PARAFAC (de l'anglais *PARalel FACtor analysis*). La HOSVD est généralement utilisée pour des problèmes de compression [35]. La décomposition CP est quant à elle très utilisée dans les problèmes de séparation de sources et d'estimation de mélanges [36]. Elle permet, en effet, de retrouver sur les colonnes de ses matrices facteurs une ou plusieurs estimations des signaux sources et/ou du mélange. L'intérêt d'utiliser la décomposition CP pour des problèmes de séparation de sources est qu'elle possède des conditions d'unicité souvent vérifiées en pratique et qu'elle permet l'estimation de mélanges sous-déterminés [37].

Il existe de nombreux algorithmes de décomposition CP. La plupart d'entre eux consistent à minimiser l'erreur quadratique entre les données observées et les données estimées. Nous pouvons distinguer trois grandes familles d'algorithmes permettant de minimiser cette erreur : les algorithmes de type alterné, les algorithmes de descente et les algorithmes basés sur une réduction de dimensions. Ce dernier type d'approche peut consister à simplement transformer le problème en un problème de décomposition CP de plus petites dimensions ou alors à réécrire le problème en un problème de diagonalisation conjointe de matrices [22, 38, 39].

Les algorithmes de type alterné ou de descente souffrent généralement de problèmes de convergence (minima locaux, faible vitesse de convergence ou coût de calcul élevé). Ils sont très sensibles à la surestimation des facteurs (lorsque le nombre de sources est inconnue) et aux facteurs corrélés [22, 40]. Pour palier ces problèmes, plusieurs auteurs ont montré comment une décomposition CP peut être réécrite en un problème de Diagonalisation Conjointe par Similitude (DCS) [22, 39].

Dans certaines applications comme la spectroscopie de fluorescence [18,20] ou l'imagerie hyperspectrale [41], les matrices facteurs et par conséquent les termes du tenseur qu'elles forment doivent contenir des valeurs non-négatives. La décomposition CP est alors appelée décomposition CP non-négative. Cette caractéristique permet de contraindre la valeur des matrices facteurs estimées et a donné lieu à plusieurs méthodes de décomposition CP sous contrainte de non-négativité. De nombreuses méthodes de décomposition CP non-négatives ont été présentées dans [42]. Prendre en compte les contraintes offertes par une décomposition CP non-négative permet de rendre les algorithmes plus robustes à la surestimation du nombre de facteurs ou d'améliorer leurs performances lorsque les facteurs sont corrélés.

1.1.2 Diagonalisation conjointe de matrices par similitude

La DCS consiste à trouver la matrice de vecteurs propres d'un ensemble de plusieurs matrices de données diagonalisables dans une même base. Il est évident qu'une simple décomposition en éléments propres sur n'importe laquelle de ces matrices permet de diagonaliser conjointement cet ensemble. Cependant, considérer l'ensemble des matrices s'avère utile dans le cas de données bruitées ou alors dans le cas de valeurs propres dégénérées [38].

La DCS fait partie de la famille des problèmes de diagonalisation conjointe de matrices. Ce problème plus général considère un ensemble de matrices conjointement diagonalisables. Les problèmes de diagonalisation conjointe de matrices interviennent dans des problèmes pratiques comme la formation de faisceaux [43–46], le débruitage de signaux [9], l'égalisation aveugle de canaux pour de systèmes de télécommunications multi-entrées multi-sorties [47], l'extraction des différences de marche d'échos Doppler en radar [48], l'analyse en composantes indépendantes [49,50] et bien sur en séparation aveugle de sources [36,51].

La résolution de ce type de problème est généralement basée sur des procédures itératives de type Jacobi [52, 53]. En effet, à chaque itération, la matrice à estimer est mise à jour de manière multiplicative. Le but étant qu'à la fin du processus itératif, les matrices de l'ensemble soient le plus diagonale possible. Les algorithmes de diagonalisation conjointe de matrices diffèrent généralement selon la structure imposée à la matrice de mise à jour et selon la fonction de coût à optimiser. Les algorithmes de diagonalisation conjointe sont généralement déclinés en deux versions, une pour traiter des données réelles et une pour traiter des données complexes. Pour le problème de DCS, les modifications effectuées pour passer à la versions complexes rendent l'étape d'optimisation plus compliquée et amoindrissent les performances [50]. Cela s'explique principalement par le fait que les versions complexes de ces algorithmes estiment séparément soit les parties réelles et imaginaires, soit les modules et les arguments des paramètres. Il y a donc deux fois plus de paramètres à estimer. Dans le cas réel, les algorithmes atteignent généralement tous un bon niveau de performances. Dans le cas complexe, une première famille d'algorithmes permet d'obtenir de bons résultats de manière peu coûteuse mais voit ses performances décroître fortement avec quand le niveau de bruit augmente. Une seconde famille d'algorithmes permet d'obtenir de meilleurs résultats lorsque le niveau de bruit est élevé, cependant le coût de calcul de ces algorithmes est très élevé.

1.2 Objectif et plan de la thèse

Dans le but d'améliorer les performances des algorithmes de DCS, nous proposons dans cette thèse de nouveaux algorithmes fonctionnant sur des données réelles et complexes sans modifications. Les stratégies proposées permettront ainsi d'obtenir de meilleures performances que celles fournies par les algorithmes existants dans le cas complexe.

Par ailleurs, il n'existe à ce jour, à notre meilleure connaissance, aucun algorithme de décomposition CP non-négative basé sur une étape de DCS. Nous utiliserons ainsi les contraintes offertes par l'algorithme de décomposition CP introduit dans [22] pour proposer des algorithmes de DCS sous contraintes de non-négativité.

Ce manuscrit est donc organisé de la manière suivante : La fin de ce chapitre est consacrée aux notations et aux outils mathématiques utilisés par la suite. Dans le second chapitre, nous présentons les outils utiles d'algèbre multilinéaire et les différentes décompositions tensorielles en nous focalisant sur la décomposition CP. Nous terminons ce chapitre par un état de l'art des différents algorithmes de décomposition CP. Le chapitre 3 est consacré à la diagonalisation conjointe de matrices. Nous y présentons d'abord les différents problèmes de diagonalisation conjointe et les stratégies de résolutions déployées, puis nous faisons l'état de l'art des différents algorithmes de DCS. Nos travaux sur les algorithmes de DCS sans contraintes fonctionnant à la fois sur des données complexes ou

réelles font l'objet du chapitre 4. Nous y présentons les différentes stratégies que nous avons mis en œuvre et nous comparons les algorithmes proposés à ceux de la littérature. Nos algorithmes de DCS sous contraintes de non-négativité sont présentés dans le chapitre 5. Nous explicitons les différentes méthodes proposées et nous les comparons à plusieurs méthodes de décomposition CP classiques. Enfin, dans le chapitre 6 nous montrons l'intérêt de nos méthodes à travers deux applications : la séparation de sources de signaux de télécommunications numériques et l'analyse de signaux de fluorescence.

1.3 Liste des contributions

Les travaux présentés dans les chapitres 4, 5 et 6 de cette thèse ont donné lieu à cinq publications dans des conférences et une dans une revue scientifique internationale :

1. R. André, T. Trainini and X. Luciani and E. Moreau, A fast algorithm for joint eigenvalue decomposition of real matrices, *EUropean SIgnal Processing COnference (EUSIPCO'2015)*, Nice, France, 2015.

2. R. André, X. Luciani and E. Moreau, A coupled joint eigenvalue decomposition algorithm for canonical polyadic decomposition of tensors, *IEEE Sensor Array and Multichannel signal processing workshop (SAM'2016)*, Rio de Janeiro, Brazil, 2016.

3. R. André, L. Albera, X. Luciani and E. Moreau, On JEVD of semi-definite positive matrices and CPD of nonnegative tensors, *IEEE Sensor Array and Multichannel signal processing workshop (SAM'2016)*, Rio de Janeiro, Brazil, 2016.

4. R. André, X. Luciani and E. Moreau, Diagonalisation conjointe par similitude avec contrainte de non négativité sur les valeurs propres, *GRETSI, XXVIeme colloque sur le Traitement du Signal et des Images*, Juan-les-Pins, France, 2017.

5. R. André, X. Luciani and E. Moreau, A fast algorithm for the CP decomposition of large tensors, *Statistics and Data Science : new challenges, new generations (SIS)*, Florence, Italie, 2017.

6. R. André, X. Luciani and E. Moreau, A new class of block coordinate algorithms for the joint eigenvalue decomposition of complex matrices, *Signal Processing*, volume 145, pages 78 - 90, 2018.

1.4 Notations et outils mathématiques

Nous présentons ici les notations utilisées dans cette thèse ainsi que la définition et les propriétés de différents produits, opérateurs et décompositions matriciels. De plus, dans le but de calculer ultérieurement la complexité numérique des méthodes proposées et existantes, nous donnons aussi le nombre de multiplications effectuées par chacun des produits présentés. Nous calculons ici le nombre de multiplications sans tenir compte des 0 et des 1 éventuels. Les définitions des opérateurs et décompositions matricielles présentées ainsi que leurs propriétés sont disponibles de manière plus détaillée dans [54, 55].

1.4.1 Notations

Dans ce manuscrit, nous notons $[a;b]_{\mathbb{N}}$ l'intervalle de $\mathbb{N}$ contenant les entiers naturels compris entre a et b. Un scalaire sera noté à l'aide d'une lettre minuscule x. Lorsque un scalaire représente une dimension d'un tableau de données (taille d'un vecteur, nombre de lignes ou de colonnes d'une matrice...), il sera noté à l'aide d'une lettre majuscule X. Un vecteur sera noté en gras et à l'aide d'une lettre minuscule $\mathbf{v}$. Le $i^{\text{ème}}$ élément de ce vecteur sera noté v_i. La norme 2 d'un vecteur $\mathbf{v}$ sera notée $||\mathbf{v}||_2$. Une matrice sera notée en gras et à l'aide d'une lettre majuscule $\mathbf{M}$. L'élément (i,j) d'une telle matrice sera noté $M_{i,j}$ et sa $j^{\text{ème}}$ colonne sera notée $\mathbf{m}_j$. Les opérateurs matriciels classiques seront notés de la manière suivante :

- $(\mathbf{M})^T$ pour l'opérateur transposé.
- $(\mathbf{M})^H$ pour l'opérateur transposé-conjugué ou transposé hermitien.
- $(\mathbf{M})^{-1}$ pour l'opérateur inverse.
- $(\mathbf{M})^\dagger$ pour la pseudo-inverse au sens de Moore-Penrose.

$\mathbf{I}_R$ dénotera la matrice identité de dimension R. Un tableau de données de plus de deux dimensions sera noté en gras, à l'aide d'une lettre calligraphique majuscule $\mathcal{T}$. Un élément $(i_1,\ldots,i_Q)$ d'un tableau de données à Q dimensions ($Q>2$) sera noté $T_{i_1,\ldots,i_Q}$. Le conjugué d'un nombre complexe c sera noté $\bar{c}$ et son module sera noté $|c|$.

1.4.2 Produits matriciels

Le produit de Hadamard. Le produit de Hadamard [56] est le produit terme à terme entre deux matrices $\mathbf{A}\in\mathbb{C}^{I\times J}$ et $\mathbf{B}\in\mathbb{C}^{I\times J}$ de mêmes dimensions. Nous le notons $\mathbf{A}\boxdot\mathbf{B}\in\mathbb{C}^{I\times J}$. Les termes de la matrice résultante du produit de Hadamard s'expriment donc

$$\forall(i,j)\in[1;I]_{\mathbb{N}}\times[1;J]_{\mathbb{N}},\quad(\mathbf{A}\boxdot\mathbf{B})_{i,j}=A_{i,j}B_{i,j}. \tag{1.1}$$

Propriétés. Considérant une matrice $\mathbf{C}\in\mathbb{C}^{I\times J}$ et un scalaire $\alpha\in\mathbb{C}$, le produit de Hadamard admet les propriétés suivantes :

- associativité : $\mathbf{A}\boxdot\mathbf{B}\boxdot\mathbf{C}=(\mathbf{A}\boxdot\mathbf{B})\boxdot\mathbf{C}=\mathbf{A}\boxdot(\mathbf{B}\boxdot\mathbf{C})$
- distributivité : $\mathbf{A}\boxdot(\mathbf{B}+\mathbf{C})=(\mathbf{A}\boxdot\mathbf{B})+(\mathbf{A}\boxdot\mathbf{C})$
- commutativité : $\mathbf{A}\boxdot\mathbf{B}=\mathbf{B}\boxdot\mathbf{A}$
- $(\mathbf{A}\boxdot\mathbf{B})^T=\mathbf{A}^T\boxdot\mathbf{B}^T$
- $\alpha(\mathbf{A}\boxdot\mathbf{B})=(\alpha\mathbf{A})\boxdot\mathbf{B}=\mathbf{A}\boxdot(\alpha\mathbf{B})$
- Si $\mathbf{A}$ et $\mathbf{B}$ sont des matrices diagonales, alors $\mathbf{A}\boxdot\mathbf{B}=\mathbf{AB}$

Nombre de multiplications du produit de Hadamard. Pour réaliser le produit $\mathbf{A}\boxdot\mathbf{B}$, il faut effectuer IJ multiplications.

Le produit de Kronecker. Le produit de Kronecker $\mathbf{A} \otimes \mathbf{B} \in \mathbb{C}^{IK \times JR}$ [57] entre deux matrices $\mathbf{A} \in \mathbb{C}^{I \times J}$ et $\mathbf{B} \in \mathbb{C}^{K \times R}$ est défini de la manière suivante :

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} A_{1,1}\mathbf{B} & A_{1,2}\mathbf{B} & \cdots & A_{1,J}\mathbf{B} \\ A_{2,1}\mathbf{B} & A_{2,2}\mathbf{B} & \cdots & A_{2,J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ A_{I,1}\mathbf{B} & A_{I,2}\mathbf{B} & \cdots & A_{I,J}\mathbf{B} \end{pmatrix}. \tag{1.2}$$

Propriétés. Considérant $\mathbf{C} \in \mathbb{C}^{K \times R}$ et un scalaire $\alpha \in \mathbb{C}$, le produit de Kronecker possède les propriétés suivantes :

- associativité : $\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C} = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})$
- distributivité :
  - ▷ $\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) + (\mathbf{A} \otimes \mathbf{C})$
  - ▷ $(\mathbf{B} + \mathbf{C}) \otimes \mathbf{A} = (\mathbf{B} \otimes \mathbf{A}) + (\mathbf{C} \otimes \mathbf{A})$
- non-commutativité : $\mathbf{A} \otimes \mathbf{B} \neq \mathbf{B} \otimes \mathbf{A}$
- $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$
- si $\mathbf{A}$ et $\mathbf{B}$ sont carrées et inversibles alors $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$
- $(\mathbf{A} \otimes \mathbf{B})^\dagger = \mathbf{A}^\dagger \otimes \mathbf{B}^\dagger$
- $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$
- $\alpha(\mathbf{A} \otimes \mathbf{B}) = (\alpha\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (\alpha\mathbf{B})$

Nombre de multiplications du produit de Kronecker. Le produit $a_{ij}\mathbf{B}$ coûte KR multiplications. Donc, selon l'expression (1.2), il est nécessaire de calculer $IJKR$ multiplications pour effectuer $\mathbf{A} \otimes \mathbf{B}$.

Le produit de Khatri-Rao. Soient deux matrices $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_J] \in \mathbb{C}^{I \times J}$ et $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_J] \in \mathbb{C}^{K \times J}$. Le produit de Khatri-Rao [58] entre $\mathbf{A}$ et $\mathbf{B}$, noté $\mathbf{A} \odot \mathbf{B} \in \mathbb{C}^{IK \times J}$, est défini comme le produit de Kronecker entre les colonnes de $\mathbf{A}$ avec les colonnes de $\mathbf{B}$ correspondantes, ainsi :

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1,\ \mathbf{a}_2 \otimes \mathbf{b}_2, \cdots,\ \mathbf{a}_J \otimes \mathbf{b}_J]. \tag{1.3}$$

Propriétés. Soit $\mathbf{C}$ une matrice ayant le même nombre de colonnes que $\mathbf{A}$ et $\mathbf{B}$ et $\alpha \in \mathbb{C}$ un scalaire. Le produit de Khatri-Rao possède les propriétés suivantes :

- associativité : $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = (\mathbf{B} \odot \mathbf{C}) \odot \mathbf{A}$
- distributivité : $(\mathbf{A} + \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot \mathbf{C} + \mathbf{B} \odot \mathbf{C}$
- non-commutativité : $\mathbf{A} \odot \mathbf{B} \neq \mathbf{B} \odot \mathbf{A}$
- $(\mathbf{A} \odot \mathbf{B})^T(\mathbf{A} \odot \mathbf{B}) = \mathbf{A}^T\mathbf{A} \boxdot \mathbf{B}^T\mathbf{B}$

- $\alpha(\mathbf{A} \odot \mathbf{B}) = (\alpha\mathbf{A}) \odot \mathbf{B} = \mathbf{A} \odot (\alpha\mathbf{B})$
- Le produit de Katri-Rao peut se réécrire à l'aide du produit matriciel simple :

$$\mathbf{A} \odot \mathbf{B} = \begin{pmatrix} \mathbf{B}\mathbf{D}_{\mathbf{A}}^{(1)} \\ \mathbf{B}\mathbf{D}_{\mathbf{A}}^{(2)} \\ \vdots \\ \mathbf{B}\mathbf{D}_{\mathbf{A}}^{(I)} \end{pmatrix}, \tag{1.4}$$

où $\mathbf{D}_{\mathbf{A}}^{(i)}$ $\forall i \in [1; I]_{\mathbb{N}}$ est une matrice diagonale contenant la $i^{\text{ème}}$ ligne de la matrice $\mathbf{A}$.

Nombre de multiplications du produit de Khatri-Rao. Pour réaliser le produit de Kronecker $\mathbf{a}_j \otimes \mathbf{b}_j$, IK multiplications sont nécessaires. Ainsi pour réaliser le produit de Khatri-Rao, IJK multiplications sont nécessaires.

Nous résumons la complexité numérique des différents produits matriciels dans le tableau 1.1.

Produits matriciels	taille de la $1^{\text{ère}}$ matrice	taille de la $2^{\text{ème}}$ matrice	complexité numérique
Produit matriciel classique	$I \times R$	$R \times J$	IJR
Produit de Hadamard	$I \times J$	$I \times J$	IJ
Produit de Kronecker	$I \times J$	$K \times R$	$IJKR$
Produit de Khatri-Rao	$I \times J$	$K \times J$	IJK

TABLE 1.1: Tableau récapitulatif des complexités numériques des différents produits matriciels.

1.4.3 Opérateurs matriciels

Opérateurs $\mathsf{Diag}\{\bullet\}$ et $\mathsf{ZDiag}\{\bullet\}$. L'opérateur $\mathsf{Diag}\{\bullet\}$ met à zéro les termes hors diagonaux de la matrice en argument. Soit une matrice $\mathbf{M}$ de dimension R, alors

$$\mathsf{Diag}\{\mathbf{M}\} = \begin{pmatrix} M_{1,1} & 0 & \dots & \dots & 0 \\ 0 & M_{2,2} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & M_{R,R} \end{pmatrix}. \tag{1.5}$$

L'opérateur $\mathsf{ZDiag}\{\bullet\}$, quant à lui, met à zéro les termes diagonaux de la matrice en argument.

Exemple : Si $\mathbf{M} = \begin{pmatrix} 1 & 5 & 6 \\ 8 & 6 & 3 \\ 10 & 7 & 4 \end{pmatrix}$, alors

$$\mathsf{ZDiag}\{\mathbf{M}\} = \begin{pmatrix} 0 & 5 & 6 \\ 8 & 0 & 3 \\ 10 & 7 & 0 \end{pmatrix}.$$

Nous avons alors

$$\mathsf{ZDiag}\{\mathbf{M}\} = \mathbf{M} - \mathsf{Diag}\{\mathbf{M}\}. \tag{1.6}$$

La trace d'une matrice. Soit une matrice carrée $\mathbf{M}$ de dimension R. La trace de $\mathbf{M}$, notée $\mathsf{trace}\{\mathbf{M}\}$, est la somme de ses éléments diagonaux, i.e

$$\mathsf{trace}\{\mathbf{M}\} = \sum_{r=1}^{R} M_{r,r}. \tag{1.7}$$

Propriétés. Soient deux matrices carrées $\mathbf{A}$ et $\mathbf{B}$ de même dimension et un scalaire α. La trace admet les propriétés suivantes :

- $\mathsf{trace}\{\mathbf{A}+\mathbf{B}\} = \mathsf{trace}\{\mathbf{A}\} + \mathsf{trace}\{\mathbf{B}\}$
- $\mathsf{trace}\{\alpha\mathbf{A}\} = \alpha\mathsf{trace}\{\mathbf{A}\}$
- $\mathsf{trace}\{\mathbf{A}^T\} = \mathsf{trace}\{\mathbf{A}\}$
- $\mathsf{trace}\{\mathbf{AB}\} = \mathsf{trace}\{\mathbf{BA}\}$
- La trace est invariante par similitude, *i.e* $\mathsf{trace}\{\mathbf{BAB}^{-1}\} = \mathsf{trace}\{\mathbf{A}\}$
- Soit $F(\mathbf{X})$ une fonction dérivable par rapport à chaque composante de $\mathbf{X}$, alors

$$\frac{\partial \mathsf{trace}\{F(\mathbf{X})\}}{\partial \mathbf{X}} = f(\mathbf{X})^T \tag{1.8}$$

avec $f(\bullet)$ la dérivée scalaire de $F(\bullet)$.

Exemple :

$$\frac{\partial \mathsf{trace}\{\mathbf{AXB}\}}{\partial \mathbf{X}} = \mathbf{A}^T\mathbf{B}^T. \tag{1.9}$$

La norme de Frobenius. Soit $||\mathbf{M}||_F$ la norme de Frobenius de la matrice $\mathbf{M} \in \mathbb{C}^{I\times J}$. Celle-ci est définie par

$$||\mathbf{M}||_F = \sqrt{\sum_{i=1}^{I}\sum_{j=1}^{J} |M_{i,j}|^2}. \tag{1.10}$$

La norme de Frobenius peut aussi s'écrire en fonction de la trace :

- Soit $\mathbf{M} \in \mathbb{R}^{I\times J}$, alors $||\mathbf{M}||_F = \mathsf{trace}\{\mathbf{MM}^T\}^{\frac{1}{2}}$
- Soit $\mathbf{M} \in \mathbb{C}^{I\times J}$, alors $||\mathbf{M}||_F = \mathsf{trace}\{\mathbf{MM}^H\}^{\frac{1}{2}}$

La norme L_1. Soit $||\mathbf{M}||_1$ la norme L_1 de la matrice $\mathbf{M} \in \mathbb{C}^{I\times J}$. Celle-ci est définie par

$$||\mathbf{M}||_1 = \sum_{i=1}^{I}\sum_{j=1}^{J} |M_{i,j}|. \tag{1.11}$$

Opérateur de vectorisation $\mathsf{vec}\{\bullet\}$ **.** L'opérateur $\mathsf{vec}\{\bullet\}$ permet de déplier une matrice en vecteur en juxtaposant chacune de ses colonnes les unes sous les autres. Ainsi, considérant une matrice $\mathbf{M} \in \mathbb{C}^{I\times J}$, nous avons

$$\mathsf{vec}\{\mathbf{M}\} = \begin{pmatrix} M_{1,1} \\ \vdots \\ M_{I,1} \\ M_{1,2} \\ \vdots \\ M_{I,2} \\ \vdots \\ M_{1,J} \\ \vdots \\ M_{I,J} \end{pmatrix} \tag{1.12}$$

Propriétés. Soient $\mathbf{A} \in \mathbb{C}^{I\times J}$, $\mathbf{B} \in \mathbb{C}^{J\times N}$, $\mathbf{C} \in \mathbb{C}^{N\times K}$, $\mathbf{E} \in \mathbb{C}^{I\times J}$ et $\alpha \in \mathbb{C}$. Nous donnons les propriétés suivantes pour l'opérateur $\mathsf{vec}\{\bullet\}$:

- $\mathsf{vec}\{\alpha\mathbf{A}\} = \alpha\mathsf{vec}\{\mathbf{A}\}$
- $\mathsf{vec}\{\mathbf{A}+\mathbf{E}\} = \mathsf{vec}\{\mathbf{A}\} + \mathsf{vec}\{\mathbf{E}\}$
- $\mathsf{vec}\{\mathbf{A}\}^T\mathsf{vec}\{\mathbf{E}\} = \mathsf{trace}\{\mathbf{A}^T\mathbf{E}\}$
- $\mathsf{vec}\{\mathbf{ABC}\} = (\mathbf{C}^T \otimes \mathbf{A})\mathsf{vec}\{\mathbf{B}\}$

Propriété 1.4.1 (Équation de Sylvester)**.** *Soit l'équation de Sylvester*

$$\mathbf{AX} + \mathbf{XB} = \mathbf{C}. \tag{1.13}$$

L'opérateur de vectorisation permet de résoudre l'équation matricielle de Sylvester lorsque la matrice $(\mathbf{I}\otimes\mathbf{A} + \mathbf{B}^T\otimes\mathbf{I})$ *est inversible. En effet, en vectorisant* (1.13) *nous obtenons*

$$\mathsf{vec}\{\mathbf{X}\} = (\mathbf{I}\otimes\mathbf{A} + \mathbf{B}^T\otimes\mathbf{I})^{-1}\mathsf{vec}\{\mathbf{C}\} \tag{1.14}$$

Opérateur $\succeq$**.** Soit $\mathbf{A} \in \mathbb{R}^{I\times J}$ et $b \in \mathbb{R}$. L'opérateur $\succeq$ est alors défini comme

$$\mathbf{A} \succeq b \equiv \forall (i,j) \in [1;I]_{\mathbb{N}} \times [1;J]_{\mathbb{N}}\ , A_{i,j} \geq b. \tag{1.15}$$

Opérateur de projection sur $\mathbb{R}^+$**.** Soit $\mathbf{A} \in \mathbb{R}^{I\times J}$, l'opérateur de projection sur $\mathbb{R}^+$ est alors défini par :

$$\left([\mathbf{A}]_+\right)_{i,j} = \begin{cases} A_{i,j} & \text{si} \quad A_{i,j} \geq 0 \\ 0, & \text{sinon} \end{cases} \tag{1.16}$$

1.4.4 Quelques décompositions matricielles utiles

La décomposition LU. Toute matrice $\mathbf{A} \in \mathbb{C}^{R \times R}$ peut être décomposée comme

$$\mathbf{A} = \mathbf{DPLU}, \tag{1.17}$$

avec $\mathbf{D}$ une matrice diagonale, $\mathbf{P}$ une matrice de permutation, $\mathbf{L}$ une matrice triangulaire inférieure ne contenant que des 1 sur sa diagonale et $\mathbf{U}$ une matrice triangulaire supérieure ne contenant que des 1 sur sa diagonale.

La décomposition QR. Toute matrice $\mathbf{A} \in \mathbb{R}^{R \times R}$ peut s'écrire sous la forme

$$\mathbf{A} = \mathbf{DQR}, \tag{1.18}$$

avec $\mathbf{D}$ une matrice diagonale, $\mathbf{R}$ une matrice triangulaire ne contenant que des 1 sur sa diagonale et $\mathbf{Q}$ une matrice orthogonale. Si $\mathbf{A} \in \mathbb{C}^{R \times R}$ alors $\mathbf{Q}$ est une matrice unitaire.

La décomposition polaire. Toute matrice $\mathbf{A} \in \mathbb{R}^{R \times R}$ peut s'écrire sous la forme

$$\mathbf{A} = \mathbf{QH}, \tag{1.19}$$

avec $\mathbf{H}$ une matrice symétrique et $\mathbf{Q}$ une matrice orthogonale. Dans le cas complexe $\mathbf{H}$ est une matrice symétrique hermitienne et $\mathbf{Q}$ est une matrice unitaire.

La décomposition polaire algébrique. Nous appelons matrice symétrique complexe toute matrice $\mathbf{H} \in \mathbb{C}^{R \times R}$ vérifiant $\mathbf{H} = \mathbf{H}^T$. De même, nous appelons matrice orthogonale complexe toute matrice $\mathbf{Q} \in \mathbb{C}^{R \times R}$ vérifiant $\mathbf{QQ}^T = \mathbf{I}$.

Soit $\mathbf{A} \in \mathbb{C}^{R \times R}$ une matrice inversible. Il est montré dans [59] que $\mathbf{A}$ admet une décomposition polaire algébrique

$$\mathbf{A} = \mathbf{QH}, \tag{1.20}$$

avec $\mathbf{H}$ une matrice symétrique complexe et $\mathbf{Q}$ une matrice orthogonale complexe.

Décomposition de Schur. Toute matrice carrée $\mathbf{M} \in \mathbb{C}^{R \times R}$ peut s'écrire

$$\mathbf{M} = \mathbf{QTQ}^H, \tag{1.21}$$

où $\mathbf{Q} \in \mathbb{C}^{R \times R}$ est une matrice unitaire et $\mathbf{T}$ est une matrice triangulaire.

Il est intéressant de noter que la matrice $\mathbf{T}$ contient sur sa diagonale les valeurs propres de la matrice $\mathbf{M}$.

Décomposition en valeur singulière. Toute matrice $\mathbf{A} \in \mathbb{C}^{M \times N}$ peut être décomposée de la manière suivante :

$$\mathbf{A} = \mathbf{USV}^H, \tag{1.22}$$

avec

- $\mathbf{U} \in \mathbb{C}^{M \times M}$ une matrice unitaire,

- $\mathbf{S} \in \mathbb{C}^{M \times N}$ est une matrice ne contenant que des zéros, excepté sur sa diagonale principale,
- $\mathbf{V} \in \mathbb{C}^{N \times N}$ une matrice unitaire.

Cette décomposition en valeur singulière est notée SVD (de l'anglais « Singular Value Decomposition »). Une SVD peut être tronquée à l'ordre Q, nous aurons alors

- $\mathbf{U} \in \mathbb{C}^{M \times Q}$ telle que $\mathbf{U}^H\mathbf{U} = \mathbf{I}_Q$,
- $\mathbf{S} \in \mathbb{C}^{Q \times Q}$ est une matrice diagonale,
- $\mathbf{V} \in \mathbb{C}^{N \times Q}$ telle que $\mathbf{V}^H\mathbf{V} = \mathbf{I}_Q$.

L'égalité (1.22) reste vraie si et seulement si Q est supérieur ou égal au rang de la matrice $\mathbf{A}$.

Théorème 1.4.1. *Théorème d'Eckart-Young.*

Soient $\mathbf{A} \in \mathbb{C}^{M \times N}$ et $\mathbf{A}_Q = \mathbf{USV}$ la matrice correspondant à la SVD de $\mathbf{A}$ tronquée au rang Q avec $\mathbf{S}$ contenant les Q valeurs singulières de plus grand module de $\mathbf{A}$. Nous avons alors

$$\mathbf{A}_Q = \underset{\mathbf{X}}{\mathsf{argmax}} ||\mathbf{A} - \mathbf{X}||_F \tag{1.23}$$

où $\mathbf{X}$ est une matrice de rang Q. Ainsi, $\mathbf{A}_Q$ est la matrice de rang Q approximant le mieux la matrice $\mathbf{A}$ au sens de la norme de Fobenius.

Chapitre 2

La décomposition Canonique Polyadique de tenseurs

Dans ce chapitre, nous nous intéressons au problème de décomposition Canonique Polyadique (CP) de tenseurs et aux différentes méthodes de l'état de l'art permettant de le résoudre. Pour pouvoir présenter et situer le problème de décomposition CP, il est nécessaire d'introduire en premier lieu certains outils mathématiques spécifiques à l'algèbre multilinéaire ainsi que les principales décompositions tensorielles utilisées en traitement du signal.

2.1 Tenseur, définitions et outils mathématiques associés

2.1.1 Définitions

Tenseur. Le terme tenseur peut avoir différentes significations selon le domaine scientifique. Ici, un tenseur définit une application multilinéaire. Lorsque les bases des espaces vectoriels sont fixées, un tenseur correspond à un tableau multidimensionnel [28,60]. Ainsi dans la suite de cette thèse, un tenseur sera assimilé à un tableau de nombres multidimensionnel.

Le nombre de dimensions d'un tenseur est appelé l'ordre du tenseur. Un tenseur d'ordre Q peut être généré par au minimum Q vecteurs à l'aide du produit externe (voir équation (2.2)).

Les dimensions d'un tenseur sont aussi appelées modes. Un exemple de tenseur d'ordre 3 et de dimensions $4\times5\times3$ est illustré sur la figure 2.1. Un élément quelconque d'un tenseur $\mathcal{T}$ d'ordre Q est noté $T_{i_1,i_2,\ldots,i_Q}$ (voir figure 2.1). Un tenseur dont toutes les dimensions sont égales à R est dit de dimension R.

Tenseur diagonal. La diagonale d'un tenseur $\mathcal{T}$ d'ordre Q représente les éléments du tenseur de coordonnées égales *i.e.* $\forall q \in [1;Q]_{\mathbb{N}}\ T_{i_q,i_q,\ldots,i_q}$. Un tenseur diagonal est un tenseur dont toutes les dimensions sont égales et ne contenant que des valeurs nulles hors des éléments diagonaux.

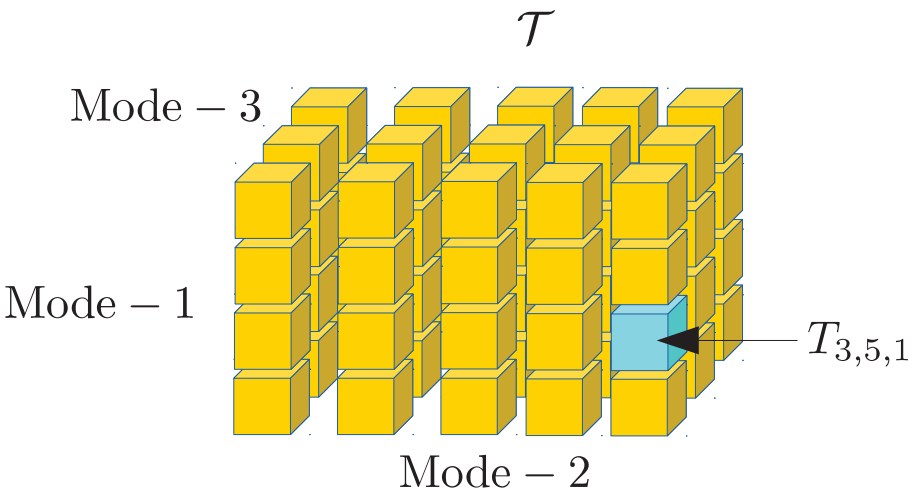

FIGURE 2.1: Schéma d'un champs de données à trois dimensions dit tenseur d'ordre 3.

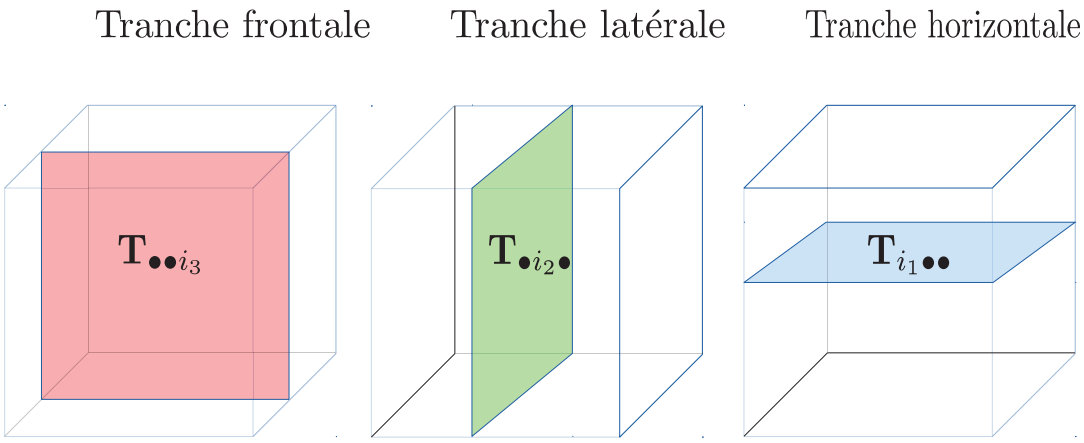

FIGURE 2.2: Schéma illustrant les trois différents types de tranches d'un tenseur d'ordre 3.

Tranche d'un tenseur. Une tranche d'un tenseur est une matrice obtenue en fixant tous les modes d'un tenseur sauf deux. Soit $\mathcal{T}$ un tenseur d'ordre Q dans $\mathbb{C}^{I_1 \times \ldots \times I_Q}$, nous notons une tranche quelconque de ce tenseur $\mathbf{T}_{i_1 \ldots \bullet \ldots \bullet \ldots i_Q}$ où $\bullet$ représente les indices des modes libres du tenseur $\mathcal{T}$. Ainsi un tenseur d'ordre Q aura au plus $(Q-1)Q/2$ types de tranches différentes. La figure 2.2 permet de visualiser les trois différentes tranches (frontales, latérales et horizontales) d'un tenseur d'ordre 3.

2.1.2 Dépliements d'un tenseur

De la même manière qu'une matrice peut être dépliée en vecteur (opération de vectorisation), un tenseur peut être déplié en un tenseur d'ordre plus petit et en particulier en matrice [19]. Dans le cas de la vectorisation d'une matrice, les deux modes de cette dernière sont fusionnés et ainsi la multiplication de ses deux dimensions donne le nombre de lignes du vecteur obtenu. Ce raisonnement peut être étendu aux tenseurs.

Dépliements d'un tenseur d'ordre trois. Il existe plusieurs manières de déplier un tenseur d'ordre trois selon ses différents modes. Une matrice de dépliement d'un tenseur est obtenue en juxtaposant ses tranches frontales, horizontales ou verticales. Par exemple, comme le montre la figure 2.3, un mode peut être isolé sur les lignes de la matrice de dépliement, les autres modes sont alors regroupés sur ses colonnes. Pour un tenseur d'ordre 3, nous considérons les trois matrices de dépliement suivantes (parmi les six possibles) :

- la matrice de dépliement dans le premier mode (concaténation des tranches frontales) a I_1 lignes et $I_2 I_3$ colonnes (voir figure 2.3). Nous la notons $\mathbf{T}_{(1)}$,
- la matrice de dépliement dans le second mode (concaténation des tranches horizontales) a I_2 lignes et $I_3 I_1$ colonnes (voir figure 2.3). Nous la notons $\mathbf{T}_{(2)}$,
- la matrice de dépliement dans le troisième mode (concaténation des tranches latérales) a I_3 lignes et $I_1 I_2$ colonnes (voir figure 2.3). Nous la notons $\mathbf{T}_{(3)}$.

Dépliements d'un tenseur d'ordre supérieur à trois. Les manières de déplier un tenseur d'ordre quelconque sont donc très nombreuses. Pour simplifier, nous choisissons de déplier un tenseur en juxtaposant chacune de ses tranches selon seulement un de ses modes. Nous notons $\mathbf{T}_{(q)}$ la matrice de dépliement du tenseur $\mathcal{T}$ selon le mode q. La matrice $\mathbf{T}_{(q)}$ a alors pour dimensions $I_q \times \prod_{\substack{k=1 \\ k \neq q}}^{Q} I_k$. Ainsi, dans cette thèse, la matrice de dépliement $\mathbf{T}_{(q)}$ aura un nombre de colonnes égale à la multiplication de toutes les dimensions du tenseur $\mathcal{T}$ à l'exception de celle du mode q qui sera égale à son nombre de lignes.

2.1.3 Norme de Frobenius d'un tenseur

De la même manière que pour une matrice, la norme de Frobenius $||\mathcal{T}||_F$ d'un tenseur $\mathcal{T}$ est définie comme la racine carrée de la somme du module au carré de tous ses éléments [28]. Par exemple, pour un tenseur d'ordre 3, nous avons :

$$||\mathcal{T}||_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \sum_{i_3=1}^{I_3} |T_{i_1,i_2,i_3}|^2}. \tag{2.1}$$

Ainsi, il est évident que la norme de Frobenius d'un tenseur est égale à la norme de Frobenius de ses matrices de dépliement.

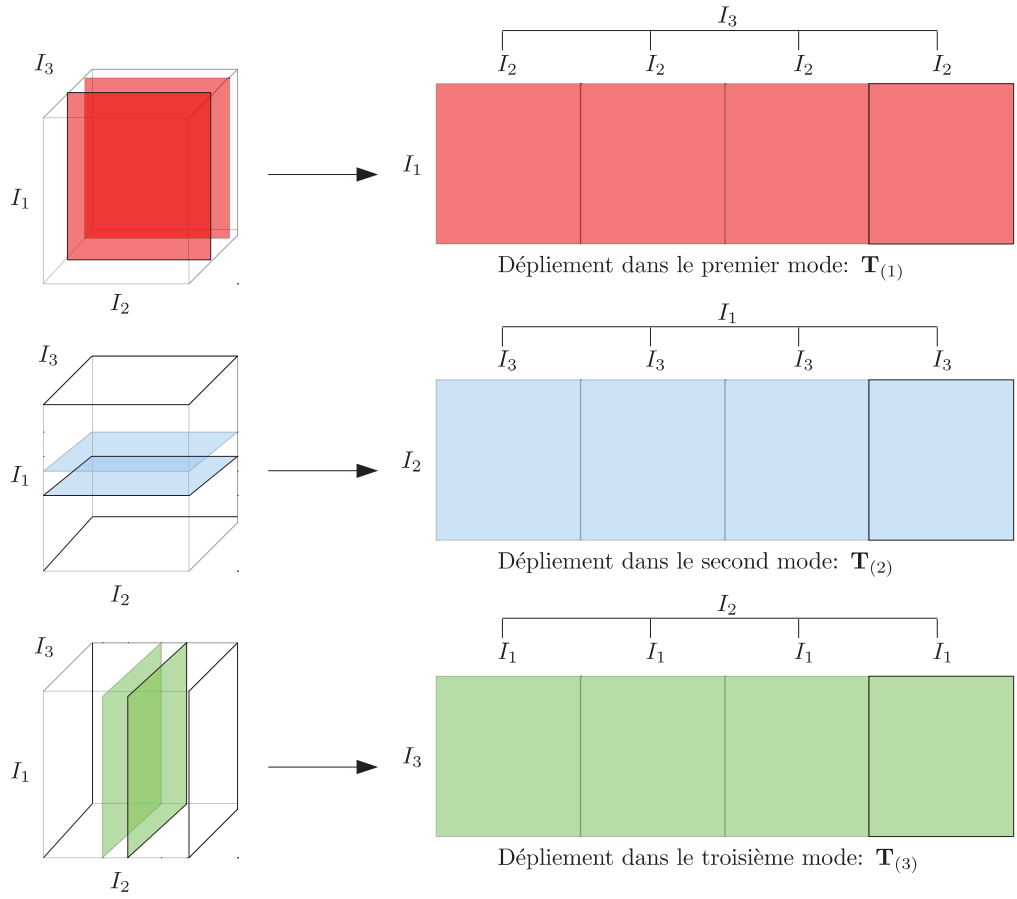

FIGURE 2.3: Schéma explicatif du dépliement matriciel d'un tenseur d'odre 3.

2.1.4 Produits sur les tenseurs

Nous présentons ici deux produits faisant intervenir des tenseurs [61]. Ces produits permettent notamment de reformuler les décompositions tensorielles de manières compactes.

Le produit externe. Soient deux tenseurs $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ et $\mathcal{S} \in \mathbb{C}^{J_1 \times J_2 \times \cdots \times J_M}$, le produit externe entre $\mathcal{T}$ et $\mathcal{S}$ est défini par

$$(\mathcal{T} \circ \mathcal{S})_{i_1,i_2,\ldots,i_N,j_1,j_2,\ldots,j_M} = T_{i_1,i_2,\ldots,i_N} S_{j_1,j_2,\ldots,j_M}. \tag{2.2}$$

Donc $\mathcal{T} \circ \mathcal{S} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$.

Cas particulier important : le produit externe entre deux vecteurs colonnes $\mathbf{a} \in \mathbb{C}^I$ et $\mathbf{b} \in \mathbb{C}^J$ donne une matrice $\mathbf{A}$ de rang 1 telle que

$$\mathbf{A} = \mathbf{a} \circ \mathbf{b} = \mathbf{a}\mathbf{b}^T. \tag{2.3}$$

En étendant ce raisonnement au produit externe de trois vecteurs $\mathbf{a} \in \mathbb{C}^I$, $\mathbf{b} \in \mathbb{C}^J$ et

$\mathbf{c} \in \mathbb{C}^K$, nous obtenons le tenseur d'ordre 3

$$\mathcal{A} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}, \tag{2.4}$$

avec

$$A_{i,j,k} = a_i b_j c_k. \tag{2.5}$$

Un tenseur pouvant s'écrire ainsi est dit de rang 1 (la notion de rang d'un tenseur sera précisée à la section 2.2.1). De la même manière, le produit externe entre Q vecteurs donnera un tenseur de rang 1 et d'ordre Q.

Le produit tensoriel n-mode par une matrice. Le produit n-mode, ou produit contracté, entre un tenseur $\mathcal{T} \in \mathbb{C}^{J_1 \times J_2 \times \cdots \times J_N}$ et une matrice $\mathbf{A} \in \mathbb{C}^{I_n \times J_n}$ est noté $\mathcal{T} \times_n \mathbf{A}$. Un élément d'un tenseur résultant d'un tel produit a pour expression

$$(\mathcal{T} \times_n \mathbf{A})_{j_1, j_2, \ldots, j_{n-1}, i_n, j_{n+1}, \ldots, j_N} = \sum_{j_n=1}^{J_n} T_{j_1, j_2, \ldots, J_N} A_{i_n, j_n}. \tag{2.6}$$

Donc $\mathcal{T} \times_n \mathbf{A} \in \mathbb{C}^{J_1 \times \cdots J_{n-1} \times I_n \times J_{n+1} \times \cdots J_N}$ est un tenseur de même ordre que le tenseur $\mathcal{T}$ dont le $n^{\text{ème}}$ mode a maintenant pour dimension le nombre de lignes de la matrice $\mathbf{A}$. On parle de produit contracté car généralement $I_n < J_n$.

Propriété : ce produit peut être appliqué successivement sur les différents modes. Le produit contracté est alors commutatif si et seulement si $m \neq n$. Soient $\mathbf{A} \in \mathbb{C}^{I_n \times J_n}$ et $\mathbf{B} \in \mathbb{C}^{I_m \times J_m}$, nous avons alors $\forall (m, n) \in [1; N]_{\mathbb{N}}, \ m \neq n$

$$(\mathcal{T} \times_n \mathbf{A}) \times_m \mathbf{B} = (\mathcal{T} \times_m \mathbf{B}) \times_n \mathbf{A} = \mathcal{T} \times_n \mathbf{A} \times_m \mathbf{B}. \tag{2.7}$$

Lien entre produit matriciel et produit tensoriel n-mode : le produit tensoriel n-mode sur un tenseur $\mathcal{T}$ peut être réécrit en fonction de sa matrice de dépliement $\mathbf{T}_{(n)}$ de la manière suivante :

$$\mathcal{T} \times_n \mathbf{A} = \mathbf{A}\mathbf{T}_{(n)}. \tag{2.8}$$

2.1.5 Décomposition de Tucker et HOSVD

Il existe de nombreuses décompositions tensorielles, nous nous intéressons ici aux décompositions multilinéaires qui peuvent être vues comme la transformation d'un tenseur en un autre tenseur du même ordre par combinaisons linéaires des vecteurs de ses différents modes [62]. Ainsi, la décomposition tensorielle la plus générale est la décomposition de Tucker [32]. Cette décomposition multilinéaire permet de décomposer tout tenseur $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_Q}$ à l'aide un tenseur cœur $\mathcal{S}$ et un ensemble de matrices $\mathbf{U}^{(1)} \ldots \mathbf{U}^{(Q)}$ de la manière suivante :

$$\mathcal{T} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_Q \mathbf{U}^{(Q)}. \tag{2.9}$$

Ainsi, si $\mathcal{S}$ est un tenseur dans $\mathbb{C}^{J_1 \times J_2 \times \ldots \times J_Q}$ alors $\mathbf{U}^{(q)} \in \mathbb{C}^{I_q \times J_q}$. Cette décomposition existe toujours pourvu que le tenseur cœur soit de dimensions suffisamment grandes. Nous nous intéressons à deux cas particuliers :

1. Les colonnes de chaque matrice $\mathbf{U}^{(q)}$ forment une base orthonormale entre elles (*i.e.* $\mathbf{U}^{(q)H}\mathbf{U}^{(q)} = \mathbf{I}_{J_q}$).

2. Le tenseur $\mathcal{S}$ est diagonal.

Le premier cas de figure est appelée HOSVD [34] (de l'anglais «High Order SVD»). Un algorithme simple de calcul de la HOSVD est le suivant : la matrice $\mathbf{U}^{(q)}$ $\forall q \in [1; Q]_{\mathbb{N}}$ est prise comme la matrice $\mathbf{U}$ de la SVD tronquée à l'ordre J_q de la matrice de dépliement $\mathbf{T}_{(q)}$, le tenseur cœur est alors calculé en inversant (2.9) :

$$\mathcal{S} = \mathcal{T} \times_1 \mathbf{U}^{(1)H} \times_2 \mathbf{U}^{(2)H} \ldots \times_Q \mathbf{U}^{(Q)H}. \tag{2.10}$$

Remarque : s'il existe au moins une valeur de q pour laquelle J_q est inférieur au rang de la matrice $\mathbf{T}_{(q)}$, la HOSVD n'est généralement pas exacte et les matrices $\mathbf{U}^{(q)}$ et le tenseur $\mathcal{S}$ ne permettent que de reconstruire une approximation de $\mathcal{T}$. Cette approximation n'est pas optimale mais elle est souvent suffisante en pratique [63]. La HOSVD est alors un outil efficace pour compresser des tenseurs.

Le deuxième cas de figure permet de définir la décomposition CP que nous allons présenter en détails dans la section suivante.

2.2 La décomposition CP

2.2.1 Définition

La décomposition CP peut donc être vue comme un cas particulier de la décomposition de Tucker où $\mathcal{S}$ est un tenseur diagonal de dimensions R. Sans perte de généralité, nous pouvons choisir de mettre à 1 les termes diagonaux de $\mathcal{S}$. En effet, cela revient à multiplier à droite l'une des matrices $\mathbf{U}^{(q)}$ par une matrice diagonale. Le tenseur $\mathcal{T}$ peut donc être factorisé à l'aide de Q matrices facteurs que nous noterons maintenant $\mathbf{F}^{(q)} \in \mathbb{C}^{I_q \times R}$ pour tout $q \in [1, Q]_{\mathbb{N}}$ de la manière suivante :

$$\mathcal{T} = \mathcal{I}_R \times_1 \mathbf{F}^{(1)} \times_2 \mathbf{F}^{(2)} \ldots \times_Q \mathbf{F}^{(Q)} \tag{2.11}$$

où $\mathcal{I}_R$ est le tenseur identité c'est-à-dire un tenseur diagonal d'ordre Q, de dimension R et ne contenant que des 1 sur sa diagonale (voir figure 2.4). Les RQ colonnes des matrices facteurs sont appelées facteurs de la décomposition.

Rang de la décomposition CP et rang du tenseur. La grandeur R ainsi définie est appelée rang de la décomposition CP. L'équation (2.11) est vraie pour plusieurs valeurs de R. La plus petite valeur de R pour laquelle (2.11) est vraie est alors appelée rang du tenseur, nous la notons $R_{\mathcal{T}}$. Cette valeur existe toujours. Autrement dit, tout tenseur admet une décomposition CP.

La distinction entre rang du tenseur et rang de la décomposition est fondamentale en pratique, nous y reviendrons dans les sections 2.2.2 et 2.3.

Les décompositions CP ont été introduites par F. L. Hitchcock dans [30] en 1927. Dans ses travaux, Hitchcock montre que la décomposition CP de rang R d'un tenseur est équivalente à la somme de R tenseurs de rang 1 à l'aide du produit externe (figure 2.5) *i.e.*

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{f}_r^{(1)} \circ \mathbf{f}_r^{(2)} \circ \ldots \circ \mathbf{f}_r^{(Q)}. \tag{2.12}$$

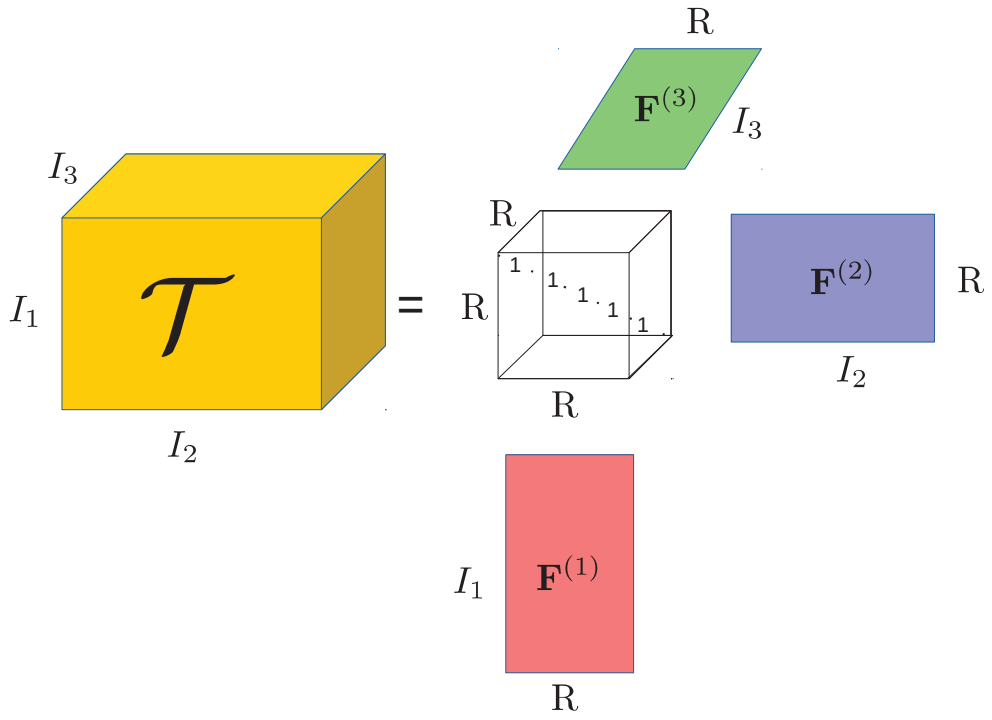

FIGURE 2.4: Schéma représentatif de la décomposition CP d'un tenseur d'ordre 3.

Finalement, un terme quelconque d'un tenseur correspondant au modèle de décomposition CP a pour expression

$$T_{i_1,i_2,\ldots,i_Q} = \sum_{r=1}^{R} F^{(1)}_{i_1,r} F^{(2)}_{i_2,r} \ldots F^{(Q)}_{i_Q,r}. \tag{2.13}$$

La décomposition CP de tenseur est un outil intéressant pour le dé-mélange de signaux. En effet, dans de nombreuses applications de séparation de sources, les colonnes des matrices facteurs contiennent une ou plusieurs estimations des signaux sources et/ou du mélange. Nous renvoyons au chapitre 6 et à l'introduction de cette thèse pour des exemples d'applications et des références bibliographiques.

2.2.2 Unicité de la décomposition CP

Notre objectif est d'estimer les matrices facteurs de la décomposition CP d'un tenseur connu. Il est donc important de savoir si ce problème a une solution unique ou pas.

Il est facile de montrer que quelque soit l'ensemble de Q matrices diagonales $\mathbf{\Lambda}^{(1)} \ldots \mathbf{\Lambda}^{(Q)}$ et de Q matrices de permutation $\mathbf{P}^{(1)} \ldots \mathbf{P}^{(Q)}$ de dimension R vérifiant $\prod_{q=1}^{Q} \mathbf{\Lambda}^{(q)} \mathbf{P}^{(q)} = \mathbf{I}_R$ alors

$$\mathcal{I}_R \times_1 \mathbf{F}^{(1)} \ldots \times_Q \mathbf{F}^{(Q)} = \mathcal{I}_R \times_1 \mathbf{F}^{(1)} \mathbf{\Lambda}^{(1)} \mathbf{P}^{(1)} \ldots \times_Q \mathbf{F}^{(Q)} \mathbf{\Lambda}^{(Q)} \mathbf{P}^{(Q)}. \tag{2.14}$$

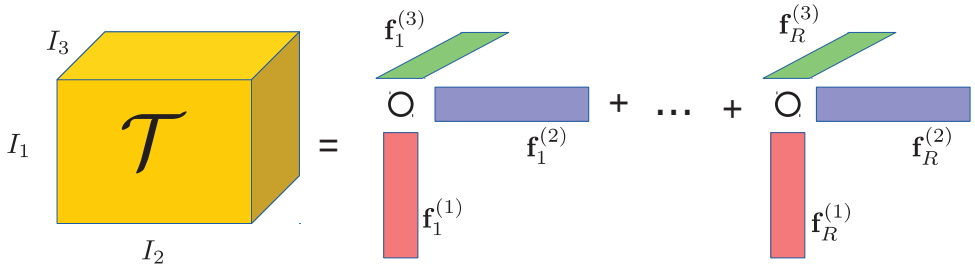

FIGURE 2.5: Schéma explicatif de la décomposition CP d'un tenseur d'ordre 3 vue comme une somme de tenseur de rang 1 (équation (2.12)).

Par conséquent, une décomposition CP de rang donné sera au mieux unique à une permutation et à une mise à l'échelle près des colonnes des matrices facteurs. On parle alors d'unicité essentielle. Nous disposons d'une condition suffisante d'unicité essentielle basée sur la notion de rang de Kruskal.

Le rang de Kruskal. Le rang de Kruskal d'une matrice $\mathbf{A}$, noté $\mathsf{rank_k}(\mathbf{A})$, est le nombre maximal k tel que chaque ensemble de k colonnes de $\mathbf{A}$ soit linéairement indépendant. Par définition, le rang de Kruskal est donc inférieur ou égal au rang d'une matrice. Cet outil mathématique a été présenté dans [64] et nommé rang de Kruskal dans [65].

Condition suffisante d'unicité d'un tenseur d'ordre 3. Dans [64] Kruskal propose une condition d'unicité suffisante pour une décomposition CP d'ordre 3 :

$$2R + 2 \leq \mathsf{rank_k}(\mathbf{F}^{(1)}) + \mathsf{rank_k}(\mathbf{F}^{(2)}) + \mathsf{rank_k}(\mathbf{F}^{(3)}). \tag{2.15}$$

Condition suffisante d'unicité d'un tenseur d'ordre quelconque. Les travaux de Kruskal ont été généralisés à l'ordre quelconque par N. D. Sidiropoulos et R. Bro dans [66], la condition suffisante d'unicité essentielle s'écrit alors

$$\sum_{q=1}^{Q} \mathsf{rank_k}(\mathbf{F}^{(q)}) \geq 2R + (Q-1). \tag{2.16}$$

Une preuve plus intuitive et facilement adaptable au cas de tenseurs à valeurs complexes a été proposée dans [67] pour la condition d'unicité essentielle de la décomposition CP.

Cette condition est vérifiée dans la plupart des applications. Par exemple, si l'on suppose que les matrices facteurs sont de rang colonne plein et supérieur à 1, la condition est automatiquement respectée pour $Q > 2$.

Dans [68], une approche différente basée sur les notions de rangs typiques et génériques fournit différents résultats et conjectures sur l'unicité des décompositions CP de tenseur

dont les éléments sont tirés aléatoirement. Ce type de tenseur est important puisque dans de nombreuses applications le tenseur de données est modélisé comme la somme d'un tenseur de rang faible contenant le signal d'intérêt et d'un tenseur de bruit aléatoire. Il en résulte un tenseur aléatoire. Il a été montré que le rang de tels tenseurs peut être largement supérieur à la plus grande des dimensions du tenseur, et que leur décomposition CP peut ne pas être essentiellement unique. La conjoncture importante ici est alors que dans ce cas là une décomposition CP de rang plus faible correspondant au rang attendu (connu ou estimé) du tenseur signal permet d'avoir unicité essentielle. On cherchera bien sûr la décomposition CP correspondant à ce rang.

2.2.3 Décomposition CP et matrices de dépliement

La décomposition CP peut se réécrire à l'aide des matrices de dépliement et du produit de Khatri-Rao. Ainsi pour un tenseur d'ordre 3, nous avons pour chaque matrice de dépliement du tenseur

$$\mathbf{T}_{(1)} = \mathbf{F}^{(1)}(\mathbf{F}^{(3)} \odot \mathbf{F}^{(2)})^T, \tag{2.17}$$

$$\mathbf{T}_{(2)} = \mathbf{F}^{(2)}(\mathbf{F}^{(1)} \odot \mathbf{F}^{(3)})^T, \tag{2.18}$$

et

$$\mathbf{T}_{(3)} = \mathbf{F}^{(3)}(\mathbf{F}^{(2)} \odot \mathbf{F}^{(1)})^T. \tag{2.19}$$

Plus généralement, pour un tenseur d'ordre supérieur à trois, la matrice $\mathbf{T}_{(q)}$ de dimensions $I_q \times \prod_{\substack{k=1 \\ k\neq q}}^{Q} I_k$ a pour expression

$$\mathbf{T}_{(q)} = \mathbf{F}^{(q)}\mathbf{K}^{(q,Q)T} \tag{2.20}$$

avec

$$\mathbf{K}^{(q,Q)} = \mathbf{F}^{(q-1)} \odot \ldots \odot \mathbf{F}^{(1)} \odot \mathbf{F}^{(Q)} \odot \ldots \odot \mathbf{F}^{(q+1)}. \tag{2.21}$$

Il est intéressant de noter que l'expression de la matrice facteurs $\mathbf{F}^{(q)}$ peut être retrouvée en pseudo-inversant la matrice $\mathbf{K}^{(q,Q)T}$. Les matrices de dépliement sont ainsi largement utilisées dans les algorithmes de décomposition CP.

Comme dit précédemment, la matrice $\mathbf{T}_{(q)}$ ne représente pas la seule manière de déplier le tenseur $\mathcal{T}$. Nous verrons par la suite qu'il peut être utile de fusionner plusieurs modes de $\mathcal{T}$ sur les lignes de la matrice de dépliement désirée. La matrice $\mathbf{F}^{(q)}$ dans (2.20) serait alors remplacée par le produit de Khatri-Rao des matrices correspondant aux modes fusionnés sur les lignes de la matrice de dépliement.

2.2.4 Décomposition CP et diagonalisation conjointe de matrices

En considérant l'équation (1.4) des propriétés du produit de Khatri-Rao, nous pouvons remarquer que les tranches frontales, verticales et horizontales d'un tenseur d'ordre trois peuvent être vues comme trois ensembles de matrices conjointement diagonalisables par deux matrices facteurs. Par exemple, pour les tranches frontales, nous avons

$$\forall i_3 \in [1; I_3]_{\mathbb{N}}\ \mathbf{T}_{\bullet\bullet i_3} = \mathbf{F}^{(1)}\mathbf{D}^{(i_3)}_{\mathbf{F}^{(3)}}\mathbf{F}^{(2)T}. \tag{2.22}$$

Pour les tranches horizontales :

$$\forall i_1 \in [1; I_1]_{\mathbb{N}} \ \mathbf{T}_{i_1 \bullet\bullet} = \mathbf{F}^{(2)} \mathbf{D}^{(i_1)}_{\mathbf{F}^{(1)}} \mathbf{F}^{(3)T}. \tag{2.23}$$

Enfin, pour les tranches latérales :

$$\forall i_2 \in [1; I_2]_{\mathbb{N}} \ \mathbf{T}_{\bullet i_2 \bullet} = \mathbf{F}^{(3)} \mathbf{D}^{(i_2)}_{\mathbf{F}^{(2)}} \mathbf{F}^{(1)T}. \tag{2.24}$$

Avec $\mathbf{D}^{(i_1)}_{\mathbf{F}^{(1)}}$, $\mathbf{D}^{(i_2)}_{\mathbf{F}^{(2)}}$ et $\mathbf{D}^{(i_3)}_{\mathbf{F}^{(3)}}$ des matrices diagonales contenant respectivement la $i_1^{\text{ème}}$, la $i_2^{\text{ème}}$ et la $i_3^{\text{ème}}$ ligne de $\mathbf{F}^{(1)}$, $\mathbf{F}^{(2)}$ et $\mathbf{F}^{(3)}$ sur leurs diagonales.

De même, en considérant l'équation (2.20) et l'équation (1.4), il est facile de voir qu'un tenseur d'ordre supérieur à 3 peut être réécrit comme un ensemble de matrices conjointement diagonalisables. En effet, une tranche $\mathbf{T}_{i_1 i_2 \ldots i_{q-2} \bullet\bullet i_{q+1} \ldots i_Q}$ aura pour expression

$$\forall i \in \left[1; \prod_{\substack{k=1 \\ k \neq q-1, k \neq q}}^{Q} I_k\right]_{\mathbb{N}} \quad \mathbf{T}_{i_1 i_2 \ldots i_{q-2} \bullet\bullet i_{q+1} \ldots i_Q} = \mathbf{F}^{(q)} \mathbf{D}^{(i)} \mathbf{F}^{(q-1)T}. \tag{2.25}$$

Avec $\mathbf{D}^{(i)}$ une matrice diagonale contenant la i-ème ligne de la matrice $\mathbf{F}^{(q-2)} \odot \ldots \odot \mathbf{F}^{(1)} \odot \mathbf{F}^{(Q)} \odot \ldots \odot \mathbf{F}^{(q+1)}$.

2.3 État de l'art des méthodes de décomposition CP

Dans cette section, nous présentons les différentes méthodes (ou algorithmes) itératives de calcul d'une décomposition CP de rang donné. Nous distinguons trois types d'algorithmes : les algorithmes de type alterné, les algorithmes de descente et les algorithmes résolvant le problème de décomposition CP après une étape de réduction de dimensions du tenseur.

Les algorithmes originaux présentés dans cette thèse permettent d'améliorer les résultats fournis par la troisième catégorie, c'est pourquoi nous lui accorderons plus d'importance ici. Cependant, il est aussi utile de rappeler ici le principe des méthodes de type alterné et de descente les plus connues, car nous serons amenés à reprendre plusieurs résultats et procédures algorithmiques établis à cette occasion.

Ces deux premiers types de méthodes formulent le problème de décomposition CP comme un problème d'optimisation classique que nous présentons dans la section suivante. Avant cela, il convient de se donner des critères pour juger de la qualité des différentes méthodes :

- Le taux de convergence de l'algorithme lié à la méthode.
- Le nombre d'itérations nécessaires pour atteindre la convergence.
- La précision d'estimation des matrices facteurs à partir d'un tenseur bruité.
- La complexité numérique de la méthode définie ici comme le nombre de multiplications effectuées à chaque itération.
- La résistance au mauvais conditionnement des matrices facteurs *i.e.* lorsque les colonnes d'une ou de plusieurs matrices facteurs sont quasi-colinéaires.

- L'insensibilité à la surestimation du rang utile de la décomposition CP (appelée communément overfactoring).

Notion de rang utile : dans la plupart des applications, le tenseur de données est modélisé comme la somme d'un tenseur de rang faible contenant le signal d'intérêt et d'un tenseur de bruit aléatoire. Par exemple dans le contexte de la séparation de sources, le rang du tenseur signal correspond au nombre de signaux sources. Il est donc évident que l'on appliquera au tenseur de données une décomposition CP de rang égal à cette valeur lorsque celle-ci est connue. On parle de rang physique ou de rang utile R_u de la décomposition CP. Cependant, le rang utile d'un tenseur est généralement inconnu en pratique. L'estimation du rang utile d'un tenseur n'est pas trivial. Des solutions ont été proposées dans [69, 70], mais elles n'en permettent pas toujours une estimation précise. Le problème est alors le suivant : Plus on augmente le rang de la décomposition, plus on minimisera l'écart quadratique entre les données et le modèle au risque d'estimer du bruit. Ainsi, on aura tendance à surestimer le rang de la décomposition par rapport au rang utile : on parle alors d'overfactoring. Dès lors deux types de comportements sont possibles lors du passage d'une décomposition de rang R_u à une décomposition de rang $R_u + r$ (avec r un entier supérieur ou égal à 1) :

1. Soit les R_u facteurs du tenseur ne changent pas et les r facteurs supplémentaires ont une contribution négligeable dans au moins un des modes. On dit alors que l'algorithme est peu sensible à l'overfactoring et l'estimation du rang utile n'est plus un problème.

2. Soit un ou plusieurs des R_u facteurs précédemment estimés sont modifiés et au moins $R_u + 1$ facteurs ont des contributions non-négligeables.

2.3.1 Formulation comme un problème d'optimisation classique

Soit $\widehat{\mathbf{F}}^{(q)}$ une estimation de la $q^{\text{ème}}$ matrice facteurs et $\widehat{\mathcal{T}}$ le tenseur reconstitué à partir des matrices facteurs estimées :

$$\widehat{\mathcal{T}} = \mathcal{I}_R \times_1 \widehat{\mathbf{F}}^{(1)} \times_2 \widehat{\mathbf{F}}^{(2)} \ldots \times_Q \widehat{\mathbf{F}}^{(Q)}. \tag{2.26}$$

Nous représentons l'écart entre le tenseur de données et sa décomposition CP de rang R à l'aide d'un tenseur $\mathcal{E}$, l'expression du tenseur à décomposer devient alors

$$\mathcal{T} = \widehat{\mathcal{T}} + \mathcal{E}. \tag{2.27}$$

Les matrices facteurs sont généralement estimées en minimisant une fonction de coût choisie comme une mesure quadratique de l'écart entre le tenseur $\mathcal{T}$ et le tenseur reconstitué $\widehat{\mathcal{T}}$:

$$\varphi(\widehat{\mathcal{T}}) = ||\mathcal{T} - \widehat{\mathcal{T}}||_F^2. \tag{2.28}$$

Notre problème est alors le suivant :

Problème 2.3.1. *Estimer les matrices facteurs $\widehat{\mathbf{F}}^{(q)}$ de la décomposition CP de rang donné du tenseur $\widehat{\mathcal{T}}$ minimisant la fonction de coût* (2.28).

On introduit alors, à l'aide des matrices de dépliements, les fonctions de coût $\varphi_{(q)}$ ne dépendant que d'une seule matrice facteurs :

$$\varphi_{(q)}(\widehat{\mathbf{F}}^{(q)}) = ||\mathbf{T}_{(q)} - \widehat{\mathbf{F}}^{(q)}(\widehat{\mathbf{K}}^{(q,Q)})^T||_F^2, \tag{2.29}$$

avec $\widehat{\mathbf{K}}^{(q,Q)}$ le produit de khatri-Rao des autres matrices facteurs estimées. La matrice $\widehat{\mathbf{F}}^{(q)}$ minimisant (2.29) est donnée par [71]

$$\widehat{\mathbf{F}}^{(q)} = \mathbf{T}_{(q)}(\widehat{\mathbf{K}}^{(q,Q)T})^{\dagger} \ \forall q \in [1;Q]_{\mathbb{N}}. \tag{2.30}$$

L'opérateur de vectorisation permet d'écrire $\varphi_{(q)}$ sous forme vectorielle [72]

$$\tilde{\varphi}_{(q)}(\mathbf{p}^{(q)}) = ||\mathbf{t}_{(q)} - \mathbf{Z}_{(q)}\mathbf{p}^{(q)}||_2^2, \tag{2.31}$$

avec

- $\mathbf{p}^{(q)} = \mathsf{vec}\{\widehat{\mathbf{F}}^{(q)}\}$ de dimensions $L_q = I_q R$,
- $\mathbf{t}_{(q)} = \mathsf{vec}\{\mathbf{T}_{(q)}\}$ de dimensions $M = \prod_{q=1}^{Q} I_q$,
- $\mathbf{Z}_{(q)} = \widehat{\mathbf{K}}^{(q,Q)} \otimes \mathbf{I}_{Iq}$ de dimensions $M \times L_q$.

Le vecteur gradient de $\tilde{\varphi}_{(q)}(\mathbf{p}_{(q)})$ a pour expression

$$\nabla_{\tilde{\varphi}_{(q)}}(\mathbf{p}^{(q)}) = 2\mathbf{Z}_{(q)}^T(-\mathbf{t}_{(q)} + \mathbf{Z}_{(q)}\mathbf{p}^{(q)}). \tag{2.32}$$

De même, nous définissons les vecteurs d'erreur

$$\forall q \in [1;Q]_{\mathbb{N}}, \ \tilde{\mathbf{e}}_{(q)}(\mathbf{p}^{(q)}) = \mathbf{t}_{(q)} - \mathbf{Z}_{(q)}\mathbf{p}^{(q)}. \tag{2.33}$$

Leurs Jacobiens sont alors donnés par

$$\mathbf{J}_{\tilde{\mathbf{e}}_{(q)}}(\mathbf{p}^{(q)}) = -\mathbf{Z}_{(q)}. \tag{2.34}$$

2.3.2 Les méthodes alternées

Ces méthodes, comme leur nom l'indique, estiment les matrices facteurs de manière alternée à chaque itération. Le principe est de considérer, à une itération donnée, toutes les matrices facteurs comme connues excepté une que l'on va estimer en fonction de toutes les autres. La structure de ces algorithmes est résumée dans la table algorithmique 1.

2.3.3 La méthode des moindres carrés alternés

La méthode des moindres carrés alternés appelée aussi ALS (de l'anglais *Alternating Least Square*) [19, 23, 33] est la méthode la plus célèbre pour résoudre le problème 2.3.1. Lors d'une itération, la fonction de coût (2.29) est minimisée Q fois en fonction de chaque matrice facteurs $\widehat{\mathbf{F}}^{(q)}$. Ainsi, pour un tenseur d'ordre Q, la mise à jour des Q matrices facteurs s'effectue comme

$$\widehat{\mathbf{F}}^{(q)}_{(it+1)} \leftarrow \mathbf{T}_{(q)}(\widehat{\mathbf{K}}^{(q,Q)T}_{(it)})^{\dagger} \ \forall q \in [1;Q]_{\mathbb{N}} \tag{2.35}$$

Algorithme 1 Forme générique des algorithmes de type alterné

Soit S_C un critère d'arrêt préalablement défini et It_{max} le nombre d'itérations maximal ;

Initialisation des matrices facteurs $\widehat{\mathbf{F}}^{(2)}$ à $\widehat{\mathbf{F}}^{(Q)}$;
$it \leftarrow 1$;
while S_C est $faux$ et $it < It_{max}$ **do**
  for $q = 1$ à Q **do**
    Estimation des paramètres de la matrice facteurs $\widehat{\mathbf{F}}^{(q)}_{(it+1)}$ en fonction des autres matrices facteurs précédemment estimées ;
  end for
  Mettre à jour S_C ;
  $it \leftarrow it + 1$;
end while

avec $\widehat{\mathbf{K}}^{(q,Q)}_{(it)} = \widehat{\mathbf{F}}^{(q-1)}_{(it+1)} \odot \ldots \odot \widehat{\mathbf{F}}^{(1)}_{(it+1)} \odot \widehat{\mathbf{F}}^{(Q)}_{(it)} \odot \ldots \odot \widehat{\mathbf{F}}^{(q+1)}_{(it)}$. Il s'agit alors de résoudre successivement Q problèmes des moindres carrés dont la solution est unique si et seulement si les matrices $\widehat{\mathbf{K}}^{(q,Q)}$ sont de rang colonnes plein.

Par exemple, pour un tenseur d'ordre 3, lors d'une itération it les matrices facteurs sont mises à jour de la manière suivante :

$$\widehat{\mathbf{F}}^{(1)}_{(it+1)} \leftarrow \mathbf{T}_{(1)}\big((\widehat{\mathbf{F}}^{(3)}_{(it)} \odot \widehat{\mathbf{F}}^{(2)}_{(it)})^T\big)^\dagger \tag{2.36}$$

$$\widehat{\mathbf{F}}^{(2)}_{(it+1)} \leftarrow \mathbf{T}_{(2)}\big((\widehat{\mathbf{F}}^{(1)}_{(it+1)} \odot \widehat{\mathbf{F}}^{(3)}_{(it)})^T\big)^\dagger \tag{2.37}$$

$$\widehat{\mathbf{F}}^{(3)}_{(it+1)} \leftarrow \mathbf{T}_{(3)}\big((\widehat{\mathbf{F}}^{(2)}_{(it+1)} \odot \widehat{\mathbf{F}}^{(1)}_{(it+1)})^T\big)^\dagger. \tag{2.38}$$

L'ALS présente plusieurs avantages. Il est simple à implémenter, il garantit la convergence vers un minimum au moins local et il est toujours possible de calculer chacune des matrices facteurs.

L'ALS présente aussi des inconvénients. Il est très sensible à l'initialisation des matrices facteurs. Il est également peu efficace lorsqu'une ou plusieurs matrices facteurs ont des colonnes colinéaires ou quasi-colinéaires ou lorsque le rang du tenseur est surestimé [22,73–76]. De plus, l'ALS est un algorithme plutôt lent à converger dans sa version de originale. En effet, la fonction de coût (2.29) décroît de manière quasi-linéaire au fil des itérations lorsque qu'elle est minimisée par l'ALS alors que d'autres méthodes permettent un taux de convergence super-linaire voir quadratique. Dans le but d'accélérer la convergence de l'ALS, une méthode appelée ELS (de l'anglais «Enhanced Line Search») a été proposée dans [77] pour des tenseurs à valeurs réelles et dans [16] pour des tenseurs à valeurs complexes. L'ELS consiste à faire une recherche linéaire optimale des matrices facteurs à partir de leur précédente estimation. À chaque itération nous avons donc une nouvelle expression pour chaque matrice facteurs :

$$\widehat{\mathbf{F}}^{(q)}_{(nouveau)} = \widehat{\mathbf{F}}^{(q)}_{(it)} + \eta_{\widehat{\mathbf{F}}^{(q)}}(\widehat{\mathbf{F}}^{(q)}_{(it+1)} - \widehat{\mathbf{F}}^{(q)}_{(it)}) \ \forall q \in [1;Q]_{\mathbb{N}} \tag{2.39}$$

où $\eta_{\widehat{\mathbf{F}}^{(q)}} \ \forall q \in [1;Q]$ sont les pas de direction de recherche linéaire. Ces derniers sont calculés, à chaque nouvelle estimation de la matrice $\mathbf{F}^{(q)}$, de manière optimale en cherchant les racines du polynôme $\varphi_{(q)}(\widehat{\mathbf{F}}^{(q)}_{(nouveau)})$. Bien que l'ELS augmente le coût de calcul par

itération de l'ALS, il permet de réduire significativement le nombre d'itérations nécessaires pour atteindre la convergence. L'ELS permet aussi à l'ALS de converger lorsque les dimensions du tenseur ne sont pas suffisamment grandes devant son rang [78].

2.3.4 ALS non-négatif

L'ALS a été adapté de plusieurs manières au problème de décomposition CP non-négative pour des tenseurs d'ordre 3. L'adaptation la plus naturelle, présentée dans [42], est la projection de chacun des termes des matrices facteurs dans $\mathbb{R}^+$. Ainsi, directement après l'estimation de chaque matrice facteurs cette dernière est projetée de la manière suivante :

$$\widehat{\mathbf{F}}^{(q)}_{(it+1)} \leftarrow \left[\widehat{\mathbf{F}}^{(q)}_{(it+1)}\right]_+. \tag{2.40}$$

La plupart du temps, les signaux représentés par une décomposition CP non-négative présentent d'autres propriétés particulières. La fonction de coût (2.29) peut alors être modifiée pour en tenir compte. Selon les cas, les signaux à séparer peuvent être de type lisse ou parcimonieux. La fonction de coût peut alors être pénalisée de deux manières :

$$\varphi(\widehat{\mathcal{T}}) + \sum_{q=1}^{Q} \alpha_q ||\widehat{\mathbf{F}}^{(q)}||_F^2 \tag{2.41}$$

pour favoriser une solution lisse, ou alors

$$\varphi(\widehat{\mathcal{T}}) + \sum_{q=1}^{Q} \alpha_q ||\widehat{\mathbf{F}}^{(q)}||_1 \tag{2.42}$$

lorsque la solution recherchée présente des propriétés de parcimonie. Avec α_q des scalaires de $\mathbb{R}^+$ appelés paramètres de régularisation. Ces paramètres de régularisation sont des paramètres heuristiques, ils peuvent ainsi être fixes au cours des itérations ou bien être mis à jour de différentes manières. Bien entendu, ces pénalisations peuvent être aussi utilisées pour l'ALS sans contrainte de non-négativité. L'expression des matrices facteurs est obtenue en minimisant une des deux fonctions de coût précédentes alternativement comme dans la procédure de l'ALS classique.

2.3.5 La Méthode des directions alternées

La méthode des directions alternées appelée aussi ADMM (pour *Alternating Direction Method of Multipliers*) [79] est une méthode utilisée pour résoudre les problèmes d'optimisation sous contrainte de la forme :

$$\begin{aligned} &\underset{\mathbf{x},\mathbf{z}}{\text{minimiser}}\ f(\mathbf{x}) + g(\mathbf{z}) \\ &\text{contraint à } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \end{aligned} \tag{2.43}$$

avec $\mathbf{x} \in \mathbb{R}^r$, $\mathbf{z} \in \mathbb{R}^s$, $\mathbf{A} \in \mathbb{R}^{p\times r}$, $\mathbf{B} \in \mathbb{R}^{p\times s}$ et $\mathbf{c} \in \mathbb{R}^p$, $f : \mathbb{R}^r \to \mathbb{R}$ et $g : \mathbb{R}^s \to \mathbb{R}$.

Cette méthode présente de bonnes propriétés de convergence ainsi qu'une preuve dans le cas convexe. Le principe est de minimiser le Lagrangien augmenté du problème (2.43) :

$$\mathrm{L}_\rho\big(\mathbf{x},\mathbf{z},\mathbf{y}\big) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) \ + (\rho/2)||\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}||_F^2 \tag{2.44}$$

avec $\rho > 0$ un paramètre de pénalité et $\mathbf{y} \in \mathbb{R}^p$ le multiplicateur de Lagrange. De la même manière que pour l'ALS, le critère (2.44) va être minimisé alternativement par rapport aux variables $\mathbf{x}$ et $\mathbf{z}$. Ainsi une itération d'ADMM est définie de la manière suivante :

$$\begin{aligned}
\mathbf{x}_{(it+1)} &\leftarrow \arg\min_{\mathbf{x}} \mathrm{L}_\rho\big(\mathbf{x}, \mathbf{z}_{(it)}, \mathbf{y}_{(it)}\big) && (2.45)\\
\mathbf{z}_{(it+1)} &\leftarrow \arg\min_{\mathbf{z}} \mathrm{L}_\rho\big(\mathbf{x}_{(it+1)}, \mathbf{z}, \mathbf{y}_{(it)}\big) && (2.46)\\
\mathbf{y}_{(it+1)} &\leftarrow \mathbf{y}_{(it)} + \rho\big(\mathbf{A}\mathbf{x}_{(it+1)} + \mathbf{B}\mathbf{z}_{(it+1)} - \mathbf{c}\big). && (2.47)
\end{aligned}$$

L'ADMM a été adaptée aux décompositions CP non-négatives dans [80] pour des tenseurs d'ordre trois. Nous proposons ici de l'étendre à l'ordre quelconque.

Pour se faire, une matrice auxiliaire $\tilde{\mathbf{F}}^{(q)}$ est introduite pour chaque matrice facteur (avec $q \in [1; Q]_{\mathbb{N}}$). Ces matrices auxiliaires permettent de définir les fonctions de coût suivantes :

$$\forall\ q \in [1; Q]_{\mathbb{N}},\ \varphi^{(q)}_{\mathrm{ADMM}}\big(\widehat{\mathbf{F}}^{(q)}, \tilde{\mathbf{F}}^{(q)}\big) = \frac{1}{2}\varphi_{(q)}\big(\widehat{\mathbf{F}}^{(q)}\big) + \sum_{j=1}^{Q} i(\tilde{\mathbf{F}}^{(j)}) \tag{2.48}$$

avec $i(\mathbf{M})$ une fonction indicatrice définie par

$$i(\mathbf{M}) = \begin{cases} 0 \text{ si } \quad \mathbf{M} \succeq 0 \\ +\infty \text{ sinon} \end{cases} \tag{2.49}$$

Le problème d'optimisation à résoudre à chaque itération devient alors

$$\begin{aligned}
\forall\ q \in [1; Q]_{\mathbb{N}},\ &\underset{\widehat{\mathbf{F}}^{(q)}, \tilde{\mathbf{F}}^{(q)}}{\text{minimiser}}\ \varphi^{(q)}_{\mathrm{ADMM}}\big(\widehat{\mathbf{F}}^{(q)}, \tilde{\mathbf{F}}^{(q)}\big)\\
&\text{contraint à } \widehat{\mathbf{F}}^{(q)} = \tilde{\mathbf{F}}^{(q)}. && (2.50)
\end{aligned}$$

Le Lagrangien augmenté d'un tel problème s'écrit

$$\begin{aligned}
\mathrm{L}_\rho(\widehat{\mathbf{F}}^{(q)}, \tilde{\mathbf{F}}^{(q)}, \mathbf{\Lambda}^{(q)}) &= \varphi^{(q)}_{\mathrm{ADMM}}(\widehat{\mathbf{F}}^{(q)}, \tilde{\mathbf{F}}^{(q)}) + \sum_{j=1}^{Q} \big(\mathsf{trace}\{\mathbf{\Lambda}^{(j)T}(\widehat{\mathbf{F}}^{(j)} - \tilde{\mathbf{F}}^{(j)})\}\\
&+ \frac{\rho^{(j)}}{2}||\widehat{\mathbf{F}}^{(j)} - \tilde{\mathbf{F}}^{(j)}||^2_F\big), && (2.51)
\end{aligned}$$

où les $\mathbf{\Lambda}^{(q)}$ sont les multiplicateurs de Lagrange.

Le but est donc maintenant de minimiser le Lagrangien augmenté (2.51) comme indiqué précédemment. Il est montré dans [81] que l'opérateur proximal qui minimise la fonction $i(x)$ est la projection de x sur $\mathbb{R}^+$. Ainsi, les matrices facteurs, les matrices auxiliaires et les multiplicateurs de Lagrange à estimer sont mis à jour de la manière suivante :

$$\begin{aligned}
\widehat{\mathbf{F}}^{(q)}_{(it+1)} &\leftarrow \big(\mathbf{T}_{(q)}\,\widehat{\mathbf{K}}^{(q,Q)}_{(it)} + \rho^{(q)}\,\tilde{\mathbf{F}}^{(q)}_{(it)} - \mathbf{\Lambda}^{(q)}_{(it)}\big)\big(\widehat{\mathbf{K}}^{(q,Q)T}_{(it)}\,\widehat{\mathbf{K}}^{(q,Q)}_{(it)} + \rho^{(q)}\mathbf{I}_R\big)^{-1}\\
\tilde{\mathbf{F}}^{(q)}_{(it+1)} &\leftarrow \big[\big(\widehat{\mathbf{F}}^{(q)}_{(it+1)} + \frac{1}{\rho^{(q)}}\mathbf{\Lambda}^{(q)}_{(it)}\big)\big]_+\\
\mathbf{\Lambda}^{(q)}_{(it+1)} &\leftarrow \mathbf{\Lambda}^{(q)}_{(it)} + \rho^{(q)}(\widehat{\mathbf{F}}^{(q)}_{(it+1)} - \tilde{\mathbf{F}}^{(q)}_{(it+1)}). && (2.52)
\end{aligned}$$

Cette méthode, nommée ADMoM dans [80] peut être facilement implémentée pour du calcul parallèle de décomposition CP de tenseur de grandes dimensions et diminue les temps de calcul en comparaison avec l'ALS pour des tenseurs de rang faible.

Définissons maintenant :

$$\mathbf{P}^{(q)}_{(it+1)} = \widehat{\mathbf{F}}^{(q)}_{(it+1)} - \tilde{\mathbf{F}}^{(q)}_{(it+1)} \tag{2.53}$$

et

$$\mathbf{Y}^{(q)}_{(it+1)} = \rho^{(q)}_{(it)}(\tilde{\mathbf{F}}^{(q)}_{(it+1)} - \tilde{\mathbf{F}}^{(q)}_{(it)}) \tag{2.54}$$

Dans [79], Il est conseillé de mettre à jour les paramètres de pénalité $\rho^{(q)}$ de la manière suivante :

$$\rho^{(q)}_{(it+1)} = \begin{cases} \rho^{(q)}_{(it)}\tau^{(1)} \text{ si } ||\mathbf{P}^{(q)}_{(it+1)}||_F > \mu||\mathbf{Y}^{(q)}_{(it+1)}||_F \\ \rho^{(q)}_{(it)}/\tau^{(2)} \text{ si } ||\mathbf{Y}^{(q)}_{(it+1)}||_F > \mu||\mathbf{P}^{(q)}_{(it+1)}||_F \\ \rho^{(q)}_{(it)} \text{ sinon,} \end{cases}$$

avec $\mu > 1$, $\tau^{(1)} > 1$ et $\tau^{(2)} > 1$. L'idée est ici d'essayer de conserver un rapport μ entre la norme de $\mathbf{Y}^{(q)}_{(it+1)}$ et de $\mathbf{P}^{(q)}_{(it+1)}$ de telle sorte que ces deux quantités convergent toutes les deux vers zéro. En effet, en observant le Lagrangien augmenté (2.51), nous pouvons remarquer que plus $\rho^{(q)}_{(it+1)}$ est grand plus $||\mathbf{P}^{(q)}_{(it+1)}||_F$ sera petit et plus $\rho^{(q)}_{(it+1)}$ est petit plus $||\mathbf{Y}^{(q)}_{(it+1)}||_F$ sera petit (voir équation (2.54)).

De plus, un critère d'arrêt est aussi proposé lorsque les normes des matrices $\mathbf{P}^{(q)}_{(it+1)}$ et $\mathbf{Y}^{(q)}_{(it+1)}$ sont suffisamment petites. Ainsi ADMoM est stoppé si : $\forall q \in [1; Q]_{\mathbb{N}}$,

$$\begin{cases} ||\mathbf{P}^{(q)}_{(it+1)}||_F \leq \sqrt{I_q R}\epsilon^{abs} + \epsilon^{rel} max\{||\widehat{\mathbf{F}}^{(q)}_{(it+1)}||_F, ||\tilde{\mathbf{F}}^{(q)}_{(it+1)}||_F\} \\ ||\mathbf{Y}^{(q)}_{(it+1)}||_F \leq \sqrt{I_q R}\epsilon^{abs} + \epsilon^{rel}||\mathbf{\Lambda}^{(q)}_{(it+1)}||_F, \end{cases}$$

avec $\epsilon^{abs} = \epsilon^{rel} = 10^{-4}$.

2.3.6 Algorithmes de descente

Principe général. La décomposition CP peut également être calculée à l'aide des algorithmes de descente classiques tels que le gradient, le gradient conjugué, Gauss-Newton, Levenberg-Marquardt ou encore BFGS pour ne citer que les principaux. Ces méthodes étant bien connues, nous ne rappelons ici que le principe général des algorithmes de descente dans le cas réel ainsi que la partie spécifique à la décomposition CP c'est-à-dire le calcul du gradient et celui du Jacobien. En effet, ceux-ci peuvent se calculer analytiquement ce qui rend la plupart de ces méthodes efficaces en pratique. Nous renvoyons à la lecture de [28, 72, 78, 82] pour les détails d'implémentation et des comparaisons avec les méthodes alternées.

En comparaison avec les algorithmes alternés, toutes les variables sont ici estimées simultanément à chaque itération. Nous définissons :

- Le vecteur paramètre

$$\mathbf{p} = \begin{pmatrix} \mathbf{p}^{(1)} \\ \mathbf{p}^{(2)} \\ \vdots \\ \mathbf{p}^{(Q)} \end{pmatrix} \tag{2.55}$$

où les vecteurs $\mathbf{p}^{(q)}$ sont définis comme dans la section 2.3.1 (le vecteur $\mathbf{p}$ est donc de dimension $L = R \sum_{q=1}^{Q} I_q$),

- $\widehat{\mathcal{T}}(\mathbf{p})$ le tenseur reconstruit à partir des matrices facteurs estimées $\widehat{\mathbf{F}}^{(q)}$ $\forall q \in [1; Q]_{\mathbb{N}}$,
- $\mathbf{t}_{(q)} = \mathsf{vec}\{\mathbf{T}_{(q)}\}$ de dimensions $M = \prod_{q=1}^{Q} I_q$,
- $\widehat{\mathbf{t}}_{(q)}(\mathbf{p}) = \mathsf{vec}\{\widehat{\mathbf{T}}(\mathbf{p})_{(q)}\}$ de dimensions $M = \prod_{q=1}^{Q} I_q$.

L'écart quadratique à minimiser (2.28) peut alors s'écrire :

$$\begin{aligned} \varphi(\mathbf{p}) &= ||\mathbf{t}_{(q)} - \widehat{\mathbf{t}}_{(q)}(\mathbf{p})||_2^2 \\ &= \mathbf{e}_{(q)}(\mathbf{p})^T \mathbf{e}_{(q)}(\mathbf{p}), \end{aligned} \tag{2.56}$$

avec $\mathbf{e}_{(q)}(\mathbf{p}) = \mathbf{t}_{(q)} - \widehat{\mathbf{t}}_{(q)}(\mathbf{p})$.

Le principe des algorithmes de descente est de minimiser (2.56) en choisissant une direction $\mathbf{d} \in \mathbb{R}^L$ et un pas $\mu \in \mathbb{R}$ à partir du point courant $\mathbf{p}_{(it)}$. Par conséquent, la règle de mise à jour des algorithmes de descente est

$$\mathbf{p}_{(it+1)} = \mathbf{p}_{(it)} + \mu_{(it)} \mathbf{d}_{(it)}. \tag{2.57}$$

Il convient donc à chaque itération de trouver $\mathbf{d}$ et μ tel que

$$\varphi_{(q)}(\mathbf{p}_{(it+1)}) < \varphi_{(q)}(\mathbf{p}_{(it)}). \tag{2.58}$$

Le choix de $\mathbf{d}$ et μ dépend de l'algorithme utilisé. Plus précisément, le calcul de $\mathbf{d}$ est basé sur le gradient de $\varphi(\mathbf{p})$ noté $\mathbf{g}(\mathbf{p})$ et le Jacobien de $\mathbf{e}_{(q)}(\mathbf{p})$ noté $\mathbf{J}_{\mathbf{e}}(\mathbf{p})$ dont nous donnons ici les expressions :

$$\mathbf{g}(\mathbf{p}) = \begin{pmatrix} \nabla_{\tilde{\varphi}_{(1)}}(\mathbf{p}^{(1)}) \\ \nabla_{\tilde{\varphi}_{(2)}}(\mathbf{p}^{(2)}) \\ \vdots \\ \nabla_{\tilde{\varphi}_{(Q)}}(\mathbf{p}^{(Q)}) \end{pmatrix} = \begin{pmatrix} 2\mathbf{Z}_{(1)}^T(-\mathbf{t}_{(1)} + \mathbf{Z}_{(1)}\mathbf{p}^{(1)}) \\ 2\mathbf{Z}_{(2)}^T(-\mathbf{t}_{(2)} + \mathbf{Z}_{(2)}\mathbf{p}^{(2)}) \\ \vdots \\ 2\mathbf{Z}_{(Q)}^T(-\mathbf{t}_{(Q)} + \mathbf{Z}_{(Q)}\mathbf{p}^{(Q)}), \end{pmatrix} \tag{2.59}$$

$$\begin{aligned} \mathbf{J}_{\mathbf{e}}(\mathbf{p}) &= \left(\mathbf{J}_{\tilde{\mathbf{e}}_{(1)}}(\mathbf{p}^{(1)}) \quad \mathbf{\Pi}^{(1)}\mathbf{J}_{\tilde{\mathbf{e}}_{(2)}}(\mathbf{p}^{(2)}) \quad \dots \quad \mathbf{\Pi}^{(Q-1)}\mathbf{J}_{\tilde{\mathbf{e}}_{(Q)}}(\mathbf{p}^{(Q)})\right) \\ &= \left(-\mathbf{Z}_{(1)} \quad -\mathbf{\Pi}^{(1)}\mathbf{Z}_{(2)} \quad \dots \quad -\mathbf{\Pi}^{(Q-1)}\mathbf{Z}_{(Q)}\right), \end{aligned} \tag{2.60}$$

où les matrices $\mathbf{\Pi}^{(1)}, \ldots, \mathbf{\Pi}^{(Q-1)}$ sont des matrices de permutation chargées de remettre les lignes de $\mathbf{J}_{\tilde{\mathbf{e}}_{(2)}}(\mathbf{p}^{(2)}) \ldots \mathbf{J}_{\tilde{\mathbf{e}}_{(Q)}}(\mathbf{p}^{(Q)})$ dans le même ordre que celles de $\mathbf{J}_{\tilde{\mathbf{e}}_{(1)}}(\mathbf{p}^{(1)})$.

On en déduit par exemple la direction de descente pour l'algorithme du gradient :

$$\mathbf{d}_{\text{grad}} = -\mathbf{g}(\mathbf{p}) \tag{2.61}$$

ou pour l'algorithme de Levenberg-Marquardt :

$$\mathbf{d}_{\text{LM}} = -[\mathbf{J}_{\mathbf{e}}^T(\mathbf{p})\mathbf{J}_{\mathbf{e}}(\mathbf{p}) + \lambda \mathbf{I}]^{-1}\mathbf{g}(\mathbf{p}). \tag{2.62}$$

Algorithmes de descente non-négatifs. Tout comme pour les méthodes alternées, il est possible de contraindre les résultats fournis par les méthodes de descente à être non-négatifs de manière triviale par projection dans $\mathbb{R}^+$ à chaque itération [83]. Toutefois, une autre manière de procéder à été proposée dans [84,85]. Cette dernière consiste à minimiser les fonctions de coût (2.42) ou (2.41) à l'aide de différents algorithmes de descente tout en paramétrisant les matrices facteurs de la manière suivante :

$$\forall q \in [1; Q]_{\mathbb{N}},\ \widehat{\mathbf{F}}^{(q)} = \widehat{\mathbf{G}}^{(q)} \boxdot \widehat{\mathbf{G}}^{(q)} \tag{2.63}$$

Les fonctions de coût (2.41) ou (2.42) sont alors minimisées par rapport aux matrices $\widehat{\mathbf{G}}^{(q)}$. Chaque terme des matrices $\widehat{\mathbf{F}}^{(q)}$ étant le carré du terme de $\widehat{\mathbf{G}}^{(q)}$ correspondant, $\widehat{\mathbf{F}}^{(q)}$ est alors non-négative par construction. Cette méthode est plutôt lente à converger pour des tenseurs de grandes dimensions. En revanche, elle fait preuve d'une grande robustesse lorsque le nombre de facteurs est surestimé. Dans [86], une paramétrisation exponentielle est utilisée :

$$\forall q \in [1; Q]_{\mathbb{N}},\ \widehat{\mathbf{F}}^{(q)} = \exp(\widehat{\mathbf{G}}^{(q)}). \tag{2.64}$$

2.3.7 Algorithmes de réduction de dimensions

Le principe de ce type de méthode consiste à réduire une ou plusieurs des dimensions du tenseur initial. L'intérêt est principalement de réduire la complexité numérique du processus itératif et généralement d'en faciliter la convergence. Les méthodes que nous allons présenter ici sont basées sur une HOSVD ou une SVD tronquée au rang R. En pratique, si R est inconnu, seules les valeurs singulières significatives sont conservées. Les algorithmes sont présentés ici dans le cas d'une décomposition CP exacte. Ils restent bien sûr applicables tels quels pour le calcul d'une décomposition CP approximée.

2.3.7.1 Compression par HOSVD

On suppose ici que R est plus petit que les dimensions du tenseur. En appliquant une HOSVD de rang R au tenseur $\mathcal{T}$, nous obtenons la relation suivante :

$$\mathcal{T} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_Q \mathbf{U}^{(Q)} = \mathcal{I}_R \times_1 \mathbf{F}^{(1)} \times_2 \mathbf{F}^{(2)} \ldots \times_Q \mathbf{F}^{(Q)}. \tag{2.65}$$

Nous rappelons que les colonnes des matrices $\mathbf{U}^{(q)} \in \mathbb{C}^{I_q \times R}$ forment une base orthonormale de $\mathbb{C}^R$ et que $\mathcal{S} \in \mathbb{C}^{R \times R \times \ldots \times R}$. La décomposition CP du tenseur $\mathcal{S}$ s'écrit alors

$$\mathcal{S} = \mathcal{I}_R \times_1 \mathbf{H}^{(1)} \times_2 \mathbf{H}^{(2)} \ldots \times_Q \mathbf{H}^{(Q)}, \tag{2.66}$$

avec $\mathbf{H}^{(q)} = \mathbf{U}^{(q)H}\mathbf{F}^{(q)} \quad \forall q \in [1;Q]_{\mathbb{N}}$.

Les matrices $\mathbf{H}^{(q)}$ peuvent alors être estimées en utilisant n'importe quelle méthode de décomposition CP appliquée au tenseur $\mathcal{S}$. On en déduit alors directement les matrices facteurs $\mathbf{F}^{(q)}$. Toutefois, nous verrons ultérieurement qu'une autre stratégie est possible.

Compression par HOSVD sous contraintes de non-négativité. Effectuer une décomposition CP non-négative à l'aide d'une compression par HOSVD consiste à résoudre le problème d'optimisation suivant :

$$\begin{aligned} &\underset{\widehat{\mathbf{H}}^{(1)}\ldots\widehat{\mathbf{H}}^{(Q)}}{\text{minimiser}}, ||\mathcal{S} - \widehat{\mathcal{S}}||_F^2 \\ &\text{contraint à } \forall\ q \in [1;Q]_{\mathbb{N}},\ \mathbf{U}^{(q)}\widehat{\mathbf{H}}^{(q)} \succeq 0. \end{aligned} \tag{2.67}$$

Dans [87, 88], plusieurs méthodes ont été proposées pour résoudre ce problème. Comme par exemple l'ALS projetée (PROCO-ALS pour *PROjected COmpressed ALS*), l'ADMM (COADMM pour *COmpressed ADMM*) ou encore le gradient conjugué. Par exemple pour PROCO-ALS, il convient tout d'abord d'estimer $\widehat{\mathbf{H}}^{(q)}_{(it+1)}$, puis de calculer $\widehat{\mathbf{F}}^{(q)}_{(it+1)} = \mathbf{U}^{(q)}\widehat{\mathbf{H}}^{(q)}_{(it+1)}$ (étape de décompression), l'étape de projection est alors

$$\widehat{\mathbf{F}}^{(q)}_{(it+1)} \leftarrow \left[\widehat{\mathbf{F}}^{(q)}_{(it+1)}\right]_+ \tag{2.68}$$

et enfin de calculer $\widehat{\mathbf{H}}^{(q)}_{(it+1)} = \mathbf{U}^{(q)H}\widehat{\mathbf{F}}^{(q)}_{(it+1)}$ (étape de recompression).

Compression par HOSVD et Diagonalisation Conjointe par Similitude. Cette méthode a été proposée dans [39,89] sous le nom de SECSI pour *SEmi-algebraic framework for approximate CP decompositions via SImultaneous matrix diagonalization*. Considérant (2.65) et (2.66), nous pouvons réécrire

$$\mathcal{S} \times_3 \mathbf{U}^{(3)} \ldots \times_Q \mathbf{U}^{(Q)} = \mathcal{I}_R \times_1 \mathbf{H}^{(1)} \times_2 \mathbf{H}^{(2)} \times_3 \mathbf{F}^{(3)} \ldots \times_Q \mathbf{F}^{(Q)}. \tag{2.69}$$

Posons maintenant $\mathcal{P} = \mathcal{S} \times_3 \mathbf{U}^{(3)} \ldots \times_Q \mathbf{U}^{(Q)}$. Nous pouvons noter que $\mathcal{P}$ est un tenseur correspondant au modèle de décomposition CP. Une tranche dans le premier mode d'un tel tenseur s'écrit alors

$$\mathbf{P}_{\bullet\bullet i_3\ldots i_Q} = \mathbf{H}^{(1)}\mathbf{\Lambda}^{(i)}\mathbf{H}^{(2)T}\ \forall i \in \left[1; \prod_{q=3}^{Q} I_q\right]_{\mathbb{N}}, \tag{2.70}$$

où la matrice $\mathbf{\Lambda}^{(i)}$ est une matrice diagonale contenant sur sa diagonale la i-ème ligne de la matrice $\mathbf{F}^{(Q)} \odot \ldots \odot \mathbf{F}^{(3)}$. Ainsi en multipliant une tranche $\mathbf{P}_{\bullet\bullet i_3\ldots i_Q}$, avec la pseudo-inverse d'une tranche $\mathbf{P}_{\bullet\bullet i'_3\ldots i'_Q}$, nous obtenons :

$$\forall (i_3, ..., i_Q) \neq (i'_3, ..., i'_Q)\ \mathbf{P}_{\bullet\bullet i_3\ldots i_Q}\mathbf{P}^{\dagger}_{\bullet\bullet i'_3\ldots i'_Q} = \mathbf{H}^{(1)}\mathbf{D}^{(j)}\mathbf{H}^{(1)-1}, \tag{2.71}$$

où la matrice $\mathbf{D}^{(j)} = \mathbf{\Lambda}^{(i)}\mathbf{\Lambda}^{(i')\dagger}$. La matrice $\mathbf{D}^{(j)}$ contient donc la division terme à terme de la i-ème ligne de la matrice $\mathbf{F}^{(Q)} \odot \ldots \odot \mathbf{F}^{(3)}$ par la i'-ème ligne de la matrice $\mathbf{F}^{(Q)} \odot \ldots \odot \mathbf{F}^{(3)}$ sauf si un élément de la i'-ème ligne de la matrice $\mathbf{F}^{(Q)} \odot \ldots \odot \mathbf{F}^{(3)}$ est égal à zéro (l'élément

correspondant de la matrice $\mathbf{D}^{(j)}$ est alors égal à zéro). Cet ensemble d'équations constitue un problème de Diagonalisation Conjointe par Similitude (DCS) qui sera abondement traité dans la section 3.3 du chapitre suivant. En répétant (2.71) pour tous les indices $i_3, \ldots, i_Q$ et $i'_3, \ldots, i'_Q$, nous obtenons un ensemble de $\prod_{q=3}^{Q} I_q(\prod_{q=3}^{Q} I_q - 1)$ matrices à diagonaliser.

Une fois ce problème de DCS résolu, nous disposons des estimées de $\mathbf{H}^{(1)}$ et des matrices $\mathbf{D}^{(j)}$, respectivement notées $\widehat{\mathbf{H}}^{(1)}$ et $\widehat{\mathbf{D}}^{(i)}$. La matrice $\mathbf{F}^{(1)}$ est alors estimée à partir de $\widehat{\mathbf{F}}^{(1)} = \mathbf{U}^{(1)}\widehat{\mathbf{H}}^{(1)}$. Les matrices $\mathbf{F}^{(3)}, \ldots, \mathbf{F}^{(Q)}$ sont estimées à partir des matrices $\widehat{\mathbf{D}}^{(i)}$ et la matrice $\mathbf{F}^{(2)}$ par pseudo-inverse (de la même manière que dans l'équation (2.30)).

On peut construire un second problème de DCS à partir des matrices $\mathbf{P}^{\dagger}_{\bullet\bullet i_3 \ldots i_Q}\mathbf{P}_{\bullet\bullet i'_3 \ldots i'_Q}$. La résolution de ce problème fournit de la même manière une seconde estimation de l'ensemble des matrices facteurs. Enfin, le même schéma peut être reconduit à partir de n'importe quelle permutation des modes du tenseur conduisant à chaque fois à deux nouvelles estimées pour chaque matrice facteurs.

Les auteurs proposent alors différentes heuristiques pour choisir les estimées des Q matrices facteurs parmi toutes les combinaisons possibles. Cette méthode a plusieurs avantages : elle est particulièrement efficace lorsqu'une ou plusieurs des matrices facteurs sont mal-conditionnées et est peu sensible au problème d'overfactoring. En outre, elle permet d'initialiser efficacement les méthodes alternées et de descente. En contre partie, le calcul de toutes les estimées et leur choix peut être inutilement coûteux et l'algorithme a une condition nécessaire de fonctionnement :

$$\exists\ (q_1, q_2) \in [1; Q]^2_{\mathbb{N}},\ q_1 \neq q_2 \text{ tels que } I_{q_1} \geq R \text{ et } I_{q_2} \geq R. \tag{2.72}$$

2.3.7.2 Compression par SVD

Le principe est ici d'appliquer une SVD de rang R sur une matrice de dépliement du tenseur, ainsi nous avons :

$$\mathbf{T}_{(q)} = \mathbf{F}^{(q)}\mathbf{K}^{(q,Q)T} = \mathbf{U}\mathbf{S}\mathbf{V}^H, \tag{2.73}$$

où la matrice $\mathbf{K}^{(q,Q)}$ est la matrice définie dans (2.21). Si $R < \min(I_q, \prod_{\substack{t=1 \\ t\neq q}}^{Q} I_t)$, il existe une matrice $\mathbf{W} \in \mathbb{C}^{R\times R}$ inversible telle que

$$\mathbf{F}^{(q)}\mathbf{W} = \mathbf{U}\mathbf{S} \tag{2.74}$$

et

$$\mathbf{W}^{-1}\mathbf{K}^{(q,Q)T} = \mathbf{V}^H. \tag{2.75}$$

Compression par SVD et Diagonalisation Conjointe par Congruence. La méthode présentée ici, proposée dans [90], permet de réécrire le problème de décomposition CP comme un problème de Diagonalisation Conjointe de matrices par Congruence (DCC). Le problème de DCC considère un ensemble de matrices conjointement congruent à un ensemble de matrices diagonales. Nous renvoyons au chapitre suivant pour plus de détails sur la DCC et sa résolution.

La méthode de décomposition CP présentée ici a été détaillée dans $\mathbb{R}$ et pour un tenseur d'ordre 3 ou pour un tenseur d'ordre 4. Nous la résumons ici pour $q = 3$ et $Q = 3$ ainsi :

$$\mathbf{V} = (\mathbf{F}^{(2)} \odot \mathbf{F}^{(1)})\mathbf{W}^{-T}. \tag{2.76}$$

Considérons en premier lieu la fonction suivante :

$$\mathbf{\Phi}(\mathbf{X}, \mathbf{Y}) : \quad \mathbb{R}^{I\times J} \times \mathbb{R}^{I\times J} \to \mathbb{R}^{I\times I\times J\times J} \tag{2.77}$$

$$(\mathbf{X}, \mathbf{Y}) \to \mathbf{\Phi}(\mathbf{X}, \mathbf{Y}) \tag{2.78}$$

avec

$$(\mathbf{\Phi}(\mathbf{X}, \mathbf{Y}))_{ijkl} = x_{ik}y_{jl} + x_{jl}y_{ik} - x_{il}y_{jk} - x_{jk}y_{il}. \tag{2.79}$$

Il est démontré dans [90] que $\mathbf{\Phi}(\mathbf{X}, \mathbf{X})$ est égale à un tenseur à valeurs nulles si et seulement si $\mathbf{X}$ est une matrice de rang inférieur ou égal 1.

La matrice $\mathbf{V}$ est maintenant reformée en R matrices $\mathbf{V}^{(r)} \in \mathbb{R}^{I_2\times I_1}$ de la manière suivante :

$$\forall i_1 \in [1; I_1]_{\mathbb{N}},\ \forall i_2 \in [1; I_2]_{\mathbb{N}},\ \forall r \in [1; R]_{\mathbb{N}},\ V^{(r)}_{i_2,i_1} = V_{(i_2-1)I_1+i_1,r}, \tag{2.80}$$

Il vient alors

$$\forall (s, r) \in [1; R]_{\mathbb{N}},\ \mathbf{\Phi}(\mathbf{V}^{(r)}, \mathbf{V}^{(s)}) = \sum_{t,u=1}^{R} (\mathbf{W}^{-T})_{t,r}(\mathbf{W}^{-T})_{u,s}\mathbf{\Phi}(\mathbf{f}^{(2)}_t\mathbf{f}^{(1)T}_t, \mathbf{f}^{(2)}_u\mathbf{f}^{(1)T}_u). \tag{2.81}$$

Nous pouvons noter que $\mathbf{\Phi}(\mathbf{V}^{(r)}, \mathbf{V}^{(s)}) = \mathbf{\Phi}(\mathbf{V}^{(s)}, \mathbf{V}^{(r)})$.

Soit $\mathbf{M}$ une matrice symétrique de dimension R et satisfaisant le système suivant :

$$\sum_{r,s=1}^{R} M_{r,s}\mathbf{\Phi}(\mathbf{V}^{(r)}, \mathbf{V}^{(s)}) = \mathcal{O}, \tag{2.82}$$

où $\mathcal{O}$ est un tenseur ne contenant que des 0. Sachant que $\mathbf{M}$ est symétrique et que $\mathbf{\Phi}(\mathbf{f}^{(2)}_t\mathbf{f}^{(1)T}_t, \mathbf{f}^{(2)}_t\mathbf{f}^{(1)T}_t) = \mathcal{O}\ \forall 1 \leq t \leq R$, nous obtenons

$$\sum_{r,s=1}^{R} \sum_{\substack{t,u=1 \\ t<u}}^{R} (\mathbf{W}^{-T})_{t,r}(\mathbf{W}^{-T})_{u,s}M_{r,s}\mathbf{\Phi}(\mathbf{f}^{(2)}_t\mathbf{f}^{(1)T}_t, \mathbf{f}^{(2)}_u\mathbf{f}^{(1)T}_u) = \mathcal{O}. \tag{2.83}$$

En posant

$$D_{t,u} = \sum_{r,s=1}^{R} (\mathbf{W}^{-T})_{t,r}(\mathbf{W}^{-T})_{u,s}M_{r,s} \tag{2.84}$$

et en supposant que les tenseurs $\mathbf{\Phi}(\mathbf{f}^{(2)}_t\mathbf{f}^{(1)T}_t, \mathbf{f}^{(2)}_u\mathbf{f}^{(1)T}_u)$ sont linéairement indépendants pour $1 \leq t < u \leq R$, alors $D_{t,u} = 0$ si $t \neq u$. En réécrivant alors (2.84) sous sa forme matricielle, nous avons

$$\mathbf{M} = \mathbf{W}^T\mathbf{D}\mathbf{W}, \tag{2.85}$$

où $\mathbf{D}$ est une matrice diagonale. Ainsi n'importe quelle matrice diagonale $\mathbf{D}$ génère une matrice $\mathbf{M}$ permettant de résoudre le système (2.82). Donc, toujours en supposant que les

tenseurs $\mathbf{\Phi}(\mathbf{f}_t^{(2)}\mathbf{f}_t^{(1)T}, \mathbf{f}_u^{(2)}\mathbf{f}_u^{(1)T})$ sont linéairement indépendants, il existe un ensemble de R matrices symétriques $\mathbf{M}^{(r)}$ vérifiant (2.82) de la forme :

$$\begin{aligned} \mathbf{M}^{(1)} &= \mathbf{W}^T\mathbf{D}^{(1)}\mathbf{W} \\ &\vdots \\ \mathbf{M}^{(R)} &= \mathbf{W}^T\mathbf{D}^{(R)}\mathbf{W}. \end{aligned} \tag{2.86}$$

Nous obtenons donc bien un ensemble de matrices $\mathbf{M}^{(r)}$ conjointement congruent à une matrice diagonale $\mathbf{D}^{(r)}$. En pratique les matrices $\mathbf{M}^{(r)}$ sont obtenues en calculant le noyau de (2.82).

Par ailleurs, $\mathbf{W}$ est estimée comme une matrice résolvant (2.86). Ainsi, il est possible de remonter aux matrices facteurs de la manière suivante :

$$\mathbf{F}^{(3)} = \mathbf{USW}^{-1} \tag{2.87}$$

$$(\mathbf{F}^{(2)} \odot \mathbf{F}^{(1)}) = \mathbf{VW}^T. \tag{2.88}$$

Finalement, afin de séparer les matrices $\mathbf{F}^{(2)}$ et $\mathbf{F}^{(1)}$, R matrices $\mathbf{G}_r \in \mathbb{R}^{I_2 \times I_1}$ sont construites à partir des colonnes de $(\mathbf{F}^{(2)} \odot \mathbf{F}^{(1)})$ de la manière suivante :

$$(\mathbf{G}_r)_{i_2,i_1} = (\mathbf{F}^{(2)} \odot \mathbf{F}^{(1)})_{(i_2-1)I_1+i_1,r} \quad \forall r = 1, \ldots, R. \tag{2.89}$$

Par conséquent,

$$\mathbf{G}_r = \mathbf{f}_r^{(2)}\mathbf{f}_r^{(1)T}. \tag{2.90}$$

Les colonnes de $\mathbf{F}^{(2)}$ et de $\mathbf{F}^{(1)}$ sont donc estimées à partir d'une approximation de rang 1 des matrices $\mathbf{G}_r$, comme par exemple une SVD tronquée à l'ordre 1.

Cette méthode est particulièrement intéressante dans les cas où le problème de décomposition CP est mal conditionné ou lorsque que le rang utile du tenseur est grand.

Compression par SVD et Diagonalisation Conjointe par Similitude. Cette méthode a été originalement proposée dans [22] sous le nom de DIAG (pour *DIrect AlGorithm*). Une variante nommée SSD-CP (pour *Simultaneous Schur Decomposition-CP*) a ensuite été proposée dans [91]. Nous les décrivons ici.

Tout d'abord, la matrice $\mathbf{K}^{(q,Q)}$ peut être réécrite comme

$$\mathbf{K}^{(q,Q)} = \mathbf{F}^{(q-1)} \odot \mathbf{K}_2^{(q,Q)} \tag{2.91}$$

avec $\mathbf{K}_2^{(q,Q)} = \mathbf{F}^{(q-2)} \odot \ldots \odot \mathbf{F}^{(1)} \odot \mathbf{F}^{(Q)} \odot \ldots \odot \mathbf{F}^{(q+1)}$. En utilisant la propriété (1.4) du produit de Khatri-Rao, nous obtenons :

$$\mathbf{K}^{(q,Q)T} = \left[\mathbf{\Lambda}^{(1)}\mathbf{K}_2^{(q,Q)T}, \ldots, \mathbf{\Lambda}^{(I_{q-1})}\mathbf{K}_2^{(q,Q)T}\right], \tag{2.92}$$

où $\mathbf{\Lambda}^{(i)} \in \mathbb{C}^{R\times R}$ est une matrice diagonale contenant la i-ème ligne de $\mathbf{F}^{(q-1)}$. D'après les équations (2.75) et (2.92), il vient

$$\mathbf{V}^H = \mathbf{W}^{-1}\left[\mathbf{\Lambda}^{(1)}\mathbf{K}_2^{(q,Q)T}, \ldots, \mathbf{\Lambda}^{(I_{q-1})}\mathbf{K}_2^{(q,Q)T}\right] = \left[\mathbf{\Gamma}^{(1)T}, \ldots, \mathbf{\Gamma}^{(I_{q-1})T}\right] \tag{2.93}$$

avec

$$\forall i \in [1; I_{q-1}]_{\mathbb{N}},\ \mathbf{\Gamma}^{(i)} = \mathbf{K}_2^{(q,Q)}\mathbf{\Lambda}^{(i)}\mathbf{W}^{-T}. \tag{2.94}$$

En calculant les pseudo-inverses des matrices $\mathbf{\Gamma}^{(i)}$, il est possible de construire un ensemble de $I_{q-1}(I_{q-1}-1)$ matrices

$$\forall(i_1,i_2)\in[1;I_{q-1}]_{\mathbb{N}}, i_2\neq i_1 \quad \mathbf{M}^{(i_1,i_2)}=\mathbf{\Gamma}^{(i_1)\dagger}\mathbf{\Gamma}^{(i_2)}. \tag{2.95}$$

En remplaçant les matrices $\mathbf{\Gamma}^{(i)}$ par leur définition, il vient

$$\forall(i_1,i_2)\in[1;I_{q-1}]_{\mathbb{N}}, i_2\neq i_1\ \mathbf{M}^{(i_1,i_2)} \;=\; \mathbf{W}^T\mathbf{D}^{(i_1,i_2)}\mathbf{W}^{-T}, \tag{2.96}$$

où la matrice $\mathbf{D}^{(i_1,i_2)}$ est une matrice diagonale contenant la division terme à terme de la i_2-ème ligne de $\mathbf{F}^{(q-1)}$ par la i_1-ème ligne de $\mathbf{F}^{(q-1)}$. Grâce à la pseudo-inverse, si la i_1-ème ligne de $\mathbf{F}^{(q-1)}$ contient des zéros, les éléments correspondant de la matrice $\mathbf{D}^{(i_1,i_2)}$ sont aussi égaux à zéro.

Les matrices $\mathbf{M}^{(i_1,i_2)}$ sont donc conjointement diagonalisables dans la même base de vecteurs propres et correspondent à un problème de Diagonalisation Conjointe par Similitude (DCS).

Une fois le problème de DCS résolu les deux approches diffèrent. Dans DIAG, nous utilisons la matrice $\mathbf{W}$ estimée et nous remontons aux matrices facteurs à l'aide des deux équations suivantes :

$$\mathbf{F}^{(q)}=\mathbf{USW}^{-1} \tag{2.97}$$

et

$$\mathbf{K}^{(q,Q)T}=\mathbf{WV}^H. \tag{2.98}$$

En formant un tenseur de rang 1 à partir de la r-ème colonne de $\mathbf{K}^{(q,Q)}$, il suffit d'appliquer une HOSVD tronquée à l'ordre 1 sur ce tenseur pour retrouver la r-ème colonne de chacune des $Q-1$ matrices composant $\mathbf{K}^{(q,Q)}$.

Dans SSD-CP, au contraire, nous utilisons la structure des matrices $\mathbf{D}^{(i_1,i_2)}$. En effet, pour chaque colonne r de $\mathbf{F}^{(q-1)}$, nous avons $I_{q-1}(I_{q-1}-1)$ équations de la forme :

$$\forall i,j,\ i\neq j,\ F_{i,r}^{(q-1)}-D_{r,r}^{(i,j)}F_{j,r}^{(q-1)}=0. \tag{2.99}$$

Le système d'équations (2.99) a une infinité de solutions. Cependant, grâce à l'indétermination d'échelle du problème de décomposition CP nous pouvons fixer à 1 les termes de la première ligne de $\mathbf{F}^{(q-1)}$ et résoudre ce système. Les matrices facteurs restantes peuvent être retrouvées de plusieurs manières différentes. Par exemple, en prenant la matrice de dépliement du tenseur $\mathcal{T}$ dans le $q-1$-ème mode nous obtenons

$$\mathbf{K}^{(q-1,Q)T}=\mathbf{F}^{(q-1)\dagger}\mathbf{T}_{(q-1)}. \tag{2.100}$$

Puis les matrices facteurs peuvent être extraites de la matrice $\mathbf{K}^{(q-1,Q)}$ comme dans DIAG. On pourrait aussi réécrire Q fois le problème de DCS à partir de toutes les matrices de dépliement de $\mathcal{T}$ et résoudre Q fois (2.99). Ces deux méthodes ont les mêmes avantages que l'approche SECSI. Néanmoins, leur complexité numérique est beaucoup plus faible et la condition nécessaire de fonctionnement est moins stricte. La matrice de dépliement $\mathbf{T}_{(q)}$ n'est pas la seule manière de déplier un tenseur. Nous notons ici la matrice de dépliement $\mathbf{T}(P)$ du tenseur $\mathcal{T}$ définie de la manière suivante :

$$\forall(m,n)\in[1;\pi_1^P]_{\mathbb{N}}\times[1;\pi_{P+1}^Q]_{\mathbb{N}}\ T(P)_{m,n}=T_{i_1,\ldots,i_Q} \tag{2.101}$$

où $\pi_a^a = I_a$, $\pi_a^b = I_a I_{a+1} \dots I_b$ et

$$\forall m \in [1; \pi_1^P]_{\mathbb{N}},\ m = i_1 + \sum_{q=2}^{P} (i_q - 1)\pi_1^{q-1}, \tag{2.102}$$

$$\forall n \in [1; \pi_{P+1}^Q]_{\mathbb{N}},\ m = i_{P+1} + \sum_{q=P+2}^{Q} (i_q - 1)\pi_{P+1}^{q-1}. \tag{2.103}$$

Ainsi contrairement à la matrice $\mathbf{T}_{(q)}$, la matrice $\mathbf{T}(P)$ permet de fusionner les P premiers modes du tenseur sur ses lignes et donc les $Q-P$ autres modes sur ses colonnes. La matrice $\mathbf{T}(P)$ a alors pour expression :

$$\mathbf{T}(P) = (\mathbf{F}^{(P)} \odot \dots \odot \mathbf{F}^{(1)})(\mathbf{F}^{(Q)} \odot \dots \odot \mathbf{F}^{(P+1)})^T. \tag{2.104}$$

Le raisonnement détaillé ci-dessus peut être alors repris en remplaçant la matrice $\mathbf{K}^{(q,Q)}$ par $\mathbf{F}^{(Q)} \odot \dots \odot \mathbf{F}^{(P+1)}$ et la matrice $\mathbf{F}^{(q)}$ par $\mathbf{F}^{(P)} \odot \dots \odot \mathbf{F}^{(1)}$.

En appliquant alors la SVD de (2.73) sur la matrice $\mathbf{T}(P)$ plutôt que sur la matrice $\mathbf{T}_{(q)}$, une condition générale nécessaire de fonctionnement de DIAG et SSD-CP est :

$\exists\ P \in [1; Q-2]_{\mathbb{N}}$, $\exists f_I$ une permutation des Q premiers entiers naturels et $\exists\ q_s > P$ tels que

$$\prod_{i=1}^{P} I(f_I(i)) \geq R \text{ et } \prod_{\substack{i=P+1 \\ i \neq q_s}}^{Q} I(f_I(i)) \geq R \tag{2.105}$$

Dans les chapitres suivants, nous présenterons et proposerons plusieurs méthodes pour résoudre le problème de DCS.

2.4 Bilan du chapitre

Dans ce chapitre, nous avons introduit différents outils mathématiques d'algèbre multilinéaire. Puis nous avons présenté le problème de décomposition CP ainsi que les différentes manière d'écrire ce dernier : l'expression tensorielle, les expressions matricielles en fonction des matrices de dépliement ou des tranches du tenseur ou encore l'expression par élément du tenseur. Nous avons alors proposé un état de l'art des méthodes de décomposition CP en distinguant trois types d'approches : les méthodes de type alterné, les méthodes de descente et les méthodes basées sur une réduction de dimensions. Ces dernières peuvent conduire à un problème de décomposition CP de plus petites dimensions ou bien à des problèmes de diagonalisation conjointe de matrices.

Nos innovations portant sur la diagonalisation conjointe par similitude induite par une réduction de dimensions par SVD (algorithmes DIAG et SSD-CP). Le chapitre suivant concernera les différents problèmes de diagonalisation conjointe de matrices. Nous y présenterons aussi différentes méthodes de la littérature permettant de résoudre ces problèmes.

Chapitre 3

La diagonalisation conjointe de matrices

Nous allons maintenant aborder le problème de diagonalisation conjointe de matrices. La forme la plus générale de ce problème s'écrit de la manière suivante :

$$\mathbf{M}^{(k)} = \mathbf{A}^{(1)}\mathbf{D}^{(k)}\mathbf{A}^{(2)\ddagger},\ \forall k \in [1, K]_{\mathbb{N}}, \tag{3.1}$$

avec $\mathbf{M}^{(k)} \in \mathbb{C}^{J_1 \times J_2}$, $\mathbf{A}^{(1)} \in \mathbb{C}^{J_1 \times R}$, $\mathbf{A}^{(2)} \in \mathbb{C}^{J_2 \times R}$, $\mathbf{D}^{(k)} \in \mathbb{C}^{R \times R}$ une matrice diagonale et où le symbole ‡ correspond soit à l'opérateur transposé, soit à l'opérateur transposé hermitienne ou alors à l'opérateur inverse (la matrice $\mathbf{A}^{(2)}$ doit alors être carrée et inversible).

Comme nous l'avons vu précédemment, les K matrices $\mathbf{M}^{(k)}$ peuvent correspondre aux tranches d'un tenseur d'ordre 3 du problème de décomposition CP. Ainsi, le problème (3.1) est équivalent au problème de décomposition CP et les méthodes vues dans le chapitre 2 peuvent aussi être utilisées pour résoudre un problème de diagonalisation conjointe de matrices.

3.1 Préliminaires

Nous nous intéresserons ici et dans la suite de ce chapitre aux cas où les matrices $\mathbf{M}^{(k)}$ sont carrées *i.e* $J_1 = J_2 = R$ et où $\mathbf{A}^{(1)} = \mathbf{A}^{(2)}$. Par conséquent, nous noterons maintenant $\mathbf{A}$ les matrices $\mathbf{A}^{(1)}$ et $\mathbf{A}^{(2)}$. Nous présentons ici plusieurs problèmes spécifiques de diagonalisation conjointe de matrices :

- La diagonalisation conjointe d'un ensemble de matrices symétriques

  $$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{T},\ \forall k \in [1, K]_{\mathbb{N}}, \tag{3.2}$$

  ou de matrices de la forme

  $$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{H},\ \forall k \in [1, K]_{\mathbb{N}}, \tag{3.3}$$

  appelée Diagonalisation Conjointe par Congruence (DCC). Dans l'équation (3.3), les matrices $\mathbf{M}^{(k)}$ sont symétriques hermitiennes lorsque les matrices $\mathbf{D}^{(k)}$ ont des valeurs réelles. Dans les cas particuliers où $\mathbf{A}^T\mathbf{A} = \mathbf{I}$ ou $\mathbf{A}^H\mathbf{A} = \mathbf{I}$ nous parlons respectivement de Diagonalisation Conjointe par Congruence Orthogonale (DCC-O) et de Diagonalisation Conjointe par Congruence Unitaire (DCC-U).

- la décomposition conjointe en éléments propres de matrices partageant la même base de vecteurs propres
$$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1},\ \forall k \in [1, K]_{\mathbb{N}}, \tag{3.4}$$
appelée aussi Diagonalisation Conjointe par Similitude (DCS).

Les problèmes de diagonalisation conjointe sont au cœur de nombreuses méthodes de séparation de sources basées sur l'utilisation de cumulants d'ordre supérieur [43, 92, 93]. Une étude très complète des méthodes de DCC est disponible dans [94].

Bien que nos travaux soient focalisés sur le problème de DCS, nous ne pouvons omettre de parler des quatre autres problèmes. Premièrement, il est évident que la DCC-O et la DCC-U sont à la fois des problèmes de DCC et de DCS. Deuxièmement, la plupart des méthodes de DCS sont inspirées des méthodes de DCC. Enfin, le problème de DCC peut être réécrit en problème de DCS lorsqu'au moins une des matrices $\mathbf{M}^{(k)}$ est inversible [95]. En effet, soient
$$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{\ddagger},\ \forall k \in [1; K]_{\mathbb{N}}, \tag{3.5}$$
avec $\ddagger \equiv T$ ou $\ddagger \equiv H$, si $\exists k_2 \in [1; K]_{\mathbb{N}}$ tel que $\mathbf{M}^{(k_2)}$ soit inversible, nous avons alors
$$\forall k_1 \in [1; K]_{\mathbb{N}},\ \mathbf{M}^{(k_1)}\mathbf{M}^{(k_2)-1} = \mathbf{A}\mathbf{D}^{(k_1)}\mathbf{D}^{(k_2)-1}\mathbf{A}^{-1}. \tag{3.6}$$
Des conditions nécessaires et suffisantes d'unicité ont été fournies pour les problèmes de DCS [96], de DCC dans le cas réel [97] et dans les cas symétrique complexe et symétrique hermitien [98].

Théorème 3.1.1. *Soit la matrice*
$$\mathbf{\Omega} = \begin{pmatrix} D_{1,1}^{(1)} & \cdots & D_{1,1}^{(K)} \\ \vdots & \cdots & \vdots \\ D_{R,R}^{(1)} & \cdots & D_{R,R}^{(K)} \end{pmatrix} \tag{3.7}$$
contenant sur ses colonnes les diagonales des matrices $\mathbf{D}^{(k)}$.

La DCC est alors essentiellement unique si et seulement si les lignes de $\mathbf{\Omega}$ *sont différentes du vecteur nul et non-colinéaires deux à deux.*

Le problème de DCS, quant à lui, est essentiellement unique si et seulement si les lignes de $\mathbf{\Omega}$ *sont toutes différentes (i.e.* $\forall m, n$ *avec* $m \neq n$, $\mathbf{\Omega}_{m\bullet} - \mathbf{\Omega}_{n\bullet} \neq \mathbf{0}$ *où* $\mathbf{\Omega}_{m\bullet}$ *et* $\mathbf{\Omega}_{n\bullet}$ *sont respectivement la* $m^{ème}$ *et la* $n^{ème}$ *ligne* $\mathbf{\Omega}$*). Ces résultats sont valables pour* $K > 1$.

Comme pour le problème de décomposition CP, les problèmes de diagonalisation conjointe admettent une indétermination d'échelle et de permutation, c'est pourquoi nous parlons aussi ici d'unicité essentielle. Nous expliciterons cette indétermination ultérieurement.

Remarque : en considérant les équations (3.5) et (3.6), nous pouvons remarquer qu'il y a équivalence entre la condition d'unicité de la DCC et celle de la DCS.

En effet, considérons un ensemble de matrices de DCC défini comme dans (3.5) *i.e.* $\ddagger \equiv T$ ou $\ddagger \equiv H$. Comme explicité dans le théorème 3.1.1, la DCC d'un tel ensemble est essentiellement unique si et seulement si $\forall r \in [1; R]_{\mathbb{N}},\ \exists k \in [1; K]_{\mathbb{N}}$ tel que $D_{r,r}^{(k)} \neq 0$ et si
$$\forall\ (r_1, r_2) \in [1; R]_{\mathbb{N}},\ \exists (k_1, k_2) \in [1; K]_{\mathbb{N}}\ \text{ tels que } \frac{D_{r_1,r_1}^{(k_1)}}{D_{r_2,r_2}^{(k_1)}} \neq \frac{D_{r_1,r_1}^{(k_2)}}{D_{r_2,r_2}^{(k_2)}} \tag{3.8}$$

avec $D^{(k_1)}_{r_2,r_2} \neq 0$ et $D^{(k_2)}_{r_2,r_2} \neq 0$. Le problème de DCS obtenu grâce à l'équation (3.6) est quant à lui essentiellement unique si et seulement si

$$\forall (r_1, r_2) \in [1; R]_{\mathbb{N}},\ \exists (k_1, k_2) \in [1; K]_{\mathbb{N}} \text{ tels que } \frac{D^{(k1)}_{r_1,r_1}}{D^{(k_2)}_{r_1,r_1}} \neq \frac{D^{(k_1)}_{r_2,r_2}}{D^{(k2)}_{r_2,r_2}} \tag{3.9}$$

et $D^{(k_2)}_{r_1,r_1} \neq 0$ et $D^{(k_2)}_{r_2,r_2} \neq 0$.

Nous avons donc bien $\frac{D^{(k1)}_{r_1,r_1}}{D^{(k_2)}_{r_1,r_1}} \neq \frac{D^{(k_1)}_{r_2,r_2}}{D^{(k2)}_{r_2,r_2}} \Leftrightarrow \frac{D^{(k_1)}_{r_1,r_1}}{D^{(k_1)}_{r_2,r_2}} \neq \frac{D^{(k_2)}_{r_1,r_1}}{D^{(k_2)}_{r_2,r_2}}$.

Idéalement, une simple décomposition en éléments propres sur n'importe quelle matrice $\mathbf{M}^{(k)}$ ou $\mathbf{M}^{(k_1)}\mathbf{M}^{(k_2)-1}$ suffirait pour résoudre respectivement les problèmes de DCS et de DCC. Cependant, la décomposition en éléments propres d'une matrice est unique si et seulement si cette matrice a toutes ses valeurs propres différentes. Ainsi prendre plusieurs matrices à diagonaliser permet d'avoir des conditions d'identifiabilité moins fortes. De plus, dans la plupart des cas pratiques, il existe un écart (bruit) entre les matrices $\mathbf{M}^{(k)}$ correspondant au modèle de diagonalisation conjointe et les données observées $\mathbf{M}^{(k)}_b$ *i.e.*

$$\mathbf{M}^{(k)}_b = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{\ddagger} + \mathbf{E}^{(k)},\ \forall k \in [1; K]_{\mathbb{N}}, \tag{3.10}$$

où $\mathbf{E}^{(k)}$ est une matrice de dimension R représentant le bruit. Dans de nombreux travaux, comme par exemple [38, 99–101], il a été montré à l'aide de simulations numériques que plus le nombre de matrices à diagonaliser est grand, plus les résultats obtenus sont robustes au bruit (pour tous les problèmes de diagonalisations conjointes).

Les premières approches de diagonalisation conjointe de matrices ont concerné un ensemble de deux matrices seulement. Ces méthodes sont connues sous le nom de GEVD (de l'anglais *Generalized Eigen Value Decomposition*) ou GSVD (de l'anglais *Generalized Singular Value Decomposition*). La GEVD a l'avantage de fournir directement (sous-entendu via un processus non-itératif) une solution. Par la suite, une méthode basée sur la GEVD de deux matrices représentant de l'ensemble a été présentée dans [102]. Cette méthode appelée EJD (de l'anglais *Exact Joint Diagonalization*) fournit, elle aussi, directement une solution sous-optimale. Dans [103] l'algorithme DIEM (pour *DIagonlization using Equivalent Matrices*), inspirée de EJD, fourni aussi une solution directe dans le cadre du problème de DCC.

3.2 Les stratégies de résolution

Nous considérons ici le problème générique suivant :

$$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{\ddagger},\ \forall k \in [1, K]_{\mathbb{N}}. \tag{3.11}$$

3.2.1 Fonctions de coût

Les différentes stratégies de résolution itératives proposent de considérer le problème de diagonalisation conjointe comme un problème d'optimisation. C'est pourquoi nous présentons ici les différents critères utilisés.

Critère direct. Le critère direct, utilisé dans [44] dans le cadre de la DCC-U, est une mesure quadratique de la différence entre les matrices à diagonaliser et leur modèle théorique. Ce dernier s'écrit comme :

$$C_{\text{direct}}(\mathbf{A}, \mathbf{D}^{(k)}) = \sum_{k=1}^{K} ||\mathbf{M}^{(k)} - \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{\ddagger}||_F^2. \tag{3.12}$$

Minimiser la fonction de coût (3.12) va donc nous donner une solution optimale au sens des moindres carrés pour la matrice $\mathbf{A}$ et pour les K matrices $\mathbf{D}^{(k)}$. Il est notable que ce critère est utilisé pour l'algorithme non-itératif EJD cité précédemment. Par la suite, ce critère a été utilisé pour les problèmes de DCC dans [104–106]. Nous avons déjà vu une fonction de coût similaire lors du chapitre précédent. En effet, l'ensemble de matrices $\mathbf{M}^{(k)}$ peut se réécrire comme un tenseur $\mathcal{M}$ dont les matrices facteurs de sa décomposition CP au rang R sont $\mathbf{A}$, $(\mathbf{A}^{\ddagger})^T$ et une matrice $\mathbf{C} \in \mathbb{C}^{K\times R}$ contenant sur chacune de ses lignes la diagonale d'une matrice $\mathbf{D}^{(k)}$. Donc le critère direct (3.12) est égal à la fonction de coût de la décomposition CP (2.29) d'un tenseur d'ordre 3 et peut s'écrire :

$$C_{\text{direct}}(\mathbf{A}, \mathbf{C}) = ||\mathbf{M}_{(1)} - \mathbf{A}(\mathbf{C} \odot (\mathbf{A}^{\ddagger})^T)^T||_F^2. \tag{3.13}$$

Ainsi, ce critère a l'avantage de pouvoir traiter le problème de diagonalisation dans le cas général (3.1) de la même manière que dans le chapitre précédent. Dans [107], le critère direct est d'ailleurs minimisé en utilisant une méthode de descente de type Gauss-Newton dans le cadre de la DCC d'un ensemble de matrices symétriques hermitiennes. Ce critère a aussi été utilisé pour un algorithme de type ADMM afin d'imposer la non-négativité sur la matrice $\mathbf{A}$ et sur les K matrices $\mathbf{D}^{(k)}$ [108].

Critère inverse. Le critère inverse est une mesure quadratique de la diagonalité des matrices à diagonaliser conjointement par une matrice $\mathbf{B}$. Ce dernier s'écrit comme :

$$C_{\text{inverse}}(\mathbf{B}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{\mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{\ddagger}\}\|_F^2. \tag{3.14}$$

Le principe de résolution est ici différent dans sa conception. En effet, le but n'est pas de faire correspondre les paramètres à estimer au modèle (3.11) mais de diagonaliser l'ensemble des matrices $\mathbf{M}^{(k)}$ *i.e.* minimiser les termes hors diagonaux de l'ensemble des matrices $\mathbf{M}^{(k)}$ simultanément.

Pour se faire la matrice $\mathbf{B}$, appelée matrice diagonalisante, doit être estimée telle que les matrices

$$\mathbf{N}^{(k)} = \mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{\ddagger}, \quad \forall\ k \in [1; K]_{\mathbb{N}} \tag{3.15}$$

soient aussi diagonales que possible. Lorsque la matrice $\mathbf{B}$ diagonalise parfaitement l'ensemble de matrices, elle est égale à l'inverse de la matrice $\mathbf{A}$ à un facteur d'échelle et à une permutation près. En effet, soient $\mathbf{\Lambda}$ une matrice diagonale inversible et $\mathbf{P}$ une matrice de permutation telles que $\mathbf{B}\mathbf{A} = \mathbf{\Lambda}\mathbf{P}$, alors

$$\begin{aligned} \mathbf{N}^{(k)} &= \mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{\ddagger} \\ &= \mathbf{\Lambda}\mathbf{P}\mathbf{D}^{(k)}\mathbf{P}^{\ddagger}\mathbf{\Lambda}^{\ddagger} \quad \forall k \in [1; K]_{\mathbb{N}} \end{aligned} \tag{3.16}$$

sont diagonales quel que soit l'opérateur représenté par ‡ et quelles que soient les valeurs des matrices $\mathbf{\Lambda}$ et $\mathbf{P}$. L'indétermination d'échelle permet d'estimer la matrice $\mathbf{B}$ avec R degrés de liberté.

Dans [49], ce critère est minimisé grâce à la méthode du gradient pour de la DCC. Cependant dans nombres d'algorithmes la matrice $\mathbf{B}$ est estimée par le biais de mises à jour multiplicatives. L'avantage de cette fonction de coût est qu'il suffit d'estimer la matrice $\mathbf{B}$ pour la minimiser, ce qui n'est pas le cas du critère direct (3.12) dans lequel les matrices $\mathbf{D}^{(k)}$ doivent aussi être estimées. En contre partie, ce critère a trois défauts : premièrement la matrice $\mathbf{A}$ doit être inversible (ce qui est toujours le cas pour le problème de DCS), deuxièmement il admet la solution triviale $\mathbf{B} = \mathbf{0}$ dans le cas de la DCC et troisièmement ce critère est sensible aux variations de norme des matrices $\mathbf{N}^{(k)}$. La solution triviale $\mathbf{B} = \mathbf{0}$ est facilement évitable, car comme explicité précédemment la matrice $\mathbf{B}$ peut être estimée avec R degrés de liberté ce qui permet de lui imposer certaines structures comme par exemple une diagonale dominante ou encore un déterminant égal à 1. Pour palier le problème de variation d'échelle, le critère suivant a été proposé dans [97] et [109] pour traiter des données réelles dans le problème de DCC :

$$C_{\text{inverse2}}(\mathbf{B}) = \sum_{k=1}^{K} \|\mathbf{M}^{(k)} - \mathbf{B}^{-1}\mathsf{Diag}\{\mathbf{N}^{(k)}\}\mathbf{B}^{-T}\|_F^2. \tag{3.17}$$

Du fait de la propriété d'invariance de la norme de Frobenius au transformation orthogonal, minimiser le critère (3.14) est équivalent à maximiser le critère suivant pour le problème de DCC-O :

$$C_{\text{inverse3}}(\mathbf{B}) = \sum_{k=1}^{K} \|\mathsf{Diag}\{\mathbf{B}\mathbf{N}^{(k)}\mathbf{B}^{T}\}\|_F^2. \tag{3.18}$$

Dans [110], ce critère est maximisé à l'aide de la méthode du gradient.

Dans [111, 112], une méthode est proposée pour séparer des signaux non-négatifs. Le critère (3.14) est alors réécrit comme un critère direct en inversant les matrices $\mathbf{M}^{(k)}$:

$$C_{\text{direct2}}(\mathbf{H}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{\mathbf{A}^{T}\mathbf{M}^{(k)-1}\mathbf{A}\}\|_F^2 \tag{3.19}$$

avec $\mathbf{A} = \mathbf{H} \boxdot \mathbf{H}$. L'ensemble de matrices symétriques est donc ici à valeurs réelles et la matrice $\mathbf{A}$ contient des valeurs non-négatives. Pour se faire, la matrice $\mathbf{A}$ est paramétrisée comme $\mathbf{A} = \mathbf{H} \boxdot \mathbf{H}$, ainsi le critère (3.19) est optimisé par rapport à la matrice $\mathbf{H}$.

Critère log-vraisemblance. Le critère du maximum de vraisemblance a été proposé en premier lieu par Flury pour résoudre le problème de DCC-O de matrices définies positives [113], puis il a été généralisé par Pham pour la DCC de matrices définies positives [114]. Ce dernier s'écrit

$$C_{\log}(\mathbf{B}) = \sum_{k=1}^{K} n_k(\log(\det(\mathsf{Diag}\{\mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{\ddagger}\})) - \log(\det(\mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{\ddagger}))), \tag{3.20}$$

avec $\ddagger \equiv H$ ou $\ddagger \equiv T$. Ce critère permet de résoudre les problèmes de diagonalisation conjointe en conservant la symétrie et la non-négativité de la diagonale de matrices symétriques définies positives. Par conséquent, ce critère n'est pas directement adapté au problème de DCS. Pham démontre que sa méthode (appelée JD-BGL) fait décroître le critère (3.20) à chaque itération. Enfin, la méthode proposée dans [115] minimise une version approximée de (3.20).

3.2.2 Stratégie de résolution globale

De nombreuses méthodes de diagonalisation sont basées sur la mise à jour multiplicative de la matrice $\mathbf{B}$. En effet, à chaque itération une matrice $\mathbf{X}$ est calculée pour mettre à jour $\mathbf{B}$ de la manière suivante :

$$\mathbf{B} \leftarrow \mathbf{XB}. \tag{3.21}$$

Par conséquent, l'ensemble de matrices $\mathbf{N}^{(k)}$ défini à l'équation (3.15) est mis à jour ainsi

$$\forall\ k \in [1; K]_{\mathbb{N}}\ \mathbf{N}^{(k)} \leftarrow \mathbf{X}\mathbf{N}^{(k)}\mathbf{X}^{\ddagger}. \tag{3.22}$$

La matrice $\mathbf{X}$ est choisie de sorte à minimiser l'un des critères précédents. Cette mise à jour multiplicative est répétée jusqu'à atteindre un point stationnaire. Par exemple, la fonction de coût (3.14) doit être optimisée en fonction de la matrice $\mathbf{X}$ à chaque itération. Nous avons donc

$$C_{\text{inverse}}(\mathbf{X}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{\mathbf{X}\mathbf{N}^{(k)}\mathbf{X}^{\ddagger}\}\|_F^2. \tag{3.23}$$

L'indétermination d'échelle et le processus algorithmique nous permettent aussi ici de factoriser une matrice diagonale inversible à la matrice $\mathbf{X}$. Ainsi nous avons R degrés de liberté et donc nous pouvons nous contenter d'estimer $R(R-1)$ paramètres. Par exemple, il est possible de contraindre la matrice $\mathbf{X}$ à n'avoir que des 1 sur sa diagonale. Nous pouvons donc écrire

$$\mathbf{X} = \mathbf{I}_R + \mathbf{Z} \tag{3.24}$$

avec $\mathbf{Z}$ une matrice de dimension R dont les termes diagonaux sont nuls. Cette paramétrisation est connue en traitement du signal plus particulièrement pour les problème de séparation aveugle de sources [116]. Elle a été utilisée par la suite pour la DCC d'un ensemble de matrices symétriques dans [117], puis cette méthode a été adaptée à la DCC de matrices symétriques hermitiennes dans [118]. Elle a enfin été généralisée aux deux problèmes dans [119].

La table algorithmique 2 présente la forme générique classique d'un algorithme de diagonalisation conjointe basé sur une approche globale.

3.2.3 Stratégie de résolution locale

La matrice $\mathbf{X}$ est ici factorisée en $R(R-1)/2$ matrices $\mathbf{X}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$ ne dépendant que de deux paramètres. $\mathbf{X}$ peut donc s'écrire comme

$$\mathbf{X} = \prod_{i=1}^{R-1} \prod_{j=i+1}^{R} \mathbf{X}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)}), \tag{3.25}$$

Algorithme 2 Forme générique des algorithmes de diagonalisation conjointe basés sur une stratégie globale

Soit S_C un critère d'arrêt préalablement défini et It_{max} le nombre d'itération maximal ;

Initialisation de $\mathbf{B}$ avec la matrice identité ou par n'importe quel choix judicieux ;
$it \leftarrow 1$;
while S_C est $faux$ et $it < It_{max}$ **do**
  Calculer les paramètres de la matrice $\mathbf{X}$;
  Mettre à jour $\mathbf{B} \leftarrow \mathbf{XB}$;
  Mettre à jour $\mathbf{N}^{(k)} \leftarrow \mathbf{X}\mathbf{N}^{(k)}\mathbf{X}^{\ddagger}$, $\forall k \in [1, K]_{\mathbb{N}}$;
  Mettre à jour S_C ;
  $it \leftarrow it + 1$;
end while

où les matrices $\mathbf{X}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$ sont égales à la matrice identité excepté pour les termes $X_{i,i}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$, $X_{j,j}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$, $X_{i,j}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$ et $X_{j,i}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$ pouvant prendre n'importe quelles valeurs dans $\mathbb{R}$ ou $\mathbb{C}$ selon les cas considérés. Pour simplifier les notations, nous remplacerons dans la suite $\mathbf{X}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$ par $\mathbf{X}^{(i,j)}$ et les termes $X_{i,i}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$, $X_{j,j}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$, $X_{i,j}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$ et $X_{j,i}^{(i,j)}(p_1^{(i,j)}, p_2^{(i,j)})$ par $X_{i,i}^{(i,j)}$, $X_{j,j}^{(i,j)}$, $X_{i,j}^{(i,j)}$ et $X_{j,i}^{(i,j)}$ respectivement.

Par exemple, pour $R = 5$, la matrice $\mathbf{X}^{(2,4)}$ est donc de la forme

$$\mathbf{X}^{(2,4)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & X_{2,2}^{(2,4)} & 0 & X_{2,4}^{(2,4)} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & X_{4,2}^{(2,4)} & 0 & X_{4,4}^{(2,4)} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.26}$$

La mise à jour globale de l'ensemble de matrices (3.22) est par conséquent décomposée en $R(R-1)/2$ mises à jour locales du type

$$\mathbf{N}^{(k)} \leftarrow \mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)\ddagger} \ \forall k \in [1; K]_{\mathbb{N}}. \tag{3.27}$$

Ainsi pour ce type d'approche, une itération consiste à balayer séquentiellement toutes les paires d'indices (i, j) avec $i < j$. Sur la figure 3.1, nous pouvons observer l'impact d'une transformation locale sur une matrice $\mathbf{N}^{(k)}$. Il est donc notable que la multiplication de droite de (3.27) affecte les colonnes i et j des matrices $\mathbf{N}^{(k)}$ et que la multiplication de gauche affecte les lignes i et j des matrices $\mathbf{N}^{(k)}$. Par conséquent, cette approche consiste à minimiser les i-ème et j-ème lignes et colonnes des matrices $\mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)\ddagger}$ (privées des termes diagonaux bien entendu), ce qui simplifie le processus d'optimisation. Cette stratégie a été inspirée de la méthode de Jacobi [52] pour la diagonalisation d'une seule matrice hermitienne à l'aide de matrices de rotation plane de Givens. Elle a été utilisée par la suite dans l'algorithme JADE [53] traitant le problème de DCC-U. Cette stratégie a été utilisée dans [120] pour de la DCC sans contrainte unitaire. La table algorithmique 3 présente le schéma de fonctionnement général d'un algorithme de diagonalisation conjointe basé sur une approche locale.

Algorithme 3 Forme générique des algorithmes de diagonalisation conjointe basés sur une stratégie locale

Soit S_C un critère d'arrêt préalablement défini et It_{max} le nombre d'itérations maximal ;

Initialisation de $\mathbf{B}$ avec la matrice identité ou par n'importe quel choix judicieux ;
$it \leftarrow 1$;
while S_C est $faux$ et $it < It_{max}$ **do**
  for $i = 1$ to $R-1$ **do**
    for $j = i+1$ to R **do**
      Estimer les paramètres de $\mathbf{X}^{(i,j)}$;
      Mettre à jour $\mathbf{B} \leftarrow \mathbf{X}^{(i,j)}\mathbf{B}$;
      Mettre à jour $\mathbf{N}^{(k)} \leftarrow \mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\left(\mathbf{X}^{(i,j)}\right)^{\ddagger}$, $\forall k \in [1,K]_{\mathbb{N}}$;
    end for
  end for
  Mettre à jour S_C
  $it \leftarrow it+1$;
end while

Nous pouvons aussi remarquer sur la figure 3.1 que les termes $N_{i,i}^{(k)}$, $N_{i,j}^{(k)}$, $N_{j,i}^{(k)}$ et $N_{j,j}^{(k)}$ sont affectés deux fois par la transformation élémentaire (3.27). Ainsi le critère (3.23) peut être approximé de la manière suivante :

$$C'_{\text{inverse}}(\mathbf{X}^{(i,j)}) = \sum_{k=1}^{K} |(\mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)\ddagger})_{i,j}|^2 + |(\mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)\ddagger})_{j,i}|^2. \tag{3.28}$$

Ce critère a été présenté pour la première fois dans [121] pour de la DCC, il a été repris par la suite pour de la DCS dans [22]. Minimiser ce critère n'est pas équivalent à minimiser tous les termes hors diagonaux des matrices $\mathbf{N}^{(k)}$, cependant il offre en pratique de bonnes propriétés de convergence. De plus, il permet de réécrire le problème initial en une série de sous problèmes de taille 2×2. Ainsi en posant

$$\tilde{\mathbf{N}}^{(k)} = \begin{pmatrix} N_{i,i}^{(k)} & N_{i,j}^{(k)} \\ N_{j,i}^{(k)} & N_{j,j}^{(k)} \end{pmatrix}, \quad \forall k \in [1,K]_{\mathbb{N}}; \tag{3.29}$$

et

$$\tilde{\mathbf{X}}^{(i,j)} = \begin{pmatrix} X_{i,i}^{(i,j)} & X_{i,j}^{(i,j)} \\ X_{j,i}^{(i,j)} & X_{j,j}^{(i,j)} \end{pmatrix}, \tag{3.30}$$

la transformation (3.27) devient alors,

$$\tilde{\mathbf{N}}^{(k)} \leftarrow \tilde{\mathbf{X}}^{(i,j)} \begin{pmatrix} N_{i,i}^{(k)} & N_{i,j}^{(k)} \\ N_{j,i}^{(k)} & N_{j,j}^{(k)} \end{pmatrix} \left(\tilde{\mathbf{X}}^{(i,j)}\right)^{-1} \ \forall k \in [1,K]_{\mathbb{N}} \tag{3.31}$$

et le critère approximé (3.28) peut s'écrire

$$C'_{\text{inverse}}(\tilde{\mathbf{X}}^{(i,j)}) = \sum_{k=1}^{K} \left| \left(\tilde{\mathbf{X}}^{(i,j)}\tilde{\mathbf{N}}^{(k)}\tilde{\mathbf{X}}^{(i,j)\ddagger}\right)_{1,2} \right|^2 + \left| \left(\tilde{\mathbf{X}}^{(i,j)}\tilde{\mathbf{N}}^{(k)}\tilde{\mathbf{X}}^{(i,j)\ddagger}\right)_{2,1} \right|^2. \tag{3.32}$$

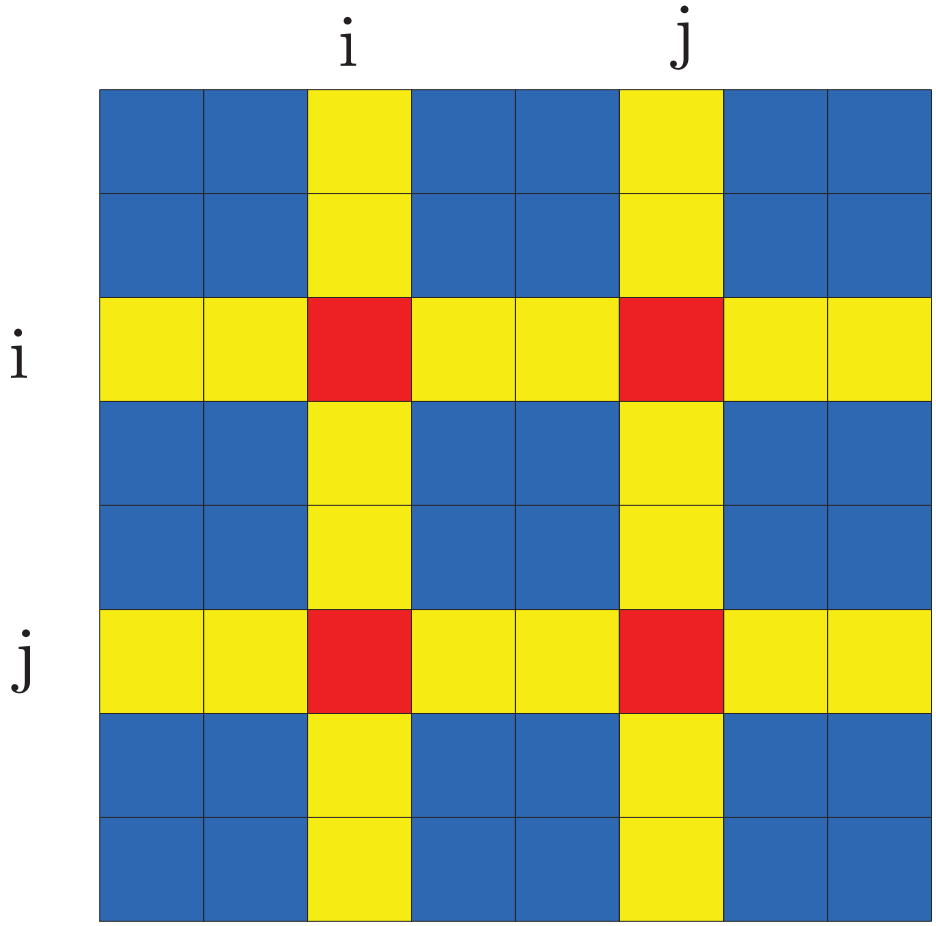

FIGURE 3.1: Schéma de l'impact de la transformation locale (3.27) par la matrice $\mathbf{X}^{(i,j)}$ sur une matrice $\mathbf{N}^{(k)}$ (les cases bleues sont les éléments non affectés par la transformation, les jaunes sont les éléments affectés une seule fois et les rouges sont les éléments affectés deux fois).

Dans [122], il est montré, dans le cadre de la DCC, que les fonctions de coût (3.23) et (3.32) sont équivalentes lorsque nous nous trouvons proche de la solution diagonalisante.

3.2.4 Factorisations matricielles

Les factorisations matricielles LU, QR et dans une moindre mesure la décomposition polaire sont fréquemment utilisées pour résoudre les problèmes de diagonalisation conjointe. Ces dernières permettent de décomposer les matrices $\mathbf{X}$ ou $\mathbf{X}^{(i,j)}$ en plusieurs matrices inversibles ne dépendant que d'un seul paramètre. La matrice $\mathbf{X}$ étant inversible, l'ordre des matrices de chaque factorisation peut être interverti.

Paramétrisation de la matrice X. Soient les propositions suivantes [123, 124] :

Proposition 1. *Toute matrice triangulaire inférieure inversible* $\mathbf{L}$ *peut être décomposée comme*

$$\mathbf{L} = \mathbf{\Lambda} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{L}^{(i,j)}, \tag{3.33}$$

où $\mathbf{\Lambda}$ *est une matrice diagonale inversible et* $\mathbf{L}^{(i,j)}$ *est une matrice triangulaire inférieure élémentaire, c'est-à-dire qu'elle est égale à la matrice identité excepté pour le terme* $L_{i,j}^{(i,j)}$ *(avec* $i < j$*). La matrice* $\mathbf{L}^{(i,j)}$ *est donc de la forme*

$$\mathbf{L}^{(i,j)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & L_{j,i}^{(i,j)} & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}. \tag{3.34}$$

Proposition 2. *Toute matrice triangulaire supérieure inversible* $\mathbf{U}$ *peut être décomposée comme*

$$\mathbf{U} = \mathbf{\Lambda} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{U}^{(i,j)}, \tag{3.35}$$

où $\mathbf{\Lambda}$ *est une matrice diagonale inversible et* $\mathbf{U}^{(i,j)}$ *est une matrice triangulaire supérieure élémentaire, c'est-à-dire qu'elle est égale à la matrice identité excepté pour le terme* $U_{j,i}^{(i,j)}$ *(avec* $i < j$*). La matrice* $\mathbf{U}^{(i,j)}$ *est donc de la forme*

$$\mathbf{U}^{(i,j)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & U_{i,j}^{(i,j)} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}. \tag{3.36}$$

Proposition 3. *Toute matrice orthogonale* $\mathbf{Q}$ *peut être décomposée comme*

$$\mathbf{Q} = \mathbf{\Lambda} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{Q}^{(i,j)}, \tag{3.37}$$

où $\mathbf{\Lambda}$ *est une matrice diagonale inversible et* $\mathbf{Q}^{(i,j)}$ *est une matrice de rotation plane de Givens, c'est-à-dire qu'elle est égale à la matrice identité excepté pour les termes* $Q_{i,i}^{(i,j)}$, $Q_{j,j}^{(i,j)}$, $Q_{j,i}^{(i,j)}$ *et* $Q_{i,j}^{(i,j)}$*. La matrice* $\mathbf{Q}^{(i,j)}$ *est donc de la forme*

$$\mathbf{Q}^{(i,j)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & c^{(i,j)} & \ddots & s^{(i,j)} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & -s^{(i,j)} & \ddots & c^{(i,j)} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \tag{3.38}$$

avec $(c^{(i,j)}$ *et* $s^{(i,j)}$ *deux paramètres réels liés par la relation* $(c^{(i,j)})^2 + (s^{(i,j)})^2 = 1$.

Proposition 4. *Toute matrice unitaire* $\mathbf{G}$ *peut être décomposée comme*

$$\mathbf{G} = \mathbf{\Lambda} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{G}^{(i,j)}, \tag{3.39}$$

où $\mathbf{\Lambda}$ *est une matrice diagonale inversible et* $\mathbf{G}^{(i,j)}$ *est une matrice de rotation plane de Givens, c'est-à-dire qu'elle est égale à la matrice identité excepté pour les terme* $G_{i,i}^{(i,j)}$, $G_{j,j}^{(i,j)}$, $G_{j,i}^{(i,j)}$ *et* $G_{i,j}^{(i,j)}$. *La matrice* $\mathbf{G}^{(i,j)}$ *est donc de la forme*

$$\mathbf{G}^{(i,j)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & c^{(i,j)} & \ddots & \bar{s}^{(i,j)} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & -s^{(i,j)} & \ddots & \bar{c}^{(i,j)} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \tag{3.40}$$

où $c^{(i,j)}$ *et* $s^{(i,j)}$ *sont des paramètres complexes liés par* $|c^{(i,j)}|^2 + |s^{(i,j)}|^2 = 1$.

Les propositions 1 et 2 se démontrent de manière triviale et constructive. Les propositions 3 et 4 découlent directement de la méthode de Jacobi [52].

En prenant encore une fois en compte l'indétermination d'échelle de notre problème, la matrice $\mathbf{X}$ peut être paramétrisée en utilisant soit :

- une factorisation LU

$$\mathbf{X} = \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{L}^{(i,j)} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{U}^{(i,j)}, \tag{3.41}$$

- une factorisation QL dans le cas réel

$$\mathbf{X} = \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{Q}^{(i,j)} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{L}^{(i,j)}, \tag{3.42}$$

- une factorisation QL dans le cas complexe

$$\mathbf{X} = \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{G}^{(i,j)} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{L}^{(i,j)}, \tag{3.43}$$

- une factorisation QR dans le cas réel

$$\mathbf{X} = \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{Q}^{(i,j)} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{U}^{(i,j)}, \tag{3.44}$$

- une factorisation QR dans le cas complexe

$$\mathbf{X} = \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{G}^{(i,j)} \prod_{i=1}^{R} \prod_{j=i+1}^{R-1} \mathbf{U}^{(i,j)}. \tag{3.45}$$

Nous pouvons noter que les matrices $\mathbf{L}^{(i,j)}$, $\mathbf{U}^{(i,j)}$, $\mathbf{Q}^{(i,j)}$ et $\mathbf{G}^{(i,j)}$ sont toutes de déterminant 1. Cela permet d'assurer l'inversibilité de la matrice diagonalisante $\mathbf{B}$. La paramétrisation de la matrice $\mathbf{X}$ avec de telles décompositions a donné naissance à un type de méthodes résumées dans la table algorithmique 4 dans le cas d'une factorisation LU. Ce type d'algorithme a été utilisé pour le problème de DCC en premier lieu dans [109] avec les factorisations QR et LU en minimisant le critère (3.14). Puis dans [122] un algorithme basé sur un factorisation LU et minimisant le critère (3.28) a été proposé dans le cas complexe.

Algorithme 4 Forme générique des algorithmes de diagonalisation conjointe basés sur une stratégie locale et paramétrant la matrice $\mathbf{X}$ avec une factorisation matricielle (utilisant ici une factorisation LU)

Soit S_C un critère d'arrêt préalablement défini et It_{max} le nombre d'itération maximal ;

Initialisation de $\mathbf{B}$ avec la matrice identité ou par n'importe quel choix judicieux ;
$it \leftarrow 1$;
while S_C est $faux$ et $it < It_{max}$ **do**
  for $i = 1$ to $R-1$ **do**
    for $j = i+1$ to R **do**
      Estimer $U_{i,j}^{(i,j)}$;
      Mettre à jour $\mathbf{B} \leftarrow \mathbf{U}^{(i,j)}\mathbf{B}$;
      Mettre à jour $\mathbf{N}^{(k)} \leftarrow \mathbf{U}^{(i,j)}\mathbf{N}^{(k)} \left(\mathbf{U}^{(i,j)}\right)^{\ddagger}$, $\forall k \in [1,K]_{\mathbb{N}}$;
    end for
  end for
  for $i = 1$ to $N-1$ **do**
    for $j = i+1$ to N **do**
      Estimer $L_{i,j}^{(i,j)}$;
      Mettre à jour $\mathbf{B} \leftarrow \mathbf{L}^{(i,j)}\mathbf{B}$;
      Mettre à jour $\mathbf{N}^{(k)} \leftarrow \mathbf{L}^{(i,j)}\mathbf{N}^{(k)} \left(\mathbf{L}^{(i,j)}\right)^{\ddagger}$, $\forall k \in [1,K]_{\mathbb{N}}$;
    end for
  end for
  Mettre à jour S_C ;
  $it \leftarrow it + 1$;
end while

Décomposition des matrices $\mathbf{X}^{(i,j)}$. Les matrices de mise à jour $\mathbf{X}^{(i,j)}$ peuvent aussi être factorisées à l'aide des décompositions matricielles LU, QR et polaire. La décomposition LU de la matrice $\mathbf{X}^{(i,j)}$ s'écrit alors

$$\mathbf{X}^{(i,j)} = \mathbf{L}^{(i,j)}\mathbf{U}^{(i,j)}. \tag{3.46}$$

La décomposition QR peut s'écrire soit

$$\mathbf{X}^{(i,j)} = \mathbf{G}^{(i,j)}\mathbf{L}^{(i,j)} \tag{3.47}$$

ou

$$\mathbf{X}^{(i,j)} = \mathbf{G}^{(i,j)}\mathbf{U}^{(i,j)} \tag{3.48}$$

dans le cas réel, ou alors

$$\mathbf{X}^{(i,j)} = \mathbf{Q}^{(i,j)}\mathbf{L}^{(i,j)} \tag{3.49}$$

ou

$$\mathbf{X}^{(i,j)} = \mathbf{Q}^{(i,j)}\mathbf{U}^{(i,j)} \tag{3.50}$$

dans le cas complexe. L'utilisation de la décomposition polaire,

$$\mathbf{X}^{(i,j)} = \mathbf{Q}^{(i,j)}\mathbf{H}^{(i,j)} \tag{3.51}$$

dans le cas réel a été introduite et justifiée par Antoine Souloumiac dans [121]. La matrice $\mathbf{H}^{(i,j)}$ est appelée matrice symétrique hyperbolique et est définie de la manière suivante :

$$\mathbf{H}^{(i,j)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & c_h^{(i,j)} & \ddots & s_h^{(i,j)} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & s_h^{(i,j)} & \ddots & c_h^{(i,j)} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \tag{3.52}$$

où $c_h^{(i,j)}$ et ${s_h}^{(i,j)}$ deux paramètres réels liés par la relation $({c_h}^{(i,j)})^2 - ({s_h}^{(i,j)})^2 = 1$. Elle a été ensuite étendue au cas complexe dans [95] et s'écrit de la manière suivante :

$$\mathbf{X}^{(i,j)} = \mathbf{G}^{(i,j)}\mathbf{K}^{(i,j)}, \tag{3.53}$$

où $\mathbf{K}^{(i,j)}$ est appelée matrice symétrique hermitienne hyperbolique et est définie de la manière suivante :

$$\mathbf{K}^{(i,j)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & c_h^{(i,j)} & \ddots & \bar{s_h}^{(i,j)} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & s_h^{(i,j)} & \ddots & {c_h}^{(i,j)} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \tag{3.54}$$

où $c_h^{(i,j)}$ un paramètre réel et ${s_h}^{(i,j)}$ un paramètre complexe tous deux liés par la relation $|{c_h}^{(i,j)}|^2 - |{s_h}^{(i,j)}|^2 = 1$ dans le cas complexe.

La paramétrisation de la matrice $\mathbf{X}^{(i,j)}$ avec de telles décompositions matricielles a donné lieu à un type de méthodes résumé dans la table algorithmique 5 dans le cas d'une factorisation LU. Dans [125], une méthode basée sur une décomposition QL de la matrice $\mathbf{X}^{(i,j)}$ est proposée pour le problème de DCC dans le cadre de données complexes. Dans [126] et [127] des méthodes basées sur une factorisation LU ont été proposées dans le cadre de la diagonalisation conjointe d'un ensemble de matrices symétriques hermitiennes et dans dans le cadre de la diagonalisation conjointe d'un ensemble de matrices symétriques complexes, la matrice $\mathbf{X}^{(i,j)}$ est paramétrisée avec une factorisation LU. Enfin, Une méthode basée sur une factorisation QR est implémentée dans [128] pour diagonaliser un ensemble de matrices symétriques à valeurs réelles.

Algorithme 5 Forme générique des algorithmes de diagonalisation conjointe basés sur une stratégie locale et estimant séparément les paramètres de la matrice $\mathbf{X}^{(i,j)}$ (exemple avec une factorisation LU)

Soit S_C un critère d'arrêt préalablement défini et It_{max} le nombre d'itérations maximal ;

Initialisation de $\mathbf{B}$ avec la matrice identité ou par n'importe quel choix judicieux ;
$it \leftarrow 1$;
while S_C est $faux$ et $it < It_{max}$ **do**
  for $i = 1$ to $R-1$ **do**
    for $j = i+1$ to R **do**
      Estimer $U^{(i,j)}$;
      Mettre à jour $\mathbf{B} \leftarrow \mathbf{U}^{(i,j)}\mathbf{B}$;
      Mettre à jour $\mathbf{N}^{(k)} \leftarrow \mathbf{U}^{(i,j)}\mathbf{N}^{(k)} \left(\mathbf{U}^{(i,j)}\right)^{\ddagger}$, $\forall k \in [1,K]_{\mathbb{N}}$;
      Estimer $L^{(i,j)}$;
      Mettre à jour $\mathbf{B} \leftarrow \mathbf{L}^{(i,j)}\mathbf{B}$;
      Mettre à jour $\mathbf{N}^{(k)} \leftarrow \mathbf{L}^{(i,j)}\mathbf{N}^{(k)} \left(\mathbf{L}^{(i,j)}\right)^{\ddagger}$, $\forall k \in [1,K]_{\mathbb{N}}$;
    end for
  end for
  Mettre à jour S_C ;
  $it \leftarrow it + 1$;
end while

Remarque : dans les trois dernières références citées, les paramètres de $\mathbf{X}^{(i,j)}$ sont estimés simultanément. Les algorithmes proposés correspondent donc à la table algorithmique 3.

3.3 État de l'art des méthodes de diagonalisation conjointe par similitude

Nos travaux portent sur la DCS (voir équation (3.4)). Ce problème de diagonalisation conjointe est utile pour de nombreuses applications en traitement du signal, comme l'estimation de direction d'arrivée [129], l'estimation d'angles et de retard conjointe [130], la récupération d'harmoniques multi-dimensionnelles [131], l'Analyse en Composantes Indépendantes (ACI) [50] et bien sûr la décomposition CP [22, 38, 39].

Pour les trois premières applications citées, seules les valeurs propres sont utiles. Les méthodes alors présentées ne sont pas des méthodes de diagonalisation conjointe de matrices à proprement parlé, mais des méthodes de triangularisation conjointe basées sur une décomposition de Schur conjointe (voir sous section 1.4.4). En effet, en écrivant la décomposition QR de la matrice de vecteurs propres $\mathbf{A}$, nous obtenons (dans le cas réel)

$$\mathbf{M}^{(k)} = \mathbf{Q}\mathbf{R}\mathbf{D}^{(k)}\mathbf{R}^{-1}\mathbf{Q}^{T}, \quad \forall k \in [1;K]_{\mathbb{N}} \tag{3.55}$$

où $\mathbf{R}$ est une matrice triangulaire inférieure ou supérieure ne contenant que des 1 sur sa diagonale. Ainsi une fois la matrice orthogonale (ou unitaire dans le cas complexe) $\mathbf{Q}$

estimée par triangularisation, l'ensemble de matrices $\mathbf{M}^{(k)}$ est transformé en

$$\mathbf{T}^{(k)} = \mathbf{R}\mathbf{D}^{(k)}\mathbf{R}^{-1} \quad \forall k \in [1; K]_{\mathbb{N}}, \tag{3.56}$$

où les matrices matrices $\mathbf{T}^{(k)}$ sont par conséquent des matrices triangulaires. Les valeurs propres sont donc directement accessibles car $\mathbf{D}^{(k)} = \text{diag}\{\mathbf{T}^{(k)}\}$, $\forall\ k \in [1; K]_{\mathbb{N}}$.

Nous allons maintenant présenter les méthodes de DCS existantes. Nous les différencierons par rapport à la factorisation matricielle utilisée : décomposition polaire ou LU. Chacun des algorithmes de DCS de la littérature a été décliné en deux versions, l'une pour diagonaliser des ensembles de matrices réelles et l'autre pour diagonaliser des ensembles de matrices complexes. En effet, les versions réelles ne sont pas directement applicables au cas complexe, là réside une des principales faiblesses des algorithmes présentés ici.

Dans le cas réel, le même niveau de performances est globalement atteint en ce qui concerne la précision de l'estimation de la matrice diagonalisante peu importe la décomposition matricielle utilisée. En revanche, la stratégie de résolution des algorithmes basés sur une factorisation LU permet de diminuer le coût calcul et donc d'estimer plus rapidement la matrice diagonalisante.

Dans le cas complexe, les algorithmes de DCS estiment les parties réelles et imaginaires des paramètres à estimer. Cette estimation se fait soit de manière non conjointe, soit de manière conjointe par minimisation d'un polynôme de degré élevé. Dans les deux cas, les performances des algorithmes se retrouvent dégradées. Les algorithmes basés sur une décomposition LU estiment mieux la matrice diagonalisante lorsque le niveau de bruit est faible. À partir d'un certain niveau de bruit, les algorithmes basés sur une décomposition polaire permettent d'obtenir une meilleure estimation de la matrice diagonalisante. Le coût de calcul de ces derniers est cependant systématiquement très élevé, car ils nécessitent beaucoup plus d'itérations pour converger que dans le cas réel.

Dans [132], une analyse de perturbation a été présentée pour différents algorithmes de DCS.

3.3.1 Algorithmes de DCS basés sur une décomposition polaire

Les algorithmes de DCS de la littérature basés sur une décomposition polaire sont tous de la forme présentée à la table algorithmique 5. Ainsi, dans le cas réel, lors d'une itération les matrices $\mathbf{N}^{(k)}$ sont mises à jour $R(R-1)/2$ fois alternativement par une matrice symétrique hyperbolique $\mathbf{H}^{(i,j)}$ puis par une matrice de Givens $\mathbf{Q}^{(i,j)}$ de la manière suivante :

$$\mathbf{N}^{(k)} \leftarrow \mathbf{H}^{(i,j)}\mathbf{N}^{(k)}\mathbf{H}^{(i,j)-1} \quad \forall k \in [1; K]_{\mathbb{N}} \tag{3.57}$$

et

$$\mathbf{N}^{(k)} \leftarrow \mathbf{Q}^{(i,j)}\mathbf{N}^{(k)}\mathbf{Q}^{(i,j)T} \quad \forall k \in [1; K]_{\mathbb{N}}. \tag{3.58}$$

Dans le cas complexe, nous utilisons les matrices symétriques hermitiennes hyperboliques $\mathbf{K}^{(i,j)}$ et les matrices de rotation de Givens unitiaires $\mathbf{G}^{(i,j)}$. Nous avons alors

$$\mathbf{N}^{(k)} \leftarrow \mathbf{K}^{(i,j)}\mathbf{N}^{(k)}\mathbf{K}^{(i,j)-1} \quad \forall k \in [1; K]_{\mathbb{N}} \tag{3.59}$$

et

$$\mathbf{N}^{(k)} \leftarrow \mathbf{G}^{(i,j)}\mathbf{N}^{(k)}\mathbf{G}^{(i,j)H}\ \forall k \in [1; K]_{\mathbb{N}}. \tag{3.60}$$

Pour simplifier, nous expliquerons le fonctionnement global de ces algorithmes dans le cas réel.

Le premier algorithme de DCS à avoir été proposé est SH-RT [133] (de l'anglais *SHear-RoTation algorithm*). Dans cet algorithme la matrice $\mathbf{H}^{(i,j)}$ est calculée en minimisant la norme des termes hors diagonaux d'une seule matrice :

$$||\mathsf{ZDiag}\{\mathbf{N}^{(r)}\}||_F^2. \tag{3.61}$$

où l'entier k est choisi de la manière suivante :

$$r = \underset{1\leq k\leq K}{\mathsf{argmax}}|N_{i,i}^{(k)} - N_{j,j}^{(k)}| \tag{3.62}$$

La matrice $\mathbf{Q}^{(i,j)}$ est quant à elle estimée en minimisant $C_{\text{inverse}}(\mathbf{Q}^{(i,j)})$ défini à l'équation (3.14).

Le second algorithme de DCS à avoir été proposé se nomme JUST [134] (de l'anglais *Joint Unitary Shear Transformation*). Cet algorithme résout le problème en minimisant alternativement les critères $C_{\text{inverse}}(\mathbf{H}^{(i,j)})$ et $C_{\text{inverse}}(\mathbf{Q}^{(i,j)})$. Enfin, le dernier algorithme basé sur une décomposition polaire a avoir été proposé se nomme JDTM [22] (pour *Joint Diagonalization algorithm based on Targeting hyperbolic Matrices*). Dans cet algorithme la matrice de rotation de Givens est estimée en minimisant $C_{\text{inverse}}(\mathbf{Q}^{(i,j)})$. La matrice hyperbolique est estimée, comme le nom de la méthode l'indique, en ciblant seulement les termes hors diagonaux $N_{i,j}^{(k)}$ et $N_{j,i}^{(k)}$. Ainsi le critère utilisé pour estimer les matrices $\mathbf{H}^{(i,j)}$ est $C'_{\text{inverse}}(\mathbf{H}^{(i,j)})$ défini à l'équation (3.28), le principe est de se ramener à un problème de décomposition en éléments propres d'une matrice de dimension 2 comme dans [121] pour le problème de DCC.

Nous pouvons noter que l'estimation des matrices de Givens de JUST et JDTM est identique à celle de l'algorithme JADE [53] pour le problème de DCC-U.

3.3.2 Algorithmes de DCS basés sur une factorisation LU

Comme pour les méthodes QR, les algorithmes de DCS utilisant une factorisation LU sont basés sur la factorisation de la matrice diagonalisante $\mathbf{B}$. La première étape consiste donc à trianguläriser l'ensemble de matrices à diagonaliser en estimant une matrice triangulaire inférieure décomposée en matrices élémentaires (c.f équation (3.33)) de manière itérative. L'avantage est de pouvoir estimer la matrice triangulaire supérieure sans processus itératif, ce qui permet de réduire le coût de calcul de l'algorithme. Pour se faire, une méthode nommée JET [50] (pour *Joint Eigenvalue decomposition algorithm based on Triangular matrices*) a été déclinée en deux versions. La première nommée JET-O dont l'estimation des matrices $\mathbf{L}^{(i,j)}$ est obtenue en minimisant le critère de triangularisation

$$C_{\text{triang}}(\mathbf{L}^{(i,j)}) = \sum_{k=1}^{K}\sum_{q=1}^{R-1}\sum_{p=q+1}^{R}(T_{p,q}^{(k)})^2 \tag{3.63}$$

avec

$$\mathbf{T}^{(k)} = \mathbf{L}^{(i,j)}\mathbf{N}^{(k)}\mathbf{L}^{(i,j)-1}\ \forall k \in [1;K]_{\mathbb{N}}. \tag{3.64}$$

La seconde, nommée JET-U, estime la matrice $\mathbf{L}^{(i,j)}$ en minimisant le critère simplifié

$$C'_{\text{triang}}(\mathbf{L}^{(i,j)}) = \sum_{k=1}^{K} (T_{i,j}^{(k)})^2. \tag{3.65}$$

Le fait de minimiser (3.65) plutôt que (3.63) permet de réduire la complexité numérique de l'algorithme d'un facteur 2 et donc d'augmenter sa vitesse de convergence. Pour les deux algorithmes, à la fin du processus itératif, les matrices $\mathbf{T}^{(k)}$ doivent être idéalement triangulaires supérieures, car

$$\mathbf{T}^{(k)} = \mathbf{U}\mathbf{D}^{(k)}\mathbf{U}^{-1}, \tag{3.66}$$

où $\mathbf{U}$ est une matrice triangulaire supérieure. Les valeurs propres d'une matrice triangulaire sont égales aux éléments de sa diagonale donc

$$\mathbf{D}^{(k)} = \mathsf{Diag}\{\mathbf{T}^{(k)}\} \quad \forall k \in [1;K]_{\mathbb{N}}. \tag{3.67}$$

Ainsi nous avons $\forall k \in [1;K]_{\mathbb{N}}$ et $\forall (i,j) \in [1;R]_{\mathbb{N}}$ avec $i < j$

$$(T_{j,j}^{(k)} - T_{i,i}^{(k)})U_{i,j} = \sum_{p=i+1}^{j} U_{p,j} T_{i,p}^{(k)}. \tag{3.68}$$

En posant

$$a_k^{(i,j)} = T_{j,j}^{(k)} - T_{i,i}^{(k)} \quad \text{et} \quad b_k^{(i,j)} = \sum_{p=i+1}^{j} U_{p,j} T_{i,p}^{(k)}, \tag{3.69}$$

l'équation (3.68) peut se réécrire comme

$$U_{i,j}\mathbf{a}^{(i,j)} = \mathbf{b}^{(i,j)}. \tag{3.70}$$

Finalement les éléments de la matrice $\mathbf{U}$ sont données au sens des moindres carrées par

$$U_{i,j} = \frac{\mathbf{a}^{(i,j)H}\mathbf{b}^{(i,j)}}{||\mathbf{a}^{(i,j)}||^2}. \tag{3.71}$$

3.3.3 Complexité numérique

Nous définissons la complexité numérique Γ d'un algorithme itératif comme le nombre de multiplications que ce dernier effectue à chaque itération. L'étape la plus coûteuse lors d'une itération étant généralement les $R(R-1)/2$ mises à jour de l'ensemble de matrices

$$\mathbf{N}^{(k)} \leftarrow \mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1}. \tag{3.72}$$

Cette étape est en $O(KR^3)$. Nous négligeons ici les complexités des étapes d'ordre de grandeur plus faible. Nous avons alors, dans le cas réel

$$\Gamma[\text{SH-RT}] \simeq 8KR^3 \tag{3.73}$$
$$\Gamma[\text{JUST}] \simeq 11KR^3 \tag{3.74}$$
$$\Gamma[\text{JDTM}] \simeq 8KR^3 \tag{3.75}$$
$$\Gamma[\text{JET-O}] \simeq 2KR^3 \tag{3.76}$$
$$\Gamma[\text{JET-U}] \simeq KR^3. \tag{3.77}$$

Nous considérons qu'une multiplication complexe implique 4 multiplications réelles (même si on pourrait se contenter de 3 multiplications). Ainsi il suffit de multiplier par 4, l'approximation proposée de la complexité des algorithmes dans le cas réel pour avoir leur complexité dans leur version complexe. Les algorithmes JET sont les moins coûteux, ceci est principalement dû au fait que le processus itératif des algorithmes JET n'estime qu'un seul paramètre à chaque mise à jour de l'ensemble de matrices. Ainsi, nous considérons le coût de calcul total d'un algorithme comme la multiplication entre le nombre d'itérations dont l'algorithme a besoin pour atteindre un point stationnaire (lorsque la fonction de coût à minimiser n'évolue plus au cours des itérations) et la complexité numérique définie ici.

3.4 Bilan du chapitre

Dans ce chapitre, nous avons présenté le problème de diagonalisation conjointe de matrices de manière générale. Nous avons ensuite introduit différentes méthodes de résolution en nous focalisant sur les méthodes de mises à jour multiplicatives de la matrice diagonalisante $\mathbf{B}$. Grâce à l'indétermination d'échelle inhérent aux problèmes de diagonalisation conjointe de matrices, la matrice de mise à jour $\mathbf{X}$ dépend de $R(R-1)$ paramètres. Nous avons alors pu distinguer quatre types d'algorithmes :

- les algorithmes correspondant à la table algorithmique 2, estimant la matrice $\mathbf{X}$ de manière globale à chaque itération,
- les algorithmes correspondant à la table algorithmique 3, estimant $R(R-1)/2$ matrices $\mathbf{X}^{(i,j)}$ à chaque à itération en calculant simultanément les paramètres de chacune d'entre elles,
- les algorithmes correspondant à la table algorithmique 4, estimant la matrice $\mathbf{X}$ de manière locale en la paramétrant avec des matrices ne dépendant que d'un seul paramètre provenant des factorisations matricielles classiques,
- les algorithmes correspondant à la table algorithmique 5, estimant $R(R-1)/2$ matrices $\mathbf{X}^{(i,j)}$ en les paramétrant à l'aide des factorisations matricielles classiques et estimant séparément chacun des paramètres de ces dernières.

Enfin, nous nous sommes focalisés sur l'état de l'art des différentes méthodes de diagonalisation conjointe de matrices par similitude. Nous présentons dans le tableau 3.1 un récapitulatif des différents problèmes de diagonalisation conjointe présentés dans ce chapitre ainsi que les principales méthodes permettant de les résoudre. Dans le chapitre 4, nous présenterons les travaux que nous avons effectués pour développer de nouvelles méthodes de DCS. Notre but est de développer des algorithmes ayant un faible coût calcul tout en étant à la fois précis et robustes dans le cas d'une DCS de matrices à valeurs complexes. Dans le chapitre 5 nous proposerons des méthodes de DCS sous contrainte de non-négativité dans le cadre de la réécriture du problème de décomposition CP non-négative.

Problème	Propriétés des matrices $\mathbf{D}^{(k)}$	Propriétés de la matrice $\mathbf{A}$	Propriétés des matrices $\mathbf{M}^{(k)}$	Algorithmes
$\mathbf{AD}^{(k)}\mathbf{A}^T$	réelles	réelle	réelles symétriques	U-WEDGE [106], FFDIAG [117], LUJ1D ,QRJ1D [109], J-DI [121]
	complexes/réelles	complexe	complexes symétriques	NOODLES [119], AC-DC [104]
	réelles	réelle orthogonale (*i.e.* $\mathbf{AA}^T = I$)	réelles symétriques	JADE [53], GAEX [110]
	réelles à valeurs positives	réelle	réelles symétriques définies positives	JD-BGL [114]
$\mathbf{AD}^{(k)}\mathbf{A}^H$	complexes	complexe	complexes	DIEM [103], CVFFDIAG [118], NOODLES [119], FAJD [135], AC-DC [104], CJDI [95], SL [125]
	réelles	complexe	complexes symétriques hermitiennes	JADE [53], EJD [102]
	complexes	complexe unitaire (*i.e.* $\mathbf{AA}^H = I$)	complexes	SOBI [93]
	réelles à valeurs positives	complexe	complexes symétriques hermitiennes définies positives	JD-BGL [114]
$\mathbf{AD}^{(k)}\mathbf{A}^{-1}$	réelles/complexes	réelle/complexe	réelles/complexes	SH-RT [133], JUST [134], JDTM [22], JET [50]

TABLE 3.1: Tableau récapitulatif des différents problèmes de diagonalisation conjointe.

Chapitre 4

Nouveaux algorithmes de diagonalisation conjointe de matrices par similitude

Comme nous l'avons vu au chapitre précédent, les méthodes de DCS de la littérature sont déclinées en deux versions : une pour traiter des données complexes et une pour traiter des données réelles. Les versions complexes des algorithmes de DCS sont généralement peu robustes au bruit ou alors très couteuse en terme de coût calcul total. Nous proposons ici deux familles de méthodes de DCS (déclinées en plusieurs algorithmes) fonctionnant à la fois dans le cas réel et complexe sans modifications. Par conséquent, dans tout ce chapitre, nous travaillons dans $\mathbb{C}$. De plus, dans le but d'avoir une expression analytique des paramètres à estimer, nous faisons l'hypothèse suivante :

Hypothèse 1. *Nous sommes au voisinage d'un point stationnaire et de la solution diagonalisante.*

Dans le contexte des algorithmes basés sur une mise à jour multiplicative (voir équation (3.22)), un point stationnaire se traduit par $\mathbf{X} = \mathbf{I}_R$. Ainsi, la première partie de l'hypothèse 1 induit

$$||\mathsf{ZDiag}\{\mathbf{X}\}||_F \ll 1. \tag{4.1}$$

La seconde partie quant à elle induit

$$||\mathsf{ZDiag}\{\mathbf{N}^{(k)}\}||_F \ll 1, \ \forall k \in [1, K]_{\mathbb{N}}. \tag{4.2}$$

Cette hypothèse dépend de la puissance du bruit et du point initial choisi pour la matrice diagonalisante $\mathbf{B}$. Une manière judicieuse d'initialiser la matrice diagonalisante est de prendre le résultat de la GEVD de deux matrices de l'ensemble à diagonaliser afin de se ramener dans le cadre de l'hypothèse 1. Toutefois, nous verrons que cela n'est pas toujours nécessaire.

4.1 Méthode basée sur un développement de Taylor de la matrice de mise à jour

Dans cette section, nous présentons une méthode pour estimer la matrice de mise à jour à l'aide de la paramétrisation $(\mathbf{I}_R + \mathbf{Z})$ présentée dans l'équation (3.24). Cette méthode, inspirée de [119], a engendré deux algorithmes : le premier pour lequel cette paramétrisation est appliquée à la matrice $\mathbf{X}$ et le second pour lequel elle est appliquée à la matrice $\mathbf{X}^{(i,j)}$.

4.1.1 Stratégie de résolution globale

Nous proposons ici un algorithme correspondant à la table algorithmique 2 (page 61). L'idée est de paramétrer la matrice de mise à jour $\mathbf{X}$ de la manière suivante

$$\mathbf{X} = (\mathbf{I}_R + \mathbf{Z}), \tag{4.3}$$

où $\mathbf{I} \in \mathbb{C}^{R\times R}$ est la matrice identité et $\mathbf{Z} \in \mathbb{C}^{R\times R}$ est une matrice ne contenant que des zéros sur sa diagonale. Cette paramétrisation est possible grâce à l'indétermination d'échelle du problème de DCS. Nous utilisons ici le critère inverse de diagonalisation (3.23). Ainsi en remplaçant l'expression de $\mathbf{X}$ dans ce critère, nous obtenons

$$C_{\mathbf{I}+\mathbf{Z}}(\mathbf{Z}) = \sum_{k=1}^{K} ||(\mathbf{I}_R + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I}_R + \mathbf{Z})^{-1}||_F^2. \tag{4.4}$$

Le terme $(\mathbf{I}_R+\mathbf{Z})^{-1}$ rend le problème d'optimisation difficile à résoudre analytiquement. En prenant en compte la première partie de l'hypothèse 1, l'équation (4.1) est alors équivalente à $||\mathbf{Z}||_F \ll 1$, puis en utilisant le développement de Taylor de $(\mathbf{I}_R + \mathbf{Z})^{-1}$ tronqué à l'ordre 1, nous pouvons faire l'approximation suivante :

$$(\mathbf{I}_R + \mathbf{Z})^{-1} \simeq \mathbf{I}_R - \mathbf{Z}. \tag{4.5}$$

Le critère à minimiser est alors de la forme

$$C_{\mathbf{I}_R+\mathbf{Z}}(\mathbf{Z}) \simeq \sum_{k=1}^{K} ||(\mathbf{I}_R + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I}_R - \mathbf{Z})||_F^2. \tag{4.6}$$

En développant et en négligeant les termes d'ordre supérieur à 1 en $\mathbf{Z}$, nous obtenons :

$$\begin{aligned} \forall k \in [1;K]_{\mathbb{N}},\ (\mathbf{I}_R + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I}_R - \mathbf{Z}) &= \mathbf{N}^{(k)} + \mathbf{Z}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z} - \mathbf{Z}\mathbf{N}^{(k)}\mathbf{Z} \\ &\simeq \mathbf{N}^{(k)} + \mathbf{Z}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z}. \end{aligned} \tag{4.7}$$

Nous décomposons à leur tour les matrices $\mathbf{N}^{(k)}$ comme la somme d'une matrice diagonale et d'une matrice dont les termes diagonaux sont nuls :

$$\forall k \in [1;K]_{\mathbb{N}},\ \mathbf{N}^{(k)} = \mathbf{\Lambda}^{(k)} + \mathbf{O}^{(k)}, \tag{4.8}$$

où $\mathbf{\Lambda}^{(k)} = \mathsf{Diag}\{\mathbf{N}^{(k)}\}$ et $\mathbf{O}^{(k)} = \mathsf{ZDiag}\{\mathbf{N}^{(k)}\}$. En se basant maintenant sur la deuxième partie de l'hypothèse 1, l'équation (4.2) est alors équivalente à $||\mathbf{O}^{(k)}||_F \ll 1$, $\forall k \in [1;K]_{\mathbb{N}}$.

Ainsi en négligeant encore une fois les termes d'ordre supérieur à 1, l'expression (4.7) devient alors :

$$\forall k \in [1;K]_{\mathbb{N}},\ \mathbf{N}^{(k)} + \mathbf{Z}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z} \simeq \mathbf{\Lambda}^{(k)} + \mathbf{O}^{(k)} + \mathbf{Z}\mathbf{\Lambda}^{(k)} - \mathbf{\Lambda}^{(k)}\mathbf{Z}. \tag{4.9}$$

Puis, en injectant l'expression (4.9) dans (4.6), le critère à minimiser est approximé par :

$$C_a(\mathbf{Z}) = \sum_{k=1}^{K} ||\mathsf{ZDiag}\left\{\mathbf{O}^{(k)} + \mathbf{Z}\mathbf{\Lambda}^{(k)} - \mathbf{\Lambda}^{(k)}\mathbf{Z}\right\}||_F^2. \tag{4.10}$$

Enfin, en développant (4.10), nous obtenons

$$C_a(\mathbf{Z}) = \sum_{\substack{m,n=1 \\ m\neq n}}^{N} f_{m,n}(Z_{m,n}) \tag{4.11}$$

avec

$$f_{m,n}(Z_{m,n}) = \sum_{k=1}^{K} |O_{m,n}^{(k)} + Z_{m,n}(\Lambda_{n,n}^{(k)} - \Lambda_{m,m}^{(k)})|^2. \tag{4.12}$$

Puisque les $f_{m,n}(Z_{m,n})$ sont indépendantes $\forall m,n$, minimiser le critère $C_a(\mathbf{Z})$ est équivalent à minimiser chaque $f_{m,n}(Z_{m,n})$ indépendamment. L'expression de la dérivée de $f_{m,n}(Z_{m,n})$ par rapport à $\bar{Z}_{m,n}$, avec $m \neq n$, est alors

$$\forall m,n\ m \neq n\ \frac{\partial f_{m,n}(Z_{m,n})}{\partial \bar{Z}_{m,n}} = \sum_{k=1}^{K} (\bar{\Lambda}_{n,n}^{(k)} - \bar{\Lambda}_{m,m}^{(k)})(O_{m,n}^{(k)} + Z_{m,n}(\Lambda_{n,n}^{(k)} - \Lambda_{m,m}^{(k)})). \tag{4.13}$$

La valeur optimale de $Z_{m,n}$ est alors obtenue en annulant $\frac{\partial f_{m,n}(Z_{m,n})}{\partial \bar{Z}_{m,n}}$, soit :

$$Z_{m,n} = \frac{\sum_{k=1}^{K} (\bar{\Lambda}_{m,m}^{(k)} - \bar{\Lambda}_{n,n}^{(k)}) O_{m,n}^{(k)}}{\sum_{k=1}^{K} |\Lambda_{m,m}^{(k)} - \Lambda_{n,n}^{(k)}|^2}. \tag{4.14}$$

Nous pouvons noter que dans le cas réel, $Z_{m,n}$ a pour expression

$$Z_{m,n} = \frac{\sum_{k=1}^{K} (\Lambda_{m,m}^{(k)} - \Lambda_{n,n}^{(k)}) O_{m,n}^{(k)}}{\sum_{k=1}^{K} (\Lambda_{m,m}^{(k)} - \Lambda_{n,n}^{(k)})^2}. \tag{4.15}$$

L'implémentation dans le cas complexe ne nécessite pas de modification pour traiter des données réelles. Nous appelons cet algorithme JDTE pour *Joint Diagonalization based on Taylor Expansion.*

Complexité numérique : la matrice de mise à jour étant ici une matrice de taille R dont les termes hors-diagonaux peuvent être différents de 0 ou 1 et dont les termes diagonaux sont égaux à 1, nous obtenons :

$$\Gamma[JDTE] \simeq 2KR^3 \tag{4.16}$$

dans le cas réel et donc

$$\Gamma[JDTE] \simeq 8KR^3 \tag{4.17}$$

dans le cas complexe.

4.1.2 Stratégie de résolution locale

Dans le but de proposer un algorithme de balayage par paire correspondant au type d'algorithme présenté à la table algorithmique 3 du chapitre précédent (page 62), nous effectuons le même raisonnement en paramétrant cette fois-ci les matrices $\mathbf{X}^{(i,j)}$ comme

$$\mathbf{X}^{(i,j)} = \mathbf{I}_R + \mathbf{Z}^{(i,j)} \tag{4.18}$$

où les entrées de $\mathbf{Z}^{(i,j)}$ sont toutes nulles exceptées pour les termes $Z_{i,j}^{(i,j)}$ et $Z_{j,i}^{(i,j)}$. Nous choisissons maintenant de minimiser un critère du type (3.28), la fonction de coût de notre algorithme s'écrit donc

$$\begin{aligned} C'_{\mathbf{I}_R+Z^{(i,j)}}(\mathbf{Z}^{(i,j)}) = \sum_{k=1}^{K} & |((\mathbf{I}_R + \mathbf{Z}^{(i,j)})\mathbf{N}^{(k)}(\mathbf{I}_R - \mathbf{Z}^{(i,j)}))_{i,j}|^2 + \\ & |((\mathbf{I}_R + \mathbf{Z}^{(i,j)})\mathbf{N}^{(k)}(\mathbf{I}_R - \mathbf{Z}^{(i,j)}))_{j,i}|^2. \end{aligned} \tag{4.19}$$

Ainsi, en appliquant la même méthode que précédemment, la fonction de coût à minimiser approximée devient

$$C'_a(\mathbf{Z}^{(i,j)}) = f_{i,j}(Z_{i,j}^{(i,j)}) + f_{j,i}(Z_{j,i}^{(i,j)}). \tag{4.20}$$

Finalement, l'expression des deux termes inconnus de la matrice $\mathbf{Z}^{(i,j)}$ est donnée par :

$$\begin{cases} Z_{i,j}^{(i,j)} = \dfrac{\sum_{k=1}^{K}(\bar{\Lambda}_{i,i}^{(k)} - \bar{\Lambda}_{j,j}^{(k)})O_{i,j}^{(k)}}{\sum_{k=1}^{K}|\Lambda_{i,i}^{(k)} - \Lambda_{j,j}^{(k)}|^2} \\ Z_{j,i}^{(i,j)} = \dfrac{\sum_{k=1}^{K}(\bar{\Lambda}_{j,j}^{(k)} - \bar{\Lambda}_{i,i}^{(k)})O_{j,i}^{(k)}}{\sum_{k=1}^{K}|\Lambda_{j,j}^{(k)} - \Lambda_{i,i}^{(k)}|^2}. \end{cases} \tag{4.21}$$

Nous appelons cette méthode SJDTE pour *Sweeped JDTE*.

Complexité numérique : la matrice de mise à jour étant décomposée comme $R(R-1)/2$ matrices de dimension R égales à la matrice identité à l'exception des termes $X_{i,j}^{(i,j)}$ et $X_{j,i}^{(i,j)}$, la complexité numérique de SJDTE est

$$\Gamma[SJDTE] \simeq 3KR^3 \tag{4.22}$$

dans le cas réel et donc

$$\Gamma[SJDTE] \simeq 12KR^3 \tag{4.23}$$

dans le cas complexe.

4.1.3 Discussion sur le calcul de Z et $\mathbf{Z}^{(i,j)}$

Il est important pour les algorithmes précédemment proposés de vérifier que le dénominateur de $Z_{m,n}$ dans (4.14) ou (4.15) et que le dénominateur de $Z_{j,i}^{(i,j)}$ et de $Z_{j,i}^{(i,j)}$ dans (4.21) ne puissent pas être égaux à zéro au cours des mises à jour *i.e.*

$$\forall(m,n),\ \sum_{k=1}^{K}|\Lambda_{m,m}^{(k)} - \Lambda_{n,n}^{(k)}|^2 \neq 0 \tag{4.24}$$

et

$$\sum_{k=1}^{K} |\Lambda_{i,i}^{(k)} - \Lambda_{j,j}^{(k)}|^2 \neq 0. \tag{4.25}$$

Nous pouvons remarquer qu'il suffit de montrer que (4.24) soit vraie pour que (4.25) soit vraie. Nous proposons donc ici une condition suffisante pour assurer (4.24) à chaque mise à jour.

Tout d'abord, nous introduisons la matrice

$$\mathbf{\Delta} = \begin{pmatrix} \Lambda_{1,1}^{(1)} & \cdots & \Lambda_{1,1}^{(K)} \\ \vdots & \cdots & \vdots \\ \Lambda_{R,R}^{(1)} & \cdots & \Lambda_{R,R}^{(K)} \end{pmatrix}. \tag{4.26}$$

Nous vérifions facilement que le dénominateur de $Z_{m,n}$, $\forall(m,n)\ m \neq n$ est non nul si et seulement si toutes les lignes de $\mathbf{\Delta}$ sont distinctes entre elles.

Lorsque $\mathbf{\Lambda}^{(k)} = \mathbf{D}^{(k)},\ \forall k \in [1,K]_{\mathbb{N}}$, il est intéressant de noter que cette condition est similaire à celle de l'unicité du problème de DCS. Nous établissons d'ailleurs le lien entre la matrice $\mathbf{\Delta}$ et la matrice $\mathbf{\Omega}$ définie dans l'équation (3.7) du chapitre 3. Pour cela nous définissons la matrice $\mathbf{G}$ comme

$$\mathbf{G} = \widehat{\mathbf{B}}\mathbf{A}, \tag{4.27}$$

où $\widehat{\mathbf{B}}$ est l'estimation de la matrice diagonalisante $\mathbf{B}$ suite à une mise à jour. Ainsi à chaque mise à jour, l'ensemble de matrices $\mathbf{N}^{(k)}$ s'écrit

$$\forall k \in [1,K]_{\mathbb{N}},\ \mathbf{N}^{(k)} = \mathbf{G}\mathbf{D}^{(k)}\mathbf{G}^{-1}. \tag{4.28}$$

Soit diag$\{\bullet\}$, le vecteur colonne contenant la diagonale de la matrice en argument. Le vecteur colonne contenant la diagonale de $\mathbf{N}^{(k)}$ peut alors être écrit comme

$$\text{diag}\left\{\mathbf{N}^{(k)}\right\} = (\mathbf{G} \boxdot \mathbf{G}^{-T})\ \text{diag}\left\{\mathbf{D}^{(k)}\right\}. \tag{4.29}$$

En posant $\mathbf{H} = \mathbf{G} \boxdot \mathbf{G}^{-T}$, nous avons finalement

$$\mathbf{\Delta} = \mathbf{H}\mathbf{\Omega} \tag{4.30}$$

car $\mathbf{\Delta} = [\text{diag}\left\{\mathbf{N}^{(1)}\right\} \ldots \text{diag}\left\{\mathbf{N}^{(K)}\right\}]$ et $\mathbf{\Omega} = [\text{diag}\left\{\mathbf{D}^{(1)}\right\} \ldots \text{diag}\left\{\mathbf{D}^{(K)}\right\}]$.

La différence entre deux lignes m et n de $\mathbf{\Delta}$ s'écrit

$$\mathbf{\Delta}_{m\bullet} - \mathbf{\Delta}_{n\bullet} = (\mathbf{H}_{m\bullet} - \mathbf{H}_{n\bullet})\mathbf{\Omega}, \tag{4.31}$$

où $\mathbf{\Delta}_{m\bullet}$, $\mathbf{\Delta}_{n\bullet}$, $\mathbf{H}_{m\bullet}$ et $\mathbf{H}_{n\bullet}$ sont respectivement la $m^{\text{ème}}$ et la $n^{\text{ème}}$ lignes des matrices $\mathbf{\Delta}$ et $\mathbf{H}$. Alors,

$$\mathbf{\Delta}_{m\bullet} - \mathbf{\Delta}_{n\bullet} \neq \mathbf{0} \Leftrightarrow\ \mathbf{\Omega}^T(\mathbf{H}_{m\bullet} - \mathbf{H}_{n\bullet})^T \neq \mathbf{0}. \tag{4.32}$$

Ainsi il suffit à la fois que $\mathbf{H}_{m\bullet} - \mathbf{H}_{n\bullet} \neq \mathbf{0}$ et que $(\mathbf{H}_{m\bullet} - \mathbf{H}_{n\bullet})^T$ ne soient pas dans le noyau de $\mathbf{\Omega}^T\ \forall(m,n),\ m \neq n$.

D'un point de vue pratique, il serait peu probable que la première des deux conditions ci-dessus ne soit pas respectée. En effet, la matrice $\mathbf{G}$ et donc $\mathbf{H}$ sont sensées devenir de

plus en plus proches de la matrice identité (à une matrice diagonale et de permutation près).

Nous pouvons cependant donner une condition suffisante plus rigoureuse. En utilisant l'hypothèse 1, nous pouvons affirmer que $\mathbf{G}$ est proche d'une matrice diagonale (à une permutation près) dans le sens où elle est à diagonale strictement dominante, *i.e.*

$$\forall i, \quad |G_{i,i}| > \sum_{\substack{j=1 \\ j\neq i}}^{N} |G_{i,j}|. \tag{4.33}$$

Dans [136] (page 27), il est montré que l'inverse d'une matrice à diagonale dominante est aussi une matrice à diagonale dominante. Donc $\mathbf{G}^{-T}$ est aussi à diagonale dominante et par conséquent la matrice $\mathbf{H}$ l'est aussi (ce dernier point est renforcé par le produit d'Hadamard). Le terme $\mathbf{H}_{m\bullet} - \mathbf{H}_{n\bullet}$ ne peut donc pas être égal au vecteur nul pour tout (m, n) avec $m \neq n$.

Pour la seconde condition, il est suffisant de supposer que la matrice $\mathbf{\Omega}^T$ est de rang plein. En effet, dans ce cas là, le noyau de l'application (4.32) est l'ensemble vide. Cette condition correspond à la condition d'unicité du problème de DCC. Rigoureusement, nous avons donc une condition suffisante plus contraignante que la condition d'unicité de la DCS.

4.2 Méthode basée sur une estimation simultanée des paramètres de mise à jour

Dans cette section nous proposons une classe d'algorithme de DCS correspondant au type d'algorithme présenté dans la table algorithmique 3 du chapitre précédent (page 62).

Le critère choisi ici est le critère (3.32), nous permettant de réécrire notre problème comme une série de problèmes de dimension 2. Nous rappelons ici l'expression de ce critère :

$$C'_{\text{inverse}}(\tilde{\mathbf{X}}^{(i,j)}) = \sum_{k=1}^{K} \left| \left(\tilde{\mathbf{X}}^{(i,j)} \tilde{\mathbf{N}}^{(k)} \tilde{\mathbf{X}}^{(i,j)-1} \right)_{1,2} \right|^2 + \left| \left(\tilde{\mathbf{X}}^{(i,j)} \tilde{\mathbf{N}}^{(k)} \tilde{\mathbf{X}}^{(i,j)-1} \right)_{2,1} \right|^2. \tag{4.34}$$

Les algorithmes proposés ici ont tous la même structure qui peut être déclinée en plusieurs versions selon la paramétrisation de la matrice $\tilde{\mathbf{X}}^{(i,j)}$. Dans le but de simplifier les notations, nous noterons $\tilde{\mathbf{X}}$ la matrice $\tilde{\mathbf{X}}^{(i,j)}$, dont les éléments seront notés

$$\tilde{\mathbf{X}} = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}. \tag{4.35}$$

4.2.1 Structure générale des algorithmes

Grâce à l'indétermination d'échelle de notre problème, nous pouvons imposer $\det\{\tilde{\mathbf{X}}\} = 1$. Nous avons donc

$$\tilde{\mathbf{X}}^{-1} = \begin{pmatrix} x_4 & -x_2 \\ -x_3 & x_1 \end{pmatrix}. \tag{4.36}$$

Soient les matrices $\tilde{\mathbf{N}}'^{(k)}$ définies par :

$$\tilde{\mathbf{N}}'^{(k)} = \tilde{\mathbf{X}}\tilde{\mathbf{N}}^{(k)}\tilde{\mathbf{X}}^{-1}, \tag{4.37}$$

les termes hors-diagonaux de $\tilde{\mathbf{N}}'^{(k)}$ peuvent donc être écrits :

$$\forall k \in [1,K]_{\mathbb{N}},\ \begin{cases} \tilde{N}'^{(k)}_{1,2} = x_1x_2(\tilde{N}^{(k)}_{2,2} - \tilde{N}^{(k)}_{1,1}) + x_1^2\tilde{N}^{(k)}_{1,2} - x_2^2\tilde{N}^{(k)}_{2,1}, \\ \tilde{N}'^{(k)}_{2,1} = x_3x_4(\tilde{N}^{(k)}_{1,1} - \tilde{N}^{(k)}_{2,2}) + x_4^2\tilde{N}^{(k)}_{2,1} - x_3^2\tilde{N}^{(k)}_{1,2}. \end{cases} \tag{4.38}$$

Définissons maintenant les deux vecteurs

$$\mathbf{v} = \begin{pmatrix} x_1x_2 \\ x_1^2 \\ x_2^2 \end{pmatrix};\ \mathbf{w} = \begin{pmatrix} x_3x_4 \\ x_4^2 \\ x_3^2 \end{pmatrix} \tag{4.39}$$

et les deux matrices

$$\mathbf{F}_{12} = \begin{pmatrix} \tilde{N}^{(1)}_{2,2} - \tilde{N}^{(1)}_{1,1} & \tilde{N}^{(1)}_{1,2} & -\tilde{N}^{(1)}_{2,1} \\ \vdots & \vdots & \vdots \\ \tilde{N}^{(K)}_{2,2} - \tilde{N}^{(K)}_{1,1} & \tilde{N}^{(K)}_{1,2} & -\tilde{N}^{(K)}_{2,1} \end{pmatrix} \tag{4.40}$$

et

$$\mathbf{F}_{21} = \begin{pmatrix} \tilde{N}^{(1)}_{1,1} - \tilde{N}^{(1)}_{2,2} & \tilde{N}^{(1)}_{2,1} & -\tilde{N}^{(1)}_{1,2} \\ \vdots & \vdots & \vdots \\ \tilde{N}^{(K)}_{1,1} - \tilde{N}^{(K)}_{2,2} & \tilde{N}^{(K)}_{2,1} & -\tilde{N}^{(K)}_{1,2} \end{pmatrix}. \tag{4.41}$$

Les deux termes composant la fonction de coût (4.34) peuvent ainsi être réécris comme

$$\begin{cases} \displaystyle\sum_{k=1}^{K} |\tilde{N}'^{(k)}_{1,2}|^2 = \mathbf{v}^H\mathbf{F}_{12}^H\mathbf{F}_{12}\mathbf{v}, \\ \displaystyle\sum_{k=1}^{K} |\tilde{N}'^{(k)}_{2,1}|^2 = \mathbf{w}^H\mathbf{F}_{21}^H\mathbf{F}_{21}\mathbf{w}. \end{cases} \tag{4.42}$$

Ces deux termes ne partagent aucune variable à optimiser et peuvent donc être minimisés séparément par rapport à $\mathbf{v}$ et $\mathbf{w}$. Les matrices $\mathbf{F}_{12}^H\mathbf{F}_{12}$ et $\mathbf{F}_{21}^H\mathbf{F}_{21}$ sont semi-définies positives par construction, nous pouvons ainsi résoudre chacun de ces deux problèmes en effectuant une décomposition en éléments propres sur les matrices $\mathbf{F}_{12}^H\mathbf{F}_{12}$ et $\mathbf{F}_{21}^H\mathbf{F}_{21}$ (comme expliqué dans [126]). En effet, il est bien connu que le vecteur de norme 1 minimisant une forme quadratique est le vecteur propre normalisé associé à la plus petite valeur propre de la matrice associée (ici $\mathbf{F}_{12}^H\mathbf{F}_{12}$ et $\mathbf{F}_{21}^H\mathbf{F}_{21}$). De plus, imposer à ce vecteur d'avoir une norme égale à 1 permet d'éviter d'avoir le vecteur nul comme solution.

Pour que ce vecteur propre soit unique, il est nécessaire que la valeur propre associée ne soit pas de multiplicité supérieure à 1 (*i.e.* ne soit pas dégénérée). Dans le cas contraire, nous aurions une infinité de solutions pour les vecteurs $\mathbf{v}$ et $\mathbf{w}$ (tous les vecteurs normés appartenant à l'espace vectoriel engendré par les différents vecteurs associés à cette valeur propre). Or, en observant la structure de $\mathbf{F}_{12}$ et $\mathbf{F}_{21}$, il est facile de voir que si la solution diagonalisante est parfaitement atteinte, ces deux matrices seront de rang 1 et donc $\mathbf{F}_{12}^H\mathbf{F}_{12}$ et $\mathbf{F}_{21}^H\mathbf{F}_{21}$ auront comme valeur propre minimale zéro et cette dernière aura une multiplicité

de 2 rendant impossible l'estimation de $\mathbf{v}$ et $\mathbf{w}$. Ainsi dans le but d'éviter ce problème, nous nous appuyons sur l'hypothèse 1. Nous allons alors négliger les termes $x_2^2\tilde{N}_{2,1}^{(k)}$ et $x_3^2\tilde{N}_{1,2}^{(k)}$ dans (4.38) et il advient donc :

$$\forall k \in [1,K]_{\mathbb{N}},\ \begin{cases} \tilde{N'}_{1,2}^{(k)} \simeq x_1x_2(\tilde{N}_{2,2}^{(k)} - \tilde{N}_{1,1}^{(k)}) + x_1^2\tilde{N}_{1,2}^{(k)}, \\ \tilde{N'}_{2,1}^{(k)} \simeq x_3x_4(\tilde{N}_{1,1}^{(k)} - \tilde{N}_{2,2}^{(k)}) + x_4^2\tilde{N}_{2,1}^{(k)}. \end{cases} \tag{4.43}$$

Définissons maintenant les deux vecteurs :

$$\mathbf{v}_1 = \begin{pmatrix} x_1x_2 \\ x_1^2 \end{pmatrix};\ \mathbf{w}_1 = \begin{pmatrix} x_3x_4 \\ x_4^2 \end{pmatrix} \tag{4.44}$$

et les deux matrices

$$\mathbf{E}_{12} = \begin{pmatrix} \tilde{N}_{2,2}^{(1)} - \tilde{N}_{1,1}^{(1)} & \tilde{N}_{1,2}^{(1)} \\ \vdots & \vdots \\ \tilde{N}_{2,2}^{(K)} - \tilde{N}_{1,1}^{(K)} & \tilde{N}_{1,2}^{(K)} \end{pmatrix} \tag{4.45}$$

et

$$\mathbf{E}_{21} = \begin{pmatrix} \tilde{N}_{1,1}^{(1)} - \tilde{N}_{2,2}^{(1)} & \tilde{N}_{2,1}^{(1)} \\ \vdots & \vdots \\ \tilde{N}_{1,1}^{(K)} - \tilde{N}_{2,2}^{(K)} & \tilde{N}_{2,1}^{(K)} \end{pmatrix}. \tag{4.46}$$

La nouvelle fonction de coût à minimiser est alors donnée par

$$C_a(\tilde{\mathbf{X}}) \simeq \mathbf{v}_1^H\mathbf{E}_{12}^H\mathbf{E}_{12}\mathbf{v}_1 + \mathbf{w}_1^H\mathbf{E}_{21}^H\mathbf{E}_{21}\mathbf{w}_1. \tag{4.47}$$

Nous pouvons maintenant observer que même si à la convergence, $\mathbf{E}_{12}^H\mathbf{E}_{12}$ et $\mathbf{E}_{21}^H\mathbf{E}_{21}$ peuvent être de rang 1, leur structure empêche l'apparition de valeur propre de multiplicité 2 si et seulement si il existe au moins un $k \in [1,K]_{\mathbb{N}}$ tel que $\tilde{N}_{1,1}^{(k)} \neq \tilde{N}_{2,2}^{(k)}$. Il est remarquable, qu'à la convergence, cette dernière condition est équivalente à la condition d'unicité de la DCS donnée au chapitre précédent (page 56). Nous pouvons aussi remarquer qu'à la convergence, nous nous situons bien à un point stationnaire de notre problème. En effet, une fois l'ensemble de matrices diagonalisé, nous avons

$$\mathbf{E}_{12}^H\mathbf{E}_{12} = \mathbf{E}_{21}^H\mathbf{E}_{21} = \begin{pmatrix} \sum_{k=1}^K |\tilde{N}_{2,2}^{(k)} - \tilde{N}_{1,1}^{(k)}|^2 & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.48}$$

Nous obtenons ainsi, $\mathbf{v}_1^T = \mathbf{w}_1^T = [0\ 1]^T$ et donc $\tilde{\mathbf{X}} = \mathbf{I}$.

Comme précédemment expliqué, la fonction de coût (4.47) admet une solution triviale où $\mathbf{v}_1$ et $\mathbf{w}_1$ sont des vecteurs nuls. Cette solution triviale n'est pas possible car nous imposons aux vecteurs propres d'être de norme 1. Cela implique également que x_1 et x_4 sont différents de 0.

Nous notons $\mathbf{e}$ et $\mathbf{f}$ les vecteurs propres normalisés associés à la plus petite valeur propre de $\mathbf{E}_{12}^H\mathbf{E}_{12}$ et $\mathbf{E}_{21}^H\mathbf{E}_{21}$ respectivement. Nous avons donc le système suivant :

$$\begin{cases} \mathbf{e} = \alpha\mathbf{v}_1, \\ \mathbf{f} = \beta\mathbf{w}_1, \end{cases} \tag{4.49}$$

où α et β sont des paramètres scalaires inconnus.
Nous pouvons directement déduire du système (4.49) et de (4.44) que

$$\begin{cases} \dfrac{x_2}{x_1} = \dfrac{e_1}{e_2}, \\ \dfrac{x_3}{x_4} = \dfrac{f_1}{f_2}. \end{cases} \tag{4.50}$$

Remarque : tout comme $\mathbf{v}_1$ et $\mathbf{w}_1$, $\mathbf{e}$ et $\mathbf{f}$ ne peuvent pas être nuls, cela assure $e_2 \neq 0$ et $f_2 \neq 0$.

Dans le but d'estimer la matrice de mise à jour $\tilde{\mathbf{X}}$, nous proposons cinq manières de paramétrer celle-ci. Pour cela, nous rappelons trois factorisations matricielles décrites au chapitre précédent et nous en introduisons deux autres.

Remarque importante : dans chacune des propositions suivantes, $\tilde{\mathbf{X}}$ est une matrice de déterminant 1, à valeurs dans $\mathbb{C}$ et possédant une indétermination d'échelle sur ses lignes, avec $x_1 \neq 0$ et $x_4 \neq 0$.

Proposition 5. *Il existe deux nombres complexes y_2 et y_3 tels que :*

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & y_2 \\ y_3 & 1 \end{pmatrix} \frac{1}{\sqrt{1 - y_2 y_3}}. \tag{4.51}$$

La proposition 5 vient du fait que l'on a factorisé une matrice diagonale à la matrice $\tilde{\mathbf{X}}$ (comme nous le permet l'indétermination d'échelle de notre problème) afin qu'elle ne dépende plus que de deux paramètres. Le terme $\frac{1}{\sqrt{1-y_2y_3}}$ permet d'imposer son déterminant à être égale à 1.

Proposition 6 (factorisation *LU*)**.** *Il existe deux nombres complexes l et u tels que :*

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} \begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix}. \tag{4.52}$$

Cette proposition découle directement de la factorisation LU.

Proposition 7 (factorisation *QR*)**.** *Il existe deux nombres complexes r et q tels que :*

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{1 + |q|^2}} \begin{pmatrix} 1 & -\bar{q} \\ q & 1 \end{pmatrix} \begin{pmatrix} 1 & r \\ 0 & 1 \end{pmatrix}. \tag{4.53}$$

Cette propositions découle directement de la factorisation QR.

Proposition 8 (factorisation *QR* algébrique)**.** *Il existe deux nombres complexes r et q tels que :*

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{1 + q^2}} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix} \begin{pmatrix} 1 & r \\ 0 & 1 \end{pmatrix} \tag{4.54}$$

La démonstration de cette proposition est donnée dans la section A.1 de l'annexe A.

Proposition 9 (décomposition polaire algébrique)**.** *Si* $(\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}})^2 \neq -1$ *et* $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}})^2 \neq -1$ *, alors il existe deux nombres complexes* h *et* q *tels que :*

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{(1-h^2)(1+q^2)}} \begin{pmatrix} 1 & h \\ h & 1 \end{pmatrix} \begin{pmatrix} 1 & -q \\ q & 1 \end{pmatrix}. \tag{4.55}$$

La démonstration de cette proposition est donnée dans la section A.1 de l'annexe A.

Dans la proposition 9, la matrice $\tilde{\mathbf{X}}$ est décomposée comme le produit d'une matrice orthogonale complexe et d'une matrice symétrique complexe. Nous rappelons qu'une matrice orthogonale complexe est une matrice complexe $\mathbf{Q}$ vérifiant $\mathbf{Q}^\intercal\mathbf{Q} = \mathbf{I}$ et une matrice complexe symétrique est une matrice complexe $\mathbf{H}$ vérifiant $\mathbf{H}^\intercal = \mathbf{H}$. La décomposition proposée dans la proposition 9 est un cas particulier de la décomposition polaire algébrique présentée et étudiée dans [137] et [59] pour laquelle la matrice orthogonale complexe s'apparente à une matrice de rotation plane de Givens et la matrice symétrique complexe s'apparente à une matrice hyperbolique. Cette décomposition est plus adaptée à notre problème que la décomposition polaire classique (définie comme le produit d'une matrice unitaire et d'une matrice symétrique hermitienne). En effet, la décomposition polaire algébrique ne faisant pas intervenir le produit croisé entre les paramètres complexes et leurs conjuguées, elle nous permet d'estimer directement les paramètres dans $\mathbb{C}$ plutôt que dans $\mathbb{R}^2$.

N'ayant pas trouvé de trace d'une telle factorisation dans la littérature, nous appelons «factorisation QR algébrique» la factorisation de la proposition 8 en référence à la décomposition polaire algébrique. Dans le cas complexe, la factorisation QR est classiquement le produit entre une matrice unitaire et une matrice triangulaire complexe. Nous remplaçons ici la matrice unitaire par une matrice orthogonale complexe.

À ce stade, nous pouvons résumer notre problème de la manière suivante :

Problème 4.2.1. *Soit* $\tilde{\mathbf{X}}$ *une matrice inconnue à valeurs dans* $\mathbb{C}$ *dont le déterminant est égal à* 1 *et dont les lignes ne peuvent être estimées qu'à un facteur d'échelle près :*

$$\tilde{\mathbf{X}} = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}. \tag{4.56}$$

Soient $\mathbf{e}$ *et* $\mathbf{f}$ *deux vecteurs non-nuls connus et définis par :*

$$\mathbf{e} = \alpha \begin{pmatrix} x_1 x_2 \\ x_1^2 \end{pmatrix};\ \mathbf{f} = \beta \begin{pmatrix} x_3 x_4 \\ x_4^2 \end{pmatrix} \tag{4.57}$$

où α, β *sont des paramètres inconnus. Nous cherchons* x_1*,* x_2*,* x_3 *et* x_4 *tels que* $\lim_{\substack{x_2 \to 0 \\ x_3 \to 0}} (x_1) = \lim_{\substack{x_2 \to 0 \\ x_3 \to 0}} (x_4) = 1$.

4.2.2 Solutions au problème 4.2.1

Nous donnons ici plusieurs manières de résoudre le problème 4.2.1.

Solution triviale. La proposition 5 fournit une solution triviale. En effet, en considérant l'équation (4.50), nous obtenons directement $y_2 = \frac{e_1}{e_2}$ et donc $y_3 = \frac{f_1}{f_2}$.

Solution basée sur une factorisation LU. La proposition 6 nous permet de chercher $\tilde{\mathbf{X}}$ de la forme :

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & u \\ l & lu+1 \end{pmatrix}. \tag{4.58}$$

Puis grâce à (4.50) et (4.58) nous en déduisons :

$$\begin{cases} u = \dfrac{e_1}{e_2}, \\ l = \dfrac{f_1 e_2}{f_2 e_2 - f_1 e_1}. \end{cases} \tag{4.59}$$

Solution basée sur la factorisation QR. La proposition 7 nous permet de chercher la matrice $\tilde{\mathbf{X}}$ sous la forme :

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{1+|q|^2}} \begin{pmatrix} 1 & r-\bar{q} \\ q & qr+1 \end{pmatrix}. \tag{4.60}$$

En utilisant (4.50) et (4.60) nous obtenons :

$$\begin{cases} r = \dfrac{e_1}{e_2} + \bar{q} \\ |q|^2 e_2 f_1 + q(e_1 f_1 - e_2 f_2) + e_1 f_2 = 0. \end{cases} \tag{4.61}$$

À la convergence, la matrice $\tilde{\mathbf{X}}$ doit être proche de la matrice identité. Cela se traduit par le fait que q doit être proche de zéro. Nous choisissons donc q comme la racine du polynôme de plus petit module.

Solution basée sur la factorisation QR algébrique. La proposition 8 nous permet de chercher $\tilde{\mathbf{X}}$ de la forme :

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{1+q^2}} \begin{pmatrix} 1 & r-q \\ q & qr+1 \end{pmatrix}. \tag{4.62}$$

Grâce à (4.50) et (4.62) nous avons :

$$\begin{cases} r = \dfrac{e_1}{e_2} + q \\ q^2 e_2 f_1 + q(e_1 f_1 - e_2 f_2) + e_1 f_2 = 0. \end{cases} \tag{4.63}$$

Nous pouvons remarquer que, dans le cas réel, les solutions fournies par les factorisations QR et QR algébrique sont égales. Comme précédemment expliqué, la matrice $\tilde{\mathbf{X}}$ doit être proche de la matrice identité à la convergence. Nous choisissons donc q comme la racine du polynôme de plus petit module.

Solution basée sur la décomposition polaire algébrique. Si nous avons $(\frac{e_1}{e_2})^2 \neq -1$ et $(\frac{f_1}{f_2})^2 \neq -1$, alors la proposition 9 nous permet de chercher $\tilde{\mathbf{X}}$ de la forme :

$$\tilde{\mathbf{X}} = \frac{1}{\sqrt{(1-h^2)(1+q^2)}} \begin{pmatrix} 1+qh & h-q \\ h+q & 1-qh \end{pmatrix} \tag{4.64}$$

En identifiant (4.50) et (4.64) nous obtenons :

$$\begin{cases} h = \dfrac{e_1 + e_2 q}{e_2 - e_1 q} \\ q^2(e_2 f_1 - f_2 e_1) + 2q(e_2 f_2 + f_1 e_1) + e_1 f_2 - f_1 e_2 = 0. \end{cases} \tag{4.65}$$

Pour les mêmes raisons que les factorisations QR et QR algébrique, nous choisissons ici q comme la racine du polynôme de plus petit module.

Solution sous forme unifiée. Comme le problème 4.2.1 admet un seul degré de liberté, nous pouvons aussi donner une forme de solution globale permettant de réécrire les solutions précédentes sous forme unifiée. Pour cela, nous proposons ici de paramétrer la matrice $\tilde{\mathbf{X}}$ comme :

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & x'_2 \\ x'_3 & x'_4 \end{pmatrix} \frac{1}{\sqrt{x'_4 - x'_2 x'_3}}, \tag{4.66}$$

avec $x'_4 \neq 0$. Nous pouvons alors immédiatement déduire de la définition de $\mathbf{e}$ et $\mathbf{f}$ que :

$$\begin{cases} x'_2 = \dfrac{e_1}{e_2} \\ x'_4 = g(\dfrac{e_1}{e_2}, \dfrac{f_1}{f_2}) \\ x'_3 = \dfrac{f_1}{f_2} x'_4, \end{cases} \tag{4.67}$$

avec g n'importe quelle fonction de $\mathbb{D} \to \mathbb{C}^*$ où $\mathbb{D} = \mathbb{C}^2 - \{(a,b) \in \mathbb{C}^2 \setminus ab \neq 1\}$ et vérifiant $\lim_{\substack{a\to 0 \\ b\to 0}} g(a,b) = 1$.

Par exemple, dans le cas de la factorisation LU $g(a,b) = \frac{1}{1-ab}$.

Les solutions précédemment données définissent une classe d'algorithme que nous appelons JAPAM (pour *Joint eigenvalue decomposition Algorithms using a PArametrized Matrix*). Nous obtenons ainsi cinq algorithmes JAPAM-1, JAPAM-2, JAPAM-3, JAPAM-4 et JAPAM-5 correspondant respectivement aux solutions précédemment proposées. La structure de ces algorithmes est résumé dans la table algorithmique 6.

Il est clair que la paramétrisation de la matrice $\tilde{\mathbf{X}}$ peut être changée au cours du processus itératif, nous proposons donc un sixième algorithme nommé JAPAM-M (pour JAPAM-Mixed) pour lequel la paramétrisation choisie est celle qui fait le plus décroître le critère (4.47) à chaque itération. Le coût de calcul supplémentaire de JAPAM-M est négligeable devant la mise à jour des matrices à diagonaliser. La particularité des algorithmes proposés est qu'ils fonctionnent tous sur des données réelles et complexes sans modifications contrairement aux méthodes existantes de diagonalisation conjointe.

Algorithme 6 Algorithme JAPAM

Soit S_C un critère d'arrêt et It_{max} le nombre maximal d'itérations ;
Initialisation de $\mathbf{B}$ avec $\mathbf{I}$ ou n'importe quel choix judicieux ;
$it \leftarrow 1$;
while S_C est $faux$ and $it < It_{max}$ **do**
  for $i = 1$ to $R - 1$ **do**
    for $j = i + 1$ to R **do**
      Calculer $\mathbf{E}_{12}^H\mathbf{E}_{12}$ et $\mathbf{E}_{21}^H\mathbf{E}_{21}$ à partir de (4.45) et (4.46) ;
      Effectuer la décomposition en éléments propres de $\mathbf{E}_{12}^H\mathbf{E}_{12}$ et $\mathbf{E}_{21}^H\mathbf{E}_{21}$ pour trouver $\mathbf{e}$ et $\mathbf{f}$;
      Calculer les paramètres de $\tilde{\mathbf{X}}$ en utilisant une des solutions proposées dans 4.2.2 ;

      Construire $\mathbf{X}^{(i,j)}$ à partir de x_1, x_2, x_3 et x_4 ;
      Mettre à jour $\mathbf{B} \leftarrow \mathbf{X}^{(i,j)}\mathbf{B}$;
      Mettre à jour $\mathbf{N}^{(k)} \leftarrow \mathbf{X}^{(i,j)}\mathbf{N}^{(k)} \left(\mathbf{X}^{(i,j)}\right)^{-1}$ $\forall k \in [1, K]_{\mathbb{N}}$;
    end for
  end for
  Mettre à jour S_C ;
  $it \leftarrow it + 1$;
end while

Complexité numérique : ici l'ensemble de matrices à diagonaliser est mis à jour $R(R-1)/2$ fois à chaque itération par une matrice égale à l'identité excepté pour quatre termes. La complexité des algorithmes JAPAM est alors

$$\Gamma[JAPAM] \simeq 4KR^3 \tag{4.68}$$

dans le cas réel et

$$\Gamma[JAPAM] \simeq 16KR^3 \tag{4.69}$$

dans le cas complexe.

4.3 Simulations numériques

Nous comparons maintenant les performances des algorithmes SJDTE, JDTE et JAPAM avec celles des algorithmes existants de DCS : SH-RT, JDTM, JUST, JET-U et JET-O. Dans nos comparaisons, nous présenterons les performances des algorithmes JAPAM-2, JAPAM-4, JAPAM-5 et JAPAM-M. En effet, suite à des études préliminaires, nous avons pu constater que ces derniers sont les plus performants des algorithmes de type JAPAM.

Dans nos simulations, l'ensemble de matrices à diagonaliser est construit de la manière suivante :

$$\mathbf{M}^{(k)} = \frac{\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}}{||\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}||_F} + \sigma\frac{\mathbf{E}^{(k)}}{||\mathbf{E}^{(k)}||_F}, \quad \forall k = 1, ..., K. \tag{4.70}$$

Les parties réelles et imaginaires des matrices $\mathbf{A}$, $\mathbf{D}^{(k)}$ et $\mathbf{E}^{(k)} \in \mathbb{C}^{R\times R}$ sont générées indépendamment selon une loi normale centrée réduite. σ est un paramètre permettant de régler le Rapport Signal sur Bruit (RSB). Le RSB est défini ici comme $-20\log_{10}(\sigma)$.

Après élimination de l'indétermination de permutation et d'échelle, nous utilisons le critère

$$r_A = \frac{||\mathbf{A} - \mathbf{B}^{-1}||_F}{||\mathbf{A}||_F} \tag{4.71}$$

dans le but de mesurer l'erreur d'estimation entre la matrice de vecteurs propres vraie et la matrice de vecteurs propres estimée.

Un second critère est utilisé pour mesurer le coût de calcul des algorithmes. Nous appelons ce critère Γ_{tot}[algorithme], nous le définissons comme le produit de la complexité numérique Γ avec le nombre d'itérations dont l'algorithme étudié a besoin pour converger.

Pour le critère d'arrêt, nous définissons aussi la fonction suivante :

$$S(\mathbf{B}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{\mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{-1}\}\|_F^2. \tag{4.72}$$

Nous considérons qu'un algorithme a atteint la convergence lorsque le critère $|S(\mathbf{B}_{it+1}) - S(\mathbf{B}_{it})|$ est inférieur à 10^{-6} (où $\mathbf{B}_{it}$ est l'estimation de la matrice diagonalisante à l'itération it). Nous fixons le nombre d'itérations maximal à 150 pour tous les algorithmes.

Nous étudions ici le comportement des algorithmes de DCS selon deux scénarios. Dans le premier, nous étudions l'impact du niveau de bruit et dans le second l'influence de la taille des matrices à diagonaliser. Pour chaque scénario, nous effectuons 200 réalisations de Monté-Carlo (pour chaque valeur du RSB ou chaque taille de matrice), bien évidement à chaque réalisation un nouvel ensemble de matrice est généré. Ainsi nous étudierons pour chaque algorithme l'erreur r_A en moyenne, en médiane ainsi que son écart type. Nous considérons que plus l'erreur moyenne et l'écart type d'un algorithme sont faibles plus celui-ci est précis.

4.3.1 Scénario 1

Paramètres : nous considérons ici des matrices de taille $R = 5$, un nombre de matrices $K = 20$ et nous varions le RSB de 0 dB à 100 dB par pas de 10 dB. La matrice $\mathbf{B}$ est initialisée avec la matrice identité.

Résultats : sur les figures 4.1 et 4.2, nous pouvons observer que toutes les méthodes proposées dans ce chapitre estiment mieux la matrice de vecteurs propres (en médiane et en moyenne) que les méthodes existantes de DCS à l'exception de JAPAM-2. En effet, pour des RSB inférieurs à 30 dB, JAPAM-2 fournit une moins bonne estimation de la matrice de vecteurs propres en médiane que tous les algorithmes existants (figure 4.1). Quant à l'estimation moyenne de la matrice de vecteurs propres (figure 4.2), JAPAM-2 a le même comportement en moyenne que JET-O et JET-U basés eux aussi sur une décomposition LU. Ainsi JAPAM-2 fournit une meilleure estimation de la matrice de vecteurs propres que les algorithmes basés sur une décomposition polaire pour les forts RSB, puis sous 60 dB, son erreur moyenne augmente. JAPAM-2 est donc peu robuste au bruit.

Concernant l'écart type du critère r_A (figure 4.3), nous pouvons remarquer que les méthodes proposées ont un écart type plus faible que les algorithmes de DCS existants pour toutes les valeurs du RSB supérieures à 30 dB. Sous 30 dB, leur écart type est quasiment

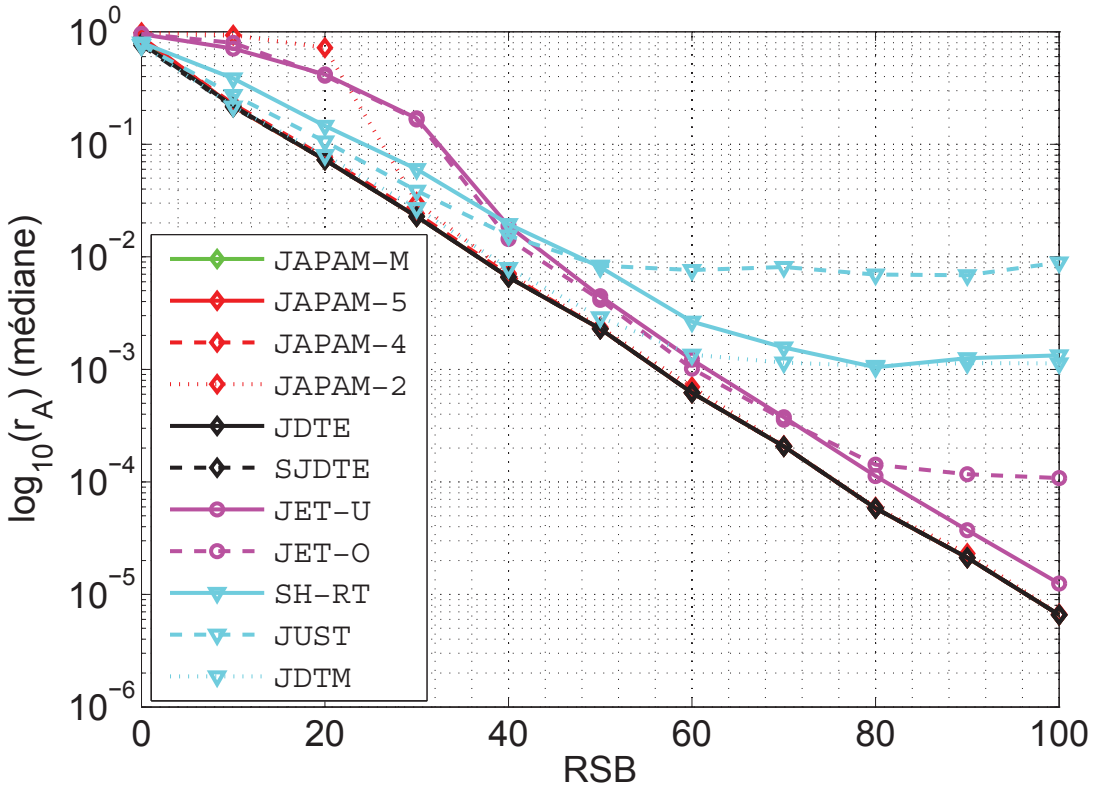

FIGURE 4.1: r_A médian en fonction du RSB.

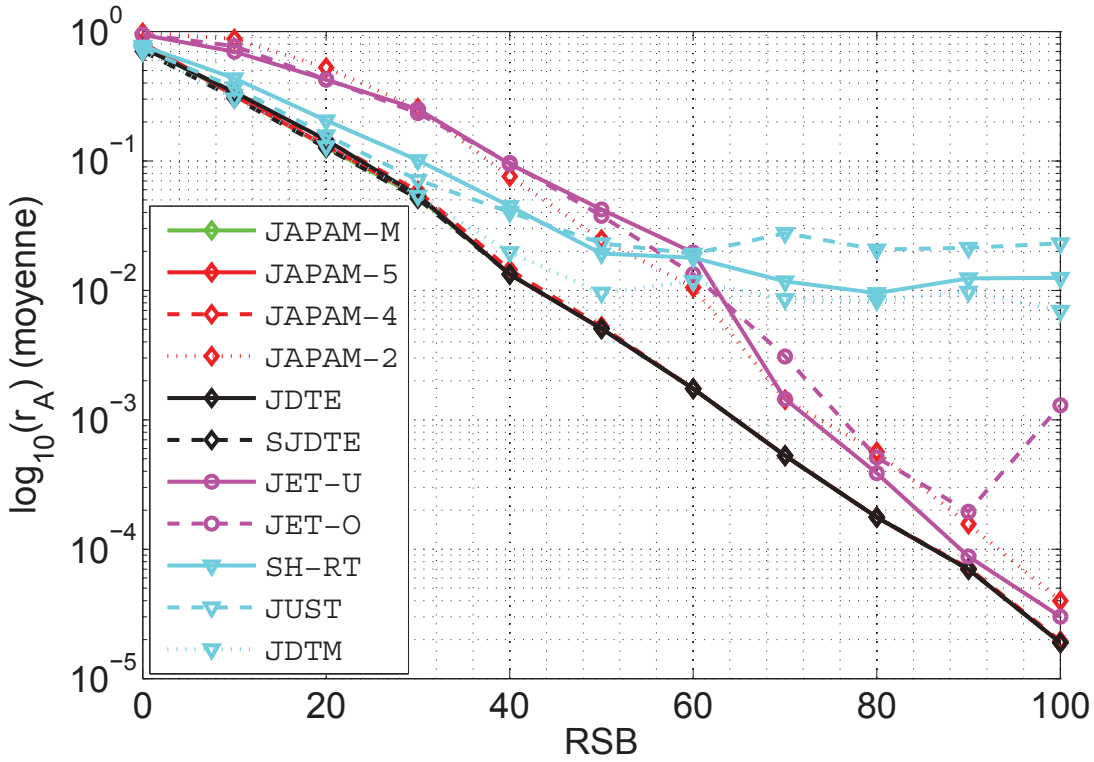

FIGURE 4.2: r_A moyen en fonction du RSB.

égal à celui que SH-RT, JUST et JDTM. Encore une fois, JAPAM-2 se comporte de la même manière que les algorithmes JET-O et JET-U.

En observant l'écart type et la moyenne de l'erreur r_A, nous pouvons voir que l'ensemble des algorithmes proposés (à l'exception de JAPAM-2) sont ici plus précis que les algorithmes existants.

Nous présentons sur la figure 4.4 les coûts de calcul totaux des algorithmes. Les algorithmes les plus coûteux sont les algorithmes existants basées sur une décomposition polaire. JET-U est, quant à lui, le moins coûteux de tous les algorithmes pour toutes les valeurs considérées du RSB. Nous pouvons observer que les algorithmes JDTE et SJDTE sont moins coûteux que tous les algorithmes étudiés pour des valeurs du RSB supérieures à 30 dB (à l'exception de JET-U). Les algorithmes de type JAPAM ont, quant à eux, un coût de calcul compris entre celui des algorithmes basés sur une décomposition polaire et JET-O. Les algorithmes les moins coûteux de la famille JAPAM sont JAPAM-5 et JAPAM-M. JAPAM-M est légèrement moins coûteux que JAPAM-5 sous 40 dB.

Ainsi dans ce premier scénario, nous montrons que les algorithmes développés dans ce chapitre sont plus précis que les algorithmes de DCS existants pour des matrices à diagonaliser de petite taille (à l'exception de JAPAM-2). De plus, les algorithmes proposés ont un coût calcul proche des algorithmes existants les moins coûteux. En résumé, JAPAM-4, JAPAM-5, JAPAM-M, JDTE et SJDTE sont bien moins coûteux que JDTM, JUST et SH-RT tout étant bien plus précis que JET-O et JET-U.

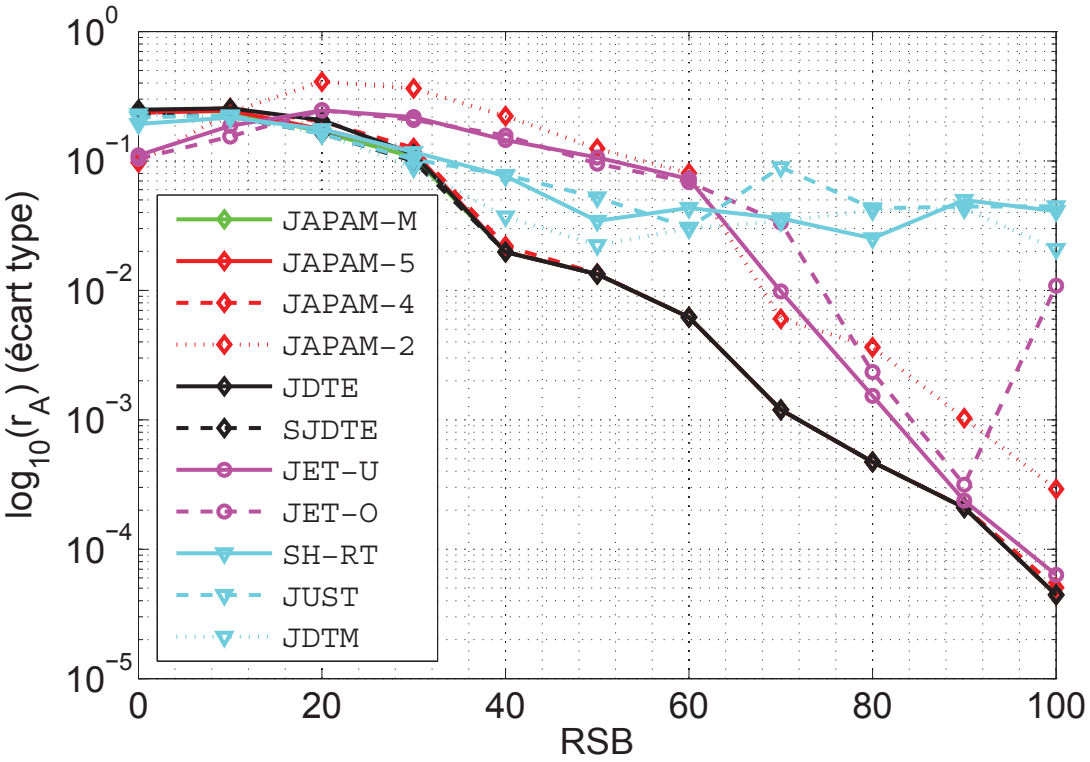

FIGURE 4.3: écart type de r_A en fonction du RSB.

4.3.2 Scénario 2.a

Paramètres : nous fixons maintenant la valeur du RSB à 50 dB, le nombre de matrices à $K = 20$, puis nous faisons varier la taille des matrices à diagonaliser de $R = 3$ à $R = 22$ par pas de 1. La matrice $\mathbf{B}$ est encore une fois initialisée avec la matrice identité. Nous choisissons $R = 22$ comme valeur maximale, car cette valeur est assez grande pour mettre en évidence les problèmes de convergence de nos méthodes.

Résultats : en observant la valeur médiane et moyenne de r_A (figure 4.5 et 4.6 respectivement). Pour chacune des méthodes proposées, nous pouvons voir que ces valeurs augmentent de manière radicale à partir d'une valeur critique de R en fonction de l'algorithme considéré. Ce scénario permet donc de définir une zone de bon fonctionnement pour chacun des algorithmes proposés.

La zone de bon fonctionnement de JAPAM-2 est située sous $R = 7$ (figure 4.5). Pour $R < 6$, il est compétitif avec les algorithmes de référence concernant la valeur moyenne (figure 4.6) et l'écart type de r_A (figure 4.7). JAPAM-2 a un coût calcul (figure 4.8) plus faible que les algorithmes de références basés sur une décomposition polaire mais reste plus coûteux que les algorithmes de type JET.

La zone de bon fonctionnement de JDTE est située sous $R = 8$. Sous $R = 7$, il fait clairement parti des algorithmes les plus précis. JDTE est un des algorithmes les moins coûteux (dans sa zone de bon fonctionnement), en effet son coût calcul se situe entre JET-O et JET-U qui sont les algorithmes de référence les moins coûteux.

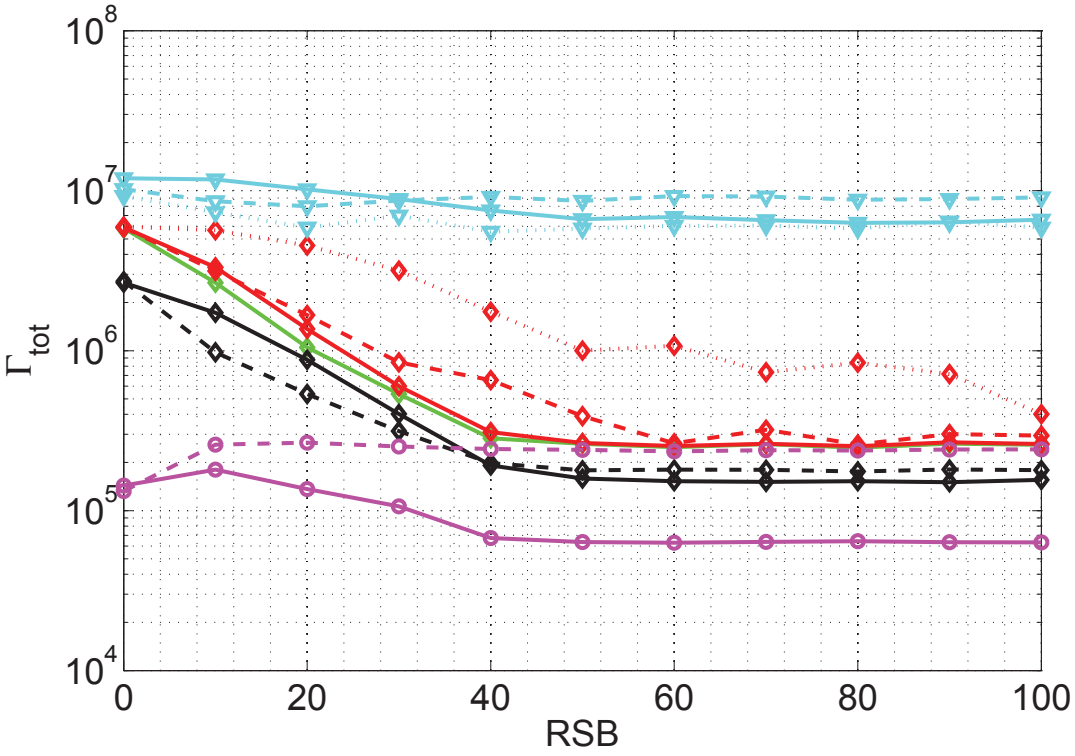

FIGURE 4.4: Coût calcul en fonction du RSB.

La zone de bon fonctionnement de JAPAM-4 se situe sous $R = 10$. Pour $R = 7$, il atteint un niveau de précision intéressant en comparaison avec les algorithmes de référence. Sous $R = 6$, il est un des algorithmes les plus précis.

La zone de bon fonctionnement de JAPAM-5 se situe sous $R = 12$. Dans cette zone, JAPAM-5 fait parti des algorithmes ayant la plus petite erreur moyenne et le plus petit écart type (à l'exception de $R = 7$ et $R = 10$). JAPAM-5 est légèrement plus coûteux que JET-O, ce qui en fait aussi un algorithme intéressant en termes de coût calcul.

La zone de bon fonctionnement de JAPAM-M se situe sous $R = 15$ ce qui justifie l'intérêt de cette version mixée. Il fournit une des meilleures estimations de la matrice de vecteurs propres pour $R \leq 12$. Son coût calcul est légèrement inférieur à celui de JAPAM-5.

Ici, SJDTE est l'algorithme le plus performant que nous proposons. En effet, sa zone de bon fonctionnement se situe sous $R = 22$, il fournit les meilleurs résultats en termes de précision pour $R \leq 20$. De plus, seul JET-U est moins coûteux que SJDTE.

Ainsi, dans ce scénario, nous avons montré que nos algorithmes fournissent des résultats intéressants en termes de précision et de coût calcul dans une zone de bon fonctionnement variable en fonction de l'algorithme considéré lorsqu'ils sont initialisés avec la matrice identité. Cependant, ces derniers ne sont pas adaptés à des matrices de très grande taille. Ce comportement peut être expliqué par les approximations induites par l'hypothèse 1, car les algorithmes de la littérature ne sont pas basés sur cette hypothèse et sont moins sensibles aux grandes valeurs de R. Dans le prochain scénario, nous montrerons que ce problème peut être corrigé facilement.

Le comportement de nos algorithmes permet de faire deux autres conclusions intéres-

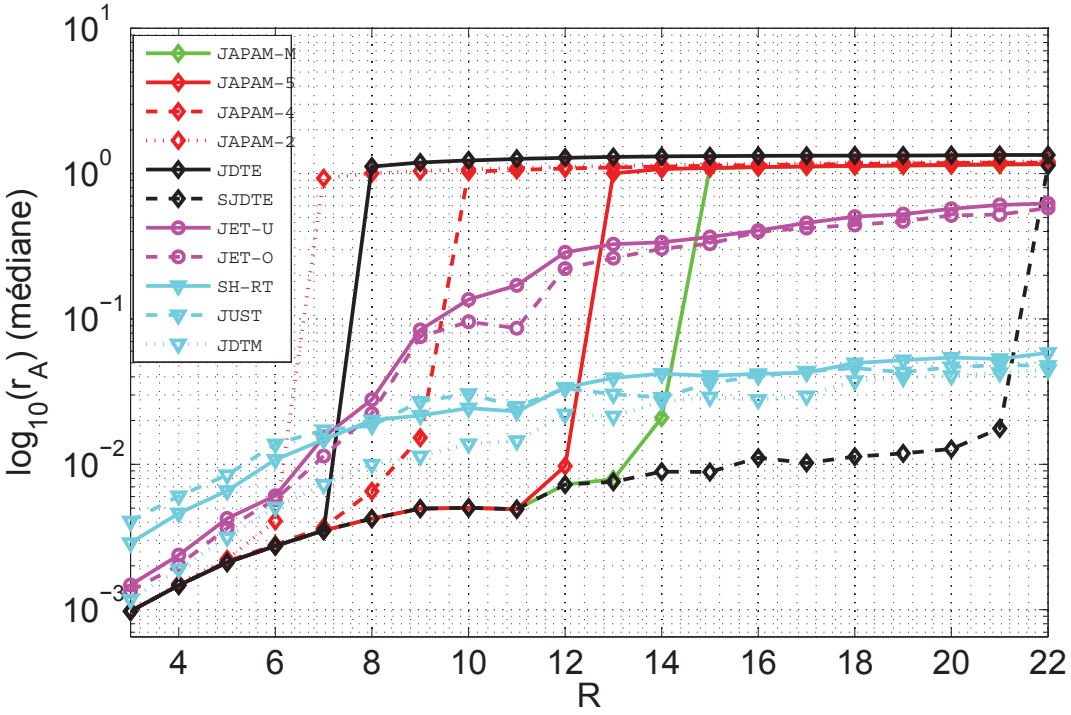

FIGURE 4.5: r_A médian en fonction de R (initialisation de la matrice diagonalisante avec la matrice identité).

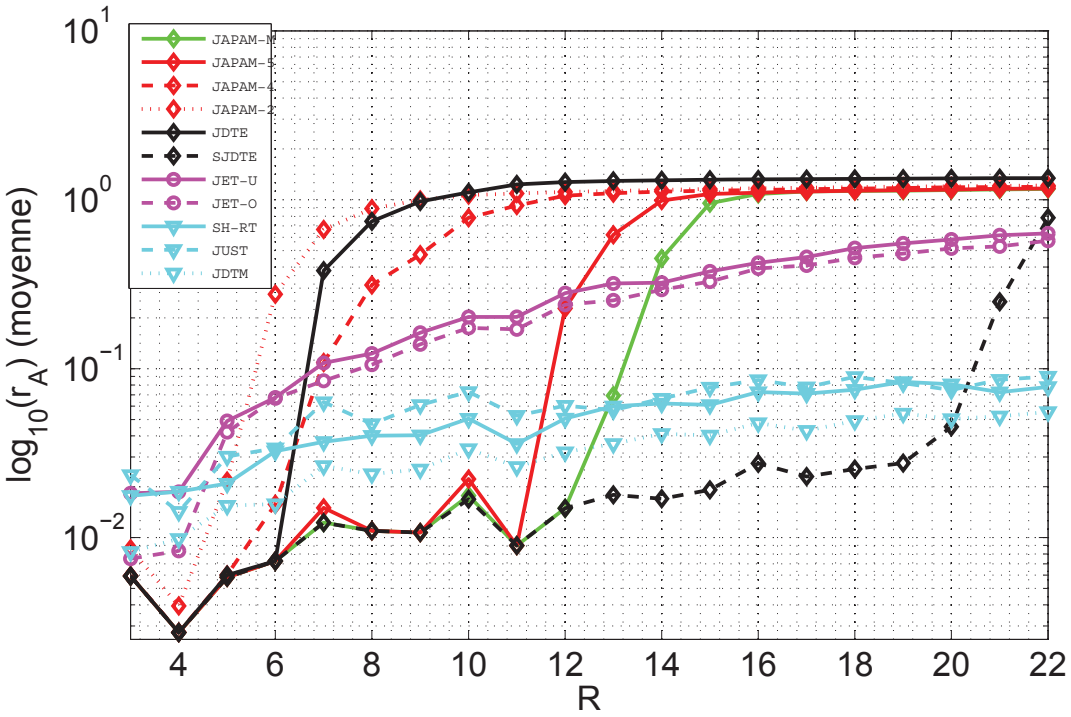

FIGURE 4.6: r_A moyen en fonction de R (initialisation de la matrice diagonalisante avec la matrice identité).

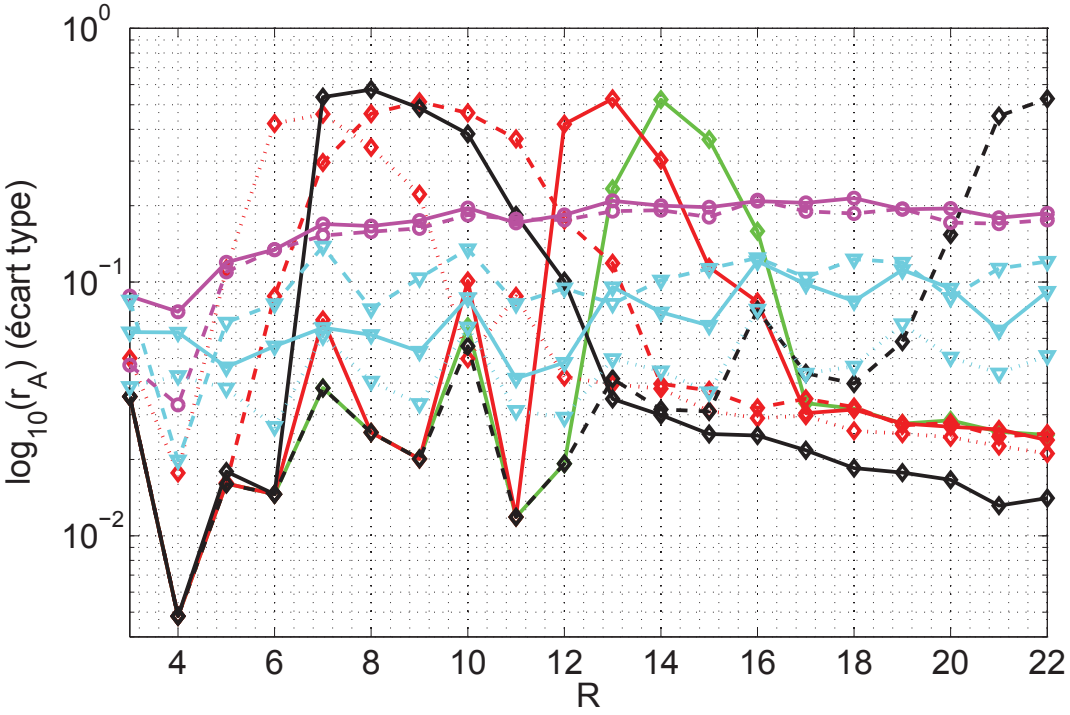

FIGURE 4.7: écart type de r_A en fonction de R (initialisation de la matrice diagonalisante avec la matrice identité).

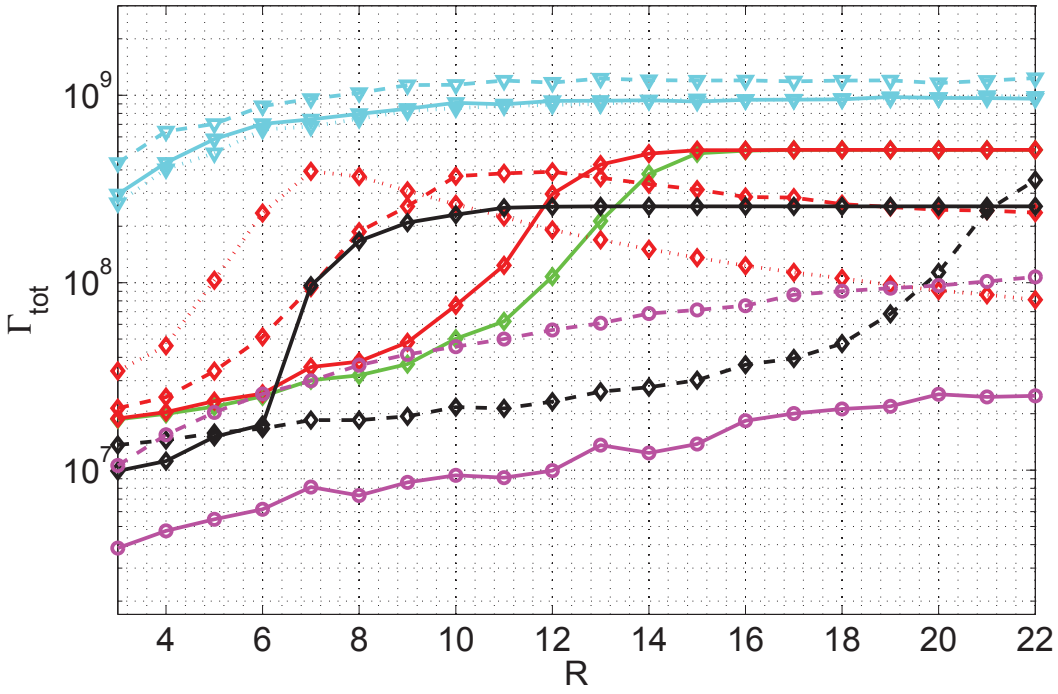

FIGURE 4.8: Coût calcul en fonction de R (initialisation de la matrice diagonalisante avec la matrice identité).

santes. Premièrement, la structure commune des algorithmes de type JAPAM permet de mettre en évidence que la factorisation LU est moins robuste aux erreurs d'approximation considérées sur le critère que la factorisation QR algébrique, elle même moins robuste que la décomposition polaire algébrique. Deuxièmement, le principe de calcul des paramètres de la matrice de mise à jour étant similaire pour JDTE et SJDTE, nous pouvons supposer que la stratégie de balayage par paire est plus robuste aux erreurs d'approximation du critère qu'une stratégie d'estimation globale.

4.3.3 Scenario 2.b

paramètres : nous fixons ici la valeur du RSB à 50 dB, le nombre de matrices à $K = 20$, puis nous faisons varier la taille des matrices à diagonaliser de $N = 3$ à $N = 22$ par pas de 1 comme pour le scénario précédent. Cependant, la matrice $\mathbf{B}$ est ici initialisée avec le résultat d'une GEVD effectuée sur deux matrices de l'ensemble $\mathbf{M}^{(k)}$.

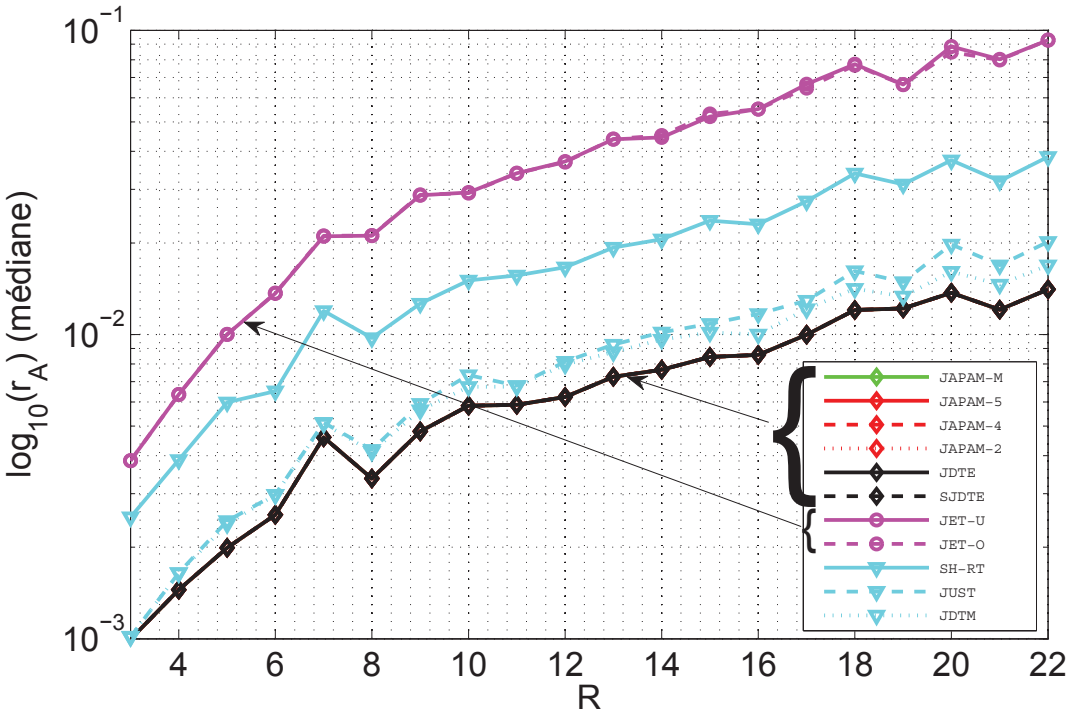

FIGURE 4.9: r_A médian en fonction de R (initialisation de la matrice diagonalisante par GEVD).

résultats : nous pouvons remarquer ici que l'initialisation par une GEVD permet aux algorithmes proposés de fonctionner pour toutes les tailles de matrices étudiées.

En médiane, nos algorithmes permettent d'obtenir la meilleure estimation et donnent des résultats similaires entre eux.

En moyenne, SJDTE donne globalement les meilleurs résultats. Les autres méthodes proposées rivalisent avec les méthodes paramétrisées par le biais d'une décomposition

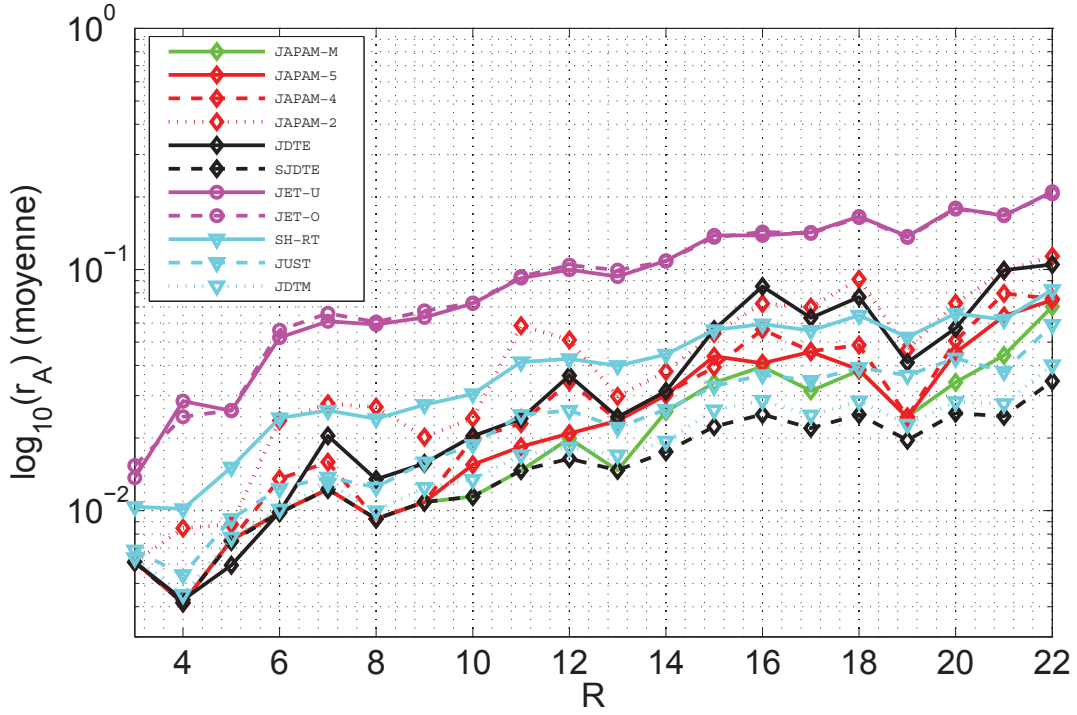

FIGURE 4.10: r_A moyen en fonction de R (initialisation de la matrice diagonalisante par GEVD).

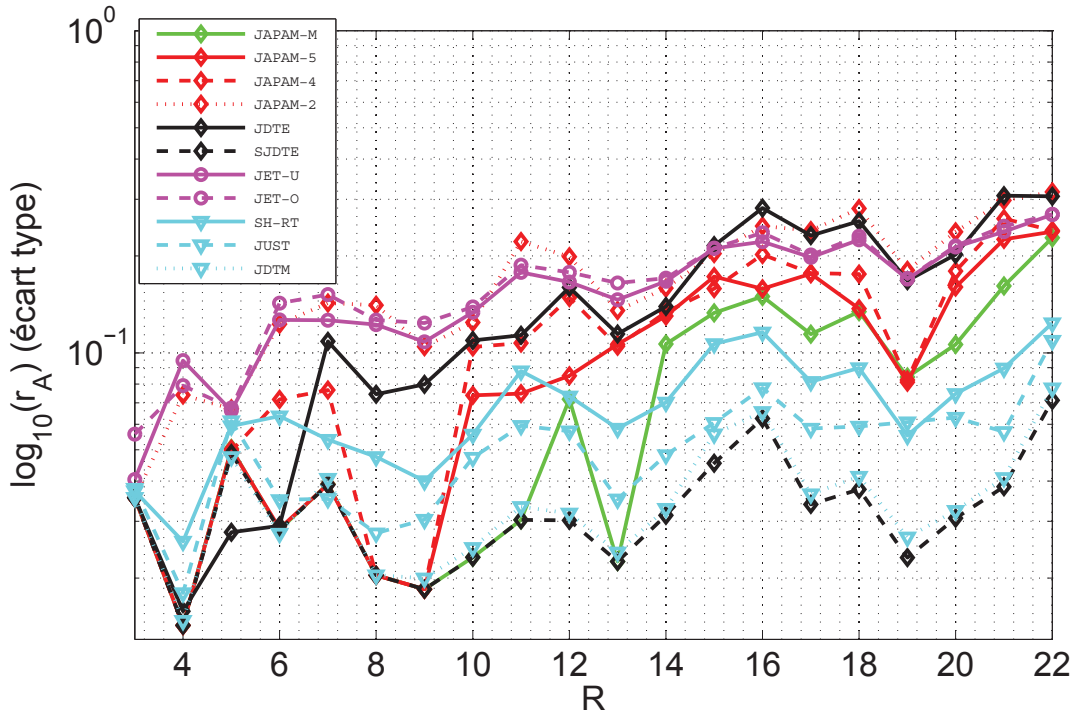

FIGURE 4.11: écart type de r_A en fonction de R (initialisation de la matrice diagonalisante par GEVD).

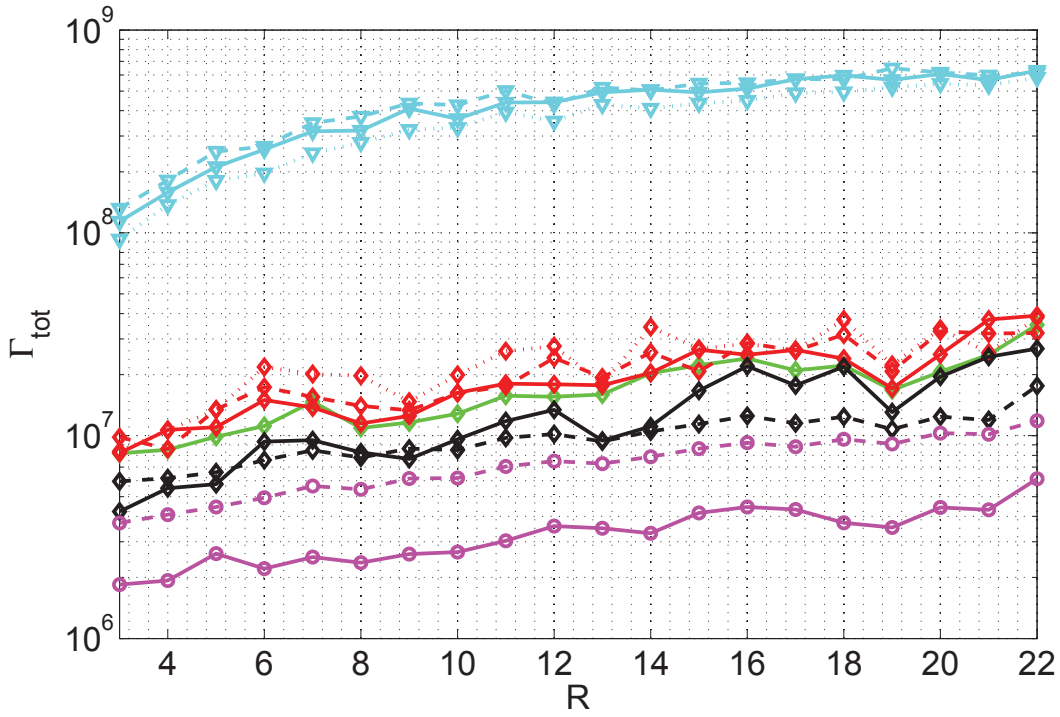

FIGURE 4.12: Coût calcul en fonction de R (initialisation de la matrice diagonalisante par GEVD).

polaire alors que JET-U et JET-O sont nettement surpassés. JAPAM-5 et JAPAM-M procurent de meilleurs résultats que JAPAM-4, JDTE et JAPAM-2.

L'écart type de r_A (figure 4.11) apparaît comme le plus faible pour SJDTE bien que JDTM atteigne quasiment le même niveau de performances. Les autres algorithmes proposés voient leur écart type augmenter lorsque la taille des matrices étudiées augmente. JAPAM-2 est l'algorithme proposé le moins stable mais atteint le même niveau de stabilité que JET-U et JET-O.

Le coût calcul des algorithmes étudiés est affiché sur la figure 4.12. Toutes les méthodes proposées sont légèrement plus coûteuses que JET-O, alors que SH-RT, JDTM et JUST sont bien plus coûteux que les autres algorithmes. SJDTE est l'algorithme proposé le moins coûteux.

4.4 Bilan du chapitre

Dans ce chapitre, nous avons présenté deux classes d'algorithme fonctionnant sur des données réelles et complexes sans modifications contrairement aux algorithmes de DCS existants. Nous proposons dans la section A.2 de l'annexe une simulation numérique pour montrer que nos méthodes fonctionnent également sur des données réelles.

Les deux classes sont basées sur l'hypothèse que nous sommes proches de la solution diagonalisante et d'un point stationnaire. Les approximations induites par cette hypothèse nous permettent d'obtenir une expression analytique des paramètres à estimer. De plus,

trois algorithmes (JAPAM-5, JAPAM-M et SJDTE) sont particulièrement efficaces et peu coûteux en comparaison avec les algorithmes existants de DCS. Cependant, les erreurs d'approximation du critère font que ces algorithmes ne fonctionnent plus à partir d'une certaine taille de matrices. Pour régler ce problème, les algorithmes peuvent être initialisés à l'aide d'une GEVD effectuée sur deux matrices de l'ensemble à diagonaliser.

La première classe est basée sur une paramétrisation particulière de la matrice de mise à jour et sur un développement de Taylor pour calculer son inverse. Les simulations permettent de mettre en évidence qu'une approche de balayage par paire est plus robuste aux erreurs d'approximations considérées sur le critère qu'une approche d'estimation globale.

La deuxième classe donne naissance à des algorithmes ayant tous la même structure, leur différence réside dans la paramètrisation de la matrice de mise à jour. À l'aide de simulations numériques, nous pouvons montrer que la factorisation LU est moins robuste aux erreurs d'approximations du critère que la factorisation QR algébrique, elle même moins robuste que la décomposition polaire algébrique. Nous proposons aussi une version mixée choisissant la meilleure solution pour calculer la matrice de mise à jour. Cette version améliore les résultats obtenus et permet de traiter des matrices de plus grande taille sans utiliser de GEVD.

Chapitre 5

Nouveaux algorithmes de DCS sous contraintes de non-négativité

Dans ce chapitre, nous allons présenter nos travaux sur le problème de DCS obtenu à partir de la réécriture du problème de décomposition CP non-négative par le biais d'une compression par SVD (chapitre 2, section 2.3.7.2).

Soient $\mathbf{U}$, $\mathbf{S}$ et $\mathbf{V}$ les matrices de la SVD appliquée sur une des matrices de dépliement du tenseur comme expliqué au chapitre 2 (équation (2.73)) et $\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}$ avec $k \in [1; K]_{\mathbb{N}}$ les matrices de l'ensemble à diagonaliser, nous avons alors comme contraintes :

$$\begin{aligned} \mathbf{D}^{(k)} &\succeq 0 \ \forall k \in [1; K]_{\mathbb{N}}, \\ \mathbf{USB}^T &\succeq 0, \\ \mathbf{VA} &\succeq 0, \end{aligned} \tag{5.1}$$

où la matrice $\mathbf{B}$ est la matrice diagonalisante de l'ensemble de matrices $\mathbf{M}^{(k)}$. Nous rappelons que dans l'algorithme DIAG [22], la matrice $\mathbf{D}^{(k)}$ contient la division terme à terme de deux lignes d'une des matrices facteurs du problème de décomposition CP. Les matrices $\mathbf{USB}^T$ et $\mathbf{VA}$ donnent directement l'expression des autres matrices facteurs ou alors du produit de Khatri-Rao des autres matrices facteurs, d'où les contraintes précédentes.

Nous proposons, en premier lieu, une méthode basée sur la mise à jour multiplicative de la matrice diagonalisante $\mathbf{B}$ contraignant les valeurs propres des matrices $\mathbf{M}^{(k)}$ à être positives. Puis nous proposerons des méthodes de type ALS et ADMM prenant en compte les trois contraintes de (5.1).

5.1 Méthode de DCS avec contrainte de positivité sur les valeurs propres

Nous considérons ici que les éléments diagonaux des matrices $\mathbf{D}^{(k)}$ sont strictement positifs. Notre première idée pour contraindre la positivité sur les valeurs propres fut d'utiliser la méthode JD-BGL [114] basée sur la fonction de coût log-vraissemblance (3.20). Cette méthode permet de diagonaliser un ensemble de matrices définies positives en conservant la positivité de leurs termes diagonaux au cours des itérations, elle ne peut donc pas être directement adaptée au problème de DCS. En effet, une matrice ayant ses valeurs propres

positives n'a pas forcément ses éléments diagonaux positifs (dans le cas où la matrice de vecteurs propres est non-orthogonale). Nous proposons donc ici une méthode permettant de transformer le problème de DCS en un problème de DCC-O de matrices définies positives.

Toute matrice carrée peut se décomposer comme le produit d'une matrice triangulaire inférieure $\mathbf{L}$ (ou supérieure) et d'une matrice orthogonale $\mathbf{Q}$ tel que $\mathbf{A} = \mathbf{LQ}$. Ainsi l'ensemble de matrices $\mathbf{M}^{(k)}$ peut se réécrire comme :

$$\mathbf{M}^{(k)} = \mathbf{LQD}^{(k)}\mathbf{Q}^T\mathbf{L}^{-1}, \quad \forall k \in [1, K]_{\mathbb{N}}. \tag{5.2}$$

En posant alors

$$\mathbf{S}^{(k)} = \mathbf{L}^{-1}\mathbf{M}^{(k)}\mathbf{L}, \quad \forall k \in [1, K]_{\mathbb{N}}, \tag{5.3}$$

il vient

$$\mathbf{S}^{(k)} = \mathbf{QD}^{(k)}\mathbf{Q}^T, \quad \forall k \in [1, K]_{\mathbb{N}}. \tag{5.4}$$

Sans bruit, les matrices $\mathbf{S}^{(k)}$ sont donc symétriques. De plus, comme les éléments diagonaux des matrices $\mathbf{D}^{(k)}$ sont strictement positifs $\forall k \in [1, K]_{\mathbb{N}}$, les éléments diagonaux des matrices $\mathbf{S}^{(k)}$ sont aussi positifs. Les matrices $\mathbf{S}^{(k)}$ forment donc un ensemble de matrices définies positives.

L'idée est alors de chercher dans un premier temps une matrice $\mathbf{L}$ réalisant la symétrisation conjointe des matrices $\mathbf{M}^{(k)}$. Cette première étape de symétrisation nous permet de transformer le problème de DCS en un problème plus simple de DCC-O résumé par l'équation (5.4). Nous préservons ainsi la propriété de positivité des matrices diagonales. La deuxième et dernière étape de l'algorithme consistera donc à résoudre ce problème de diagonalisation conjointe sous contrainte de positivité des valeurs propres.

Cette stratégie originale de résolution en deux étapes permet aussi de résoudre le problème de DCS sans contrainte de positivité. En effet, suite à l'étape de symétrisation, n'importe quel algorithme de DCC-O (ou de DCC) permet de diagonaliser l'ensemble de matrices.

5.1.1 Principe de la symétrisation conjointe

Nous proposons pour symétriser les matrices $\mathbf{M}^{(k)}$ un algorithme de balayage par paire. Le principe est d'estimer $\mathbf{L}$ comme

$$\mathbf{L} = \left[\prod_{j=1}^{R-1} \left[\prod_{i=j+1}^{R} \mathbf{L}^{(i,j)} \right] \mathbf{\Lambda}^{(j)} \right] \mathbf{\Lambda}^{(R)}, \tag{5.5}$$

où les matrices $\mathbf{L}^{(i,j)}$ sont des matrices triangulaires inférieures élémentaires de même type que celles définies dans le chapitre 3 (proposition 1) et les matrices $\mathbf{\Lambda}^{(j)}$ sont des matrices diagonales élémentaires. C'est-à-dire qu'elles sont égales à la matrice identité à l'exception du terme $\Lambda^{(j)}_{j,j}$.

Contrairement aux algorithmes de diagonalisation conjointe, il n'est pas possible de symétriser un ensemble de matrices à une matrice diagonale près (sauf si cette matrice a tous ses éléments diagonaux égaux). C'est pour cette raison que nous ne pouvons pas imposer ici à la matrice $\mathbf{L}$ d'avoir ses termes diagonaux égaux à 1.

La méthode de symétrisation présentée ici est basée sur des mises à jour multiplicatives. Nous nommerons la matrice de mise à jour $\mathbf{Y}$. Par conséquent, à chaque itération, nous avons

$$\mathbf{N}^{(k)} \leftarrow \mathbf{Y}^{-1}\mathbf{N}^{(k)}\mathbf{Y} \ \forall k \in [1;K]_{\mathbb{N}}, \tag{5.6}$$

où $\mathbf{N}^{(k)} = \mathbf{M}^{(k)} \ \forall k \in [1;K]_{\mathbb{N}}$ à la première itération. De plus, nous proposons le critère de symétrisation suivant

$$\begin{cases} C_{\text{sym}}(\mathbf{Y}) = \displaystyle\sum_{k=1}^{K}\sum_{p=1}^{R-1}\sum_{q=p+1}^{R}(N_{p,q}^{(k)\prime} - N_{q,p}^{(k)\prime})^2 \\ \text{où} \ \ \mathbf{N}^{(k)\prime} = \mathbf{Y}^{-1}\mathbf{N}^{(k)}\mathbf{Y}. \end{cases} \tag{5.7}$$

Lors d'une itération, le critère (5.7) est minimisé par les $R(R-1)/2$ matrices $\mathbf{L}^{(i,j)}$ et par les R matrices $\mathbf{\Lambda}^{(j)}$ et $\mathbf{\Lambda}^{(R)}$.

5.1.2 Construction des matrices $\mathbf{S}^{(k)}$ et estimation de L

D'après l'équation (5.5), l'ensemble de matrices $\mathbf{N}^{(k)}$ est mis à jour successivement par les $R(R-1)/2$ matrices $\mathbf{L}^{(i,j)}$ pour chaque paire d'indices (i,j), $j \in [1;R-1]_{\mathbb{N}}$ et $i \in [1;R]_{\mathbb{N}}$ à une itération donnée. De plus, lorsque $i = R$ l'ensemble des matrices est mis à jour par la matrice $\mathbf{\Lambda}^{(j)}$ et à la fin du balayage (*i.e.* $i = R$ et $j = R-1$) une mise à jour par $\mathbf{\Lambda}^{(R)}$ est effectuée.

Dans le but de rendre l'ensemble de matrices $\mathbf{N}^{(k)}$ le plus symétrique possible, nous choisissons la matrice minimisant de manière optimale le critère (5.7).

Considérons, en premier lieu, une mise à jour par une matrice $\mathbf{L}^{(i,j)}$ (*i.e.* $\mathbf{Y} = \mathbf{L}^{(i,j)}$). Nous vérifions facilement que la mise à jour n'affecte que les lignes i et les colonnes j de l'ensemble de matrices à symétriser et donc

$$\begin{aligned} C_{\text{sym}}(\mathbf{L}^{(i,j)}) = \sum_{k=1}^{K}\Bigg(\sum_{\substack{p\neq i\\ p\neq j}}\Big((N_{p,i}^{(k)} - N_{i,p}^{(k)\prime})^2 + (N_{p,j}^{(k)\prime} - N_{j,p}^{(k)})^2\Big) \ + \\ (N_{i,j}^{(k)\prime} - N_{j,i}^{(k)})^2\Bigg) + \Delta \end{aligned} \tag{5.8}$$

où Δ est une constante ne dépendant pas de $\mathbf{L}^{(i,j)}$ et

$$\begin{cases} \forall\ p \neq j, \quad N_{i,p}^{(k)\prime} = -L_{i,j}^{(i,j)}N_{j,p}^{(k)} + N_{i,p}^{(k)}, \\ \forall\ p \neq i, \quad N_{p,j}^{(k)\prime} = L_{i,j}^{(i,j)}N_{p,i}^{(k)} + N_{p,j}^{(k)}, \\ N_{i,j}^{(k)\prime} = -L_{i,j}^{(i,j)2}N_{j,i}^{(k)} + L_{i,j}^{(i,j)}(N_{i,i}^{(k)} - N_{j,j}^{(k)}) + N_{i,j}^{(k)}. \end{cases} \tag{5.9}$$

Ainsi, nous avons :

$$C_{\text{sym}}(\mathbf{L}^{(i,j)}) = P(L_{i,j}^{(i,j)}) = \alpha_4 L_{i,j}^{(i,j)4} + \alpha_3 L_{i,j}^{(i,j)3} + \alpha_2 L_{i,j}^{(i,j)2} + \alpha_1 L_{i,j}^{(i,j)} + \alpha_0 + \Delta \tag{5.10}$$

avec :

$$\alpha_4 = \sum_{k=1}^{K} (N_{j,i}^{(k)})^2; \tag{5.11}$$

$$\alpha_3 = -2 \sum_{k=1}^{K} N_{j,i}^{(k)} (N_{i,i}^{(k)} - N_{j,j}^{(k)}); \tag{5.12}$$

$$\alpha_2 = \sum_{k=1}^{K} \Bigg(- 2N_{j,i}^{(k)} (N_{i,j}^{(k)} - N_{j,i}^{(k)}) + (N_{i,i}^{(k)} - N_{j,j}^{(k)})^2 + \sum_{\substack{p \neq i \\ p \neq j}} \Big((N_{j,p}^{(k)})^2 + (N_{p,i}^{(k)})^2 \Big) \Bigg); \tag{5.13}$$

$$\alpha_1 = 2 \sum_{k=1}^{K} \Bigg((N_{i,i}^{(k)} - N_{j,j}^{(k)})(N_{i,j}^{(k)} - N_{j,i}^{(k)}) + \sum_{\substack{p \neq i \\ p \neq j}} \Big(N_{p,i}^{(k)} (N_{p,j}^{(k)} - N_{j,p}^{(k)}) - N_{j,p}^{(k)} (N_{i,p}^{(k)} - N_{p,i}^{(k)}) \Big) \Bigg). \tag{5.14}$$

Nous choisirons donc $L_{i,j}^{(i,j)}$ comme la racine réelle de la dérivée du polynôme P minimisant P. Ce qui revient à rechercher les racines d'un polynôme d'ordre 3, la solution de notre problème se calcule donc de manière analytique.

Intéressons nous maintenant au calcul des matrices $\mathbf{\Lambda}^{(j)}$. Le principe est similaire. Lors de la mise à jour par la matrice élémentaire $\mathbf{\Lambda}^{(j)}$ ($\mathbf{Y} = \mathbf{\Lambda}^{(j)}$ dans (5.7)), nous avons cette fois-ci :

$$C_{\text{sym}}(\mathbf{\Lambda}^{(j)}) = \sum_{k=1}^{K} \sum_{p \neq j} (N_{j,p}^{(k)'} - N_{p,j}^{(k)'})^2 + \Delta' \tag{5.15}$$

$$= \sum_{k=1}^{K} \sum_{p \neq j} \left(\frac{(N_{j,p}^{(k)})^2}{\Lambda_{j,j}^{(j)}} - \Lambda_{j,j}^{(j)} (N_{p,j}^{(k)})^2 \right)^2 + \Delta' \tag{5.16}$$

$$= f(\Lambda_{j,j}^{(j)}) + \Delta' \tag{5.17}$$

où Δ' est une constante ne dépendant pas de $\mathbf{\Lambda}^{(j)}$. Et donc :

$$\forall \Lambda_{j,j}^{(j)} \neq 0, \ \frac{\partial f(\Lambda_{j,j}^{(j)})}{\partial \Lambda_{j,j}^{(j)}} = 0 \Leftrightarrow \Lambda_{j,j}^{(j)4} = \frac{\sum_{k=1}^{K} \sum_{p \neq j} (N_{j,p}^{(k)})^2}{\sum_{k=1}^{K} \sum_{p \neq j} (N_{p,j}^{(k)})^2}. \tag{5.18}$$

f étant paire, nous choisissons

$$\Lambda_{j,j}^{(j)} = \left(\frac{\sum_{k=1}^{K} \sum_{p \neq j} (N_{j,p}^{(k)})^2}{\sum_{k=1}^{K} \sum_{p \neq j} (N_{p,j}^{(k)})^2} \right)^{\frac{1}{4}}. \tag{5.19}$$

Remarque : pour symétriser un ensemble de matrices, il est aussi possible d'utiliser un critère simplifié du type :

$$C'_{\text{sym}}(\mathbf{Y}) = \sum_{k=1}^{K} (N_{i,j}^{(k)'} - N_{j,i}^{(k)'})^2. \tag{5.20}$$

Nous obtenons alors pour $\Lambda_{j,j}^{(j)}$:

$$\Lambda_{j,j}^{(j)} = \left(\frac{\sum_{k=1}^{K} (N_{j,i}^{(k)})^2}{\sum_{k=1}^{K} (N_{i,j}^{(k)})^2} \right)^{\frac{1}{4}}. \tag{5.21}$$

Pour estimer le terme $L_{i,j}^{(i,j)}$, il suffit de remplacer dans le polynôme (5.8) α_2 par

$$\alpha_2 = \sum_{k=1}^{K} \left(-2N_{j,i}^{(k)}(N_{i,j}^{(k)} - N_{j,i}^{(k)}) + (N_{i,i}^{(k)} - N_{j,j}^{(k)})^2 \right); \tag{5.22}$$

et α_1 par

$$\alpha_1 = 2 \sum_{k=1}^{K} (N_{i,i}^{(k)} - N_{j,j}^{(k)})(N_{i,j}^{(k)} - N_{j,i}^{(k)}). \tag{5.23}$$

Cependant nous avons constater que l'utilisation de ce critère donne de moins bons résultats en pratique. Comme dit dans le chapitre 3, ce type de critère est justifié par le fait qu'un ou plusieurs éléments hors diagonaux de chaque matrice $\mathbf{N}^{(k)}$ sont modifiés deux fois par la mise à jour. C'est ici le cas pour la mise à jour par la matrice $\mathbf{L}^{(i,j)}$ où le terme $N_{i,j}^{(k)}$ est affecté deux fois. Cependant lors d'une mise à jour par la matrice $\mathbf{\Lambda}^{(j)}$, la $j^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne sont les seules modifiées et par conséquent aucun terme hors diagonal n'est affecté deux fois.

La procédure algorithmique de la méthode de symétrisation proposée est résumée dans la table 7.

À la fin du processus itératif de symétrisation, nous avons $\mathbf{N}^{(k)} = \mathbf{S}^{(k)}\ \forall k \in [1; K]_{\mathbb{N}}$ lorsque la matrice $\mathbf{L}$ est parfaitement estimée. L'algorithme que nous choisissons pour estimer la matrice $\mathbf{Q}$ est l'algorithme JD-BGL proposé par Pham dans [114]. Cet algorithme est parfaitement adapté au problème de diagonalisation de matrices définies positives. En effet, à chaque mise à jour, les éléments diagonaux des matrices $\mathbf{N}^{(k)}$ sont bornés entre la plus petite et la plus grande valeur propre de chacune de ces dernières. De plus, cet algorithme fournit de très bons résultats et sa convergence est démontrée. Nous appelons l'algorithme proposé JDJS pour « Joint Diagonalisation based on Joint Symetrisation ».

En pratique, l'ensemble des matrices $\mathbf{M}^{(k)}$ ne peut être parfaitement symétrisé (à cause d'un écart entre le modèle et les données mesurées par exemple). Les matrices $\mathbf{N}^{(k)}$ ne sont alors pas forcément parfaitement symétriques définies positives et la conservation de la positivité des éléments diagonaux des matrices $\mathbf{N}^{(k)}$ n'est pas forcément possible. Cependant, nous pouvons fabriquer un ensemble de matrices parfaitement symétriques et définies positives de la manière suivante :

$$\begin{aligned} \mathbf{S}_2^{(k)} &= \mathbf{S}^{(k)T}\mathbf{S}^{(k)} \\ &= \mathbf{Q}\mathbf{D}^{(k)}\mathbf{D}^{(k)}\mathbf{Q}^T. \end{aligned} \tag{5.24}$$

Algorithme 7 Algorithme de symétrisation

Soit S_C un critère d'arrêt et It_{max} le nombre maximal d'itérations ;
Initialisation de $\mathbf{L}$ avec $\mathbf{I}$ ou n'importe quelle choix judicieux ;
$it \leftarrow 1$;
while S_C est $faux$ et $it < It_{max}$ **do**
  for $j = 1$ to $R - 1$ **do**
    for $i = j + 1$ to R **do**
      Estimer $L_{i,j}^{(i,j)}$ comme la racine réelle de la dérivée de $P(L_{i,j}^{(i,j)})$ minimisant $P(L_{i,j}^{(i,j)})$;
      Construire $\mathbf{L}^{(i,j)}$;
      Mettre à jour $\mathbf{L} \leftarrow \mathbf{L}\mathbf{L}^{(i,j)}$;
      Mettre à jour $\mathbf{N}^{(k)} \leftarrow (\mathbf{L}^{(i,j)})^{-1}\mathbf{N}^{(k)}\mathbf{L}^{(i,j)}$ $\forall k \in [1; K]_{\mathbb{N}}$;
    end for
    Estimer $\Lambda_{j,j}^{(j)}$ à l'aide de (5.19) ;
    Construire $\mathbf{\Lambda}^{(j)}$;
    Mettre à jour $\mathbf{L} \leftarrow \mathbf{L}\mathbf{\Lambda}^{(j)}$;
    Mettre à jour $\mathbf{N}^{(k)} \leftarrow (\mathbf{\Lambda}^{(j)})^{-1}\mathbf{N}^{(k)}\mathbf{\Lambda}^{(j)}$ $\forall k \in [1; K]_{\mathbb{N}}$;
  end for
  Estimer $\Lambda_{R,R}^{(R)}$ à l'aide de (5.19) ;
  Construire $\mathbf{\Lambda}^{(R)}$;
  Mettre à jour $\mathbf{L} \leftarrow \mathbf{L}\mathbf{\Lambda}^{(R)}$;
  Mettre à jour $\mathbf{N}^{(k)} \leftarrow (\mathbf{\Lambda}^{(R)})^{-1}\mathbf{N}^{(k)}\mathbf{\Lambda}^{(R)}$ $\forall k \in [1; K]_{\mathbb{N}}$;
  Mettre à jour S_C ;
  $it \leftarrow it + 1$;
end while

Ainsi, n'importe quel algorithme de DCC basé sur des mises à jour multiplicatives conservera par construction la positivité des éléments diagonaux des matrices $\mathbf{S}_2^{(k)}$. Pour retrouver les valeurs propres de notre problème, il suffira alors de prendre la racine carrée positive des éléments des matrices diagonales estimées. Cette stratégie peut être appliquée pour n'importe quel problème de DCS (sans obligation de contrainte de positivité sur les valeurs propres) lorsque nous nous intéressons uniquement à l'estimation de la matrice de vecteurs propres. Cette stratégie est particulièrement utile pour améliorer l'estimation des valeurs propres lorsque la contrainte de positivité sur les valeurs propres est avérée. Nous choisissons une fois encore pour l'étape de diagonalisation JD-BGL et nous appelons ce second algorithme JDJS2.

La complexité numérique d'une itération de l'étape de symétrisation est

$$\Gamma[sym] \simeq 2KR^3 \tag{5.25}$$

et la complexité de JD-BGL pour l'étape de diagonalsiation est

$$\Gamma[JD-BGL] \simeq 4KR^3. \tag{5.26}$$

Remarque : la matrice $\mathbf{L}^{(i,j)}$ pouvant être paramétrée grâce à une décomposition polaire, l'étape de symétrisation peut aussi être réalisée à l'aide d'une matrice symétrique. En pratique, il semble que symétriser avec une matrice triangulaire ou symétrique donne des résultats similaires. Nous donnerons les détails du calcul de cette matrice dans la section A.3 de l'annexe.

5.2 Méthodes alternées de DCS sous contraintes de non-négativité

Nous proposons ici de résoudre le problème de DCS en utilisant des méthodes alternées de type ALS et ADMM. Comme expliqué au début du chapitre 3, les matrices $\mathbf{M}^{(k)}$ peuvent être vues comme les tranches d'un tenseur d'ordre trois que nous notons $\mathcal{M}$. Le problème de DCS peut donc se réécrire comme la décomposition CP de rang R d'un tel tenseur :

$$\mathcal{M} = \mathcal{I}_R \times_1 \mathbf{A} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{C}, \tag{5.27}$$

où $\mathbf{A}^{(2)} = \mathbf{A}^{-T}$ et où la matrice $\mathbf{C} \in (\mathbb{R}^+)^{K\times R}$ contient la diagonale des matrices $\mathbf{D}^{(k)}$ sur ses lignes.

Les algorithmes de DCS étant généralement basés sur des mises à jour multiplicatives, nous proposons en premier lieu un algorithme de type ALS sans contraintes afin d'observer son comportement par rapport aux algorithmes de DCS classiques. L'ALS et l'ADMM existant déjà [19, 23, 33, 80] pour le problème de décomposition CP non-négative, l'originalité des méthodes proposées réside principalement dans la prise en compte des contraintes induites par la compression par SVD du problème de décomposition CP non-négative.

5.2.1 ALS sans contraintes de non-négativité

Nous choisissons de minimiser ici l'erreur quadratique de reconstruction du tenseur

$$\Psi(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}) = ||\mathcal{M} - \mathcal{I}_R \times_1 \widehat{\mathbf{A}} \times_2 \widehat{\mathbf{A}}^{(2)} \times_3 \widehat{\mathbf{C}}||_F^2 \tag{5.28}$$

où $\widehat{\mathbf{A}}$, $\widehat{\mathbf{A}}^{(2)}$ et $\widehat{\mathbf{C}}$ sont les estimations de $\mathbf{A}$, $\mathbf{A}^{(2)}$ et $\mathbf{C}$. Nous rappelons que la procédure de l'ALS consiste à minimiser alternativement à chaque itération :

$$\Psi_1(\widehat{\mathbf{A}}) = ||\mathbf{M}_{(1)} - \widehat{\mathbf{A}}(\widehat{\mathbf{C}} \odot \widehat{\mathbf{A}}^{(2)})^T||_F^2 \tag{5.29}$$

en fonction de $\widehat{\mathbf{A}}$,

$$\Psi_2(\widehat{\mathbf{A}}^{(2)}) = ||\mathbf{M}_{(2)} - \widehat{\mathbf{A}}^{(2)}(\widehat{\mathbf{A}} \odot \widehat{\mathbf{C}})^T||_F^2 \tag{5.30}$$

en fonction de $\widehat{\mathbf{A}}^{(2)}$ et

$$\Psi_3(\widehat{\mathbf{C}}) = ||\mathbf{M}_{(3)} - \widehat{\mathbf{C}}(\widehat{\mathbf{A}}^{(2)} \odot \widehat{\mathbf{A}})^T||_F^2 \tag{5.31}$$

en fonction de $\widehat{\mathbf{C}}$. Dans le but de tenir compte de la contrainte $\widehat{\mathbf{A}}^{-T} = \widehat{\mathbf{A}}^{(2)}$, nous modifions la fonction (5.28) de la manière suivante :

$$\varphi\big(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}\big) = \Psi\big(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}\big) + \alpha||\widehat{\mathbf{A}}^{(2)}\widehat{\mathbf{A}}^T - I_R||_F^2, \tag{5.32}$$

où $\alpha \in \mathbb{R}^+$ est un paramètre de régulation dont le choix sera expliqué dans la partie simulations numériques. Finalement, l'expression des variables à estimer est obtenue en dérivant (5.32) alternativement par rapport à $\widehat{\mathbf{A}}$, $\widehat{\mathbf{A}}^{(2)}$ et $\widehat{\mathbf{C}}$ à chaque itération. Ainsi, les matrices facteurs du tenseur $\mathcal{M}$ sont estimées de la manière suivante :

$$\begin{aligned}
\widehat{\mathbf{A}}_{(it+1)} &= \big(\mathbf{M}_{(1)}\big(\widehat{\mathbf{C}}_{(it)} \odot \widehat{\mathbf{A}}^{(2)}_{(it)}\big) + \alpha\widehat{\mathbf{A}}^{(2)}_{(it)}\big) \\
&\quad \big(\big(\widehat{\mathbf{C}}_{(it)} \odot \widehat{\mathbf{A}}^{(2)}_{(it)}\big)^T\big(\widehat{\mathbf{C}}_{(it)} \odot \widehat{\mathbf{A}}^{(2)}_{(it)}\big) + \alpha\widehat{\mathbf{A}}^{(2)T}_{(it)}\widehat{\mathbf{A}}^{(2)}_{(it)}\big)^{-1} && (5.33) \\
\widehat{\mathbf{A}}^{(2)}_{(it+1)} &= \big(\mathbf{M}_{(2)}\big(\widehat{\mathbf{C}}_{(it)} \odot \widehat{\mathbf{A}}_{(it+1)}\big) + \alpha\widehat{\mathbf{A}}_{(it+1)}\big) \\
&\quad \big(\big(\widehat{\mathbf{C}}_{it} \odot \widehat{\mathbf{A}}_{(it+1)}\big)^T\big(\widehat{\mathbf{C}}_{(it)} \odot \widehat{\mathbf{A}}_{(it+1)}\big) + \alpha\widehat{\mathbf{A}}^T_{(it+1)}\widehat{\mathbf{A}}_{(it+1)}\big)^{-1} && (5.34) \\
\widehat{\mathbf{C}}_{(it+1)} &= \mathbf{M}_{(3)}\big(\widehat{\mathbf{A}}^{(2)}_{(it+1)} \odot \widehat{\mathbf{A}}_{(it+1)}\big)^{-T} && (5.35)
\end{aligned}$$

Évidement, l'ordre d'estimation des matrices peut être inter-changé à chaque itération.

Nous avons ainsi proposé un algorithme de type ALS adapté au problème de DCS.

5.2.2 ALS sous contraintes de non-négativité

En reprenant le critère (5.32) et ajoutant maintenant les contraintes (5.1), nous obtenons le problème d'optimisation sous contraintes suivant :

Problème 5.2.1.

$$\begin{aligned}
&\underset{\widehat{\mathbf{A}},\widehat{\mathbf{A}}^{(2)},\widehat{\mathbf{C}}}{\text{minimiser}}\ \varphi\big(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}\big) \\
&\textit{contraint à} && (5.36) \\
&\qquad \mathbf{V}\widehat{\mathbf{A}} \succeq 0 \\
&\qquad \mathbf{US}\widehat{\mathbf{A}}^{(2)} \succeq 0 \\
&\qquad \widehat{\mathbf{C}} \succeq 0. && (5.37)
\end{aligned}$$

La méthode proposée ici est en partie inspirée de [87] où une méthode de type ALS projetée est utilisée (page 49 de cette thèse). Lors d'une itération, nous calculons directement après l'estimation de la matrice $\widehat{\mathbf{A}}$, la matrice

$$\mathbf{W}^{(1)} = \mathbf{V}\widehat{\mathbf{A}}. \tag{5.38}$$

Puis, nous projetons la matrice $\mathbf{W}^{(1)}$ sur $\mathbb{R}^+$ de la manière suivante :

$$\mathbf{W}^{(1)} \leftarrow [\mathbf{W}^{(1)}]_+. \tag{5.39}$$

Finalement, nous obtenons pour la matrice $\widehat{\mathbf{A}}$:

$$\widehat{\mathbf{A}} = \mathbf{V}^T\mathbf{W}^{(1)}. \tag{5.40}$$

Pour calculer la matrice $\widehat{\mathbf{A}}^{(2)}$, nous procédons de la même manière en calculant cette fois-ci la matrice

$$\mathbf{W}^{(2)} = \mathbf{U}\mathbf{S}\widehat{\mathbf{A}}^{(2)}, \tag{5.41}$$

puis nous effectuons l'étape de projection

$$\mathbf{W}^{(2)} \leftarrow [\mathbf{W}^{(2)}]_+ \tag{5.42}$$

et enfin nous avons

$$\widehat{\mathbf{A}}^{(2)} = \mathbf{S}^{-1}\mathbf{U}^T\mathbf{W}^{(2)}. \tag{5.43}$$

La matrice $\mathbf{C}$ est, quant à elle, directement projetée de la manière suivante :

$$\widehat{\mathbf{C}} \leftarrow [\widehat{\mathbf{C}}]_+. \tag{5.44}$$

5.2.3 ADMM adaptée au problème de DCS

L'ADMM est une manière élégante de résoudre des problèmes d'optimisation sous contraintes. Comme présentée dans le chapitre 2, cette méthode a été adaptée au problème de décomposition CP non-négative dans [80]. De la même manière, nous introduisons trois variables auxiliaires $\tilde{\mathbf{C}}$, $\tilde{\mathbf{W}}$ et $\tilde{\mathbf{X}}$ ainsi que la fonction de coût suivante :

$$\varphi_{\text{ADMM}}\big(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}, \tilde{\mathbf{C}}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}\big) = \frac{1}{2}\varphi\big(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}\big) + i(\tilde{\mathbf{C}}) + i(\tilde{\mathbf{W}}) + i(\tilde{\mathbf{X}}) \tag{5.45}$$

avec $i(\bullet)$ la fonction indicatrice définie comme dans la section 2.3.5. Résoudre le problème (5.36) revient donc à résoudre le problème :

Problème 5.2.2.

$$\underset{\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}, \tilde{\mathbf{C}}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}}{\text{minimiser}} \varphi_{ADMM}\big(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}}, \tilde{\mathbf{C}}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}\big)$$

$$\textit{contraint à} \tag{5.46}$$

$$\widehat{\mathbf{C}} = \tilde{\mathbf{C}}$$

$$\mathbf{U}\mathbf{S}\widehat{\mathbf{A}}^{(2)} = \tilde{\mathbf{W}}$$

$$\mathbf{V}\widehat{\mathbf{A}} = \tilde{\mathbf{X}}. \tag{5.47}$$

Le Lagrangien augmenté d'un tel problème est donc

$$\begin{aligned}\mathrm{L}_\rho(\widehat{\mathbf{A}},\widehat{\mathbf{A}}^{(2)},\widehat{\mathbf{C}},\tilde{\mathbf{C}},\tilde{\mathbf{W}},\tilde{\mathbf{X}},\mathbf{\Lambda}) &= \varphi_{\mathrm{ADMM}}(\widehat{\mathbf{A}},\widehat{\mathbf{A}}^{(2)},\widehat{\mathbf{C}},\tilde{\mathbf{C}},\tilde{\mathbf{W}},\tilde{\mathbf{X}}) + \\ &\quad \mathsf{trace}\{\mathbf{\Lambda}^{(1)T}(\widehat{\mathbf{C}}-\tilde{\mathbf{C}})\} + \frac{\rho_1}{2}||\widehat{\mathbf{C}}-\tilde{\mathbf{C}}||_F^2 + \\ &\quad \mathsf{trace}\{\mathbf{\Lambda}^{(2)T}(\mathbf{US}\widehat{\mathbf{A}}^{(2)}-\tilde{\mathbf{W}})\} + \frac{\rho_2}{2}||\mathbf{US}\widehat{\mathbf{A}}^{(2)}-\tilde{\mathbf{W}}||_F^2 + \\ &\quad \mathsf{trace}\{\mathbf{\Lambda}^{(3)T}(\mathbf{V}\widehat{\mathbf{A}}-\tilde{\mathbf{X}})\} + \frac{\rho_3}{2}||\mathbf{V}\widehat{\mathbf{A}}-\tilde{\mathbf{X}}||_F^2, \qquad (5.48)\end{aligned}$$

où $\rho = [\rho_1;\rho_2;\rho_3]$ et $\mathbf{\Lambda} = \left[\mathbf{\Lambda}^{(1)};\mathbf{\Lambda}^{(2)};\mathbf{\Lambda}^{(3)}\right]$. L'expression des variables à estimer est alors obtenue en dérivant alternativement (5.48) par rapport à chacune d'entre elles, ainsi lors d'une itération, nous avons :

$$\begin{aligned}\widehat{\mathbf{A}}_{(it+1)} &= \big(\mathbf{M}_{(1)}(\widehat{\mathbf{C}}_{(it)}\odot\widehat{\mathbf{A}}^{(2)}_{(it)}) + \alpha\widehat{\mathbf{A}}^{(2)}_{(it)} + \rho_3\mathbf{V}^T\tilde{\mathbf{X}}_{(it)} - \mathbf{V}^T\mathbf{\Lambda}^{(3)}\big) \\ &\quad \big((\widehat{\mathbf{C}}_{(it)}\odot\widehat{\mathbf{A}}^{(2)}_{(it)})^T(\widehat{\mathbf{C}}_{(it)}\odot\widehat{\mathbf{A}}^{(2)}_{(it)}) + \alpha\widehat{\mathbf{A}}^{(2)T}_{(it)}\widehat{\mathbf{A}}^{(2)}_{(it)} + \rho_3\mathbf{I}_R\big)^{-1} \qquad (5.49)\\ \mathsf{vec}\{\widehat{\mathbf{A}}^{(2)}_{(it+1)}\} &= \big((\mathbf{K}_{(it)}\otimes\mathbf{I}_R) + (\mathbf{I}_R\otimes\rho_2\mathbf{SS})\big)^{-1}\mathbf{z}_{(it)} \qquad (5.50)\\ \widehat{\mathbf{C}}_{(it+1)} &= \big(\mathbf{M}_{(3)}(\widehat{\mathbf{A}}^{(2)}_{(it+1)}\odot\widehat{\mathbf{A}}_{(it+1)}) + \rho_1\tilde{\mathbf{C}}_{(it)} - \mathbf{\Lambda}^{(1)}\big) \\ &\quad \big((\widehat{\mathbf{A}}^{(2)}_{(it+1)}\odot\widehat{\mathbf{A}}_{(it+1)})^T(\widehat{\mathbf{A}}^{(2)}_{(it+1)}\odot\widehat{\mathbf{A}}_{(it+1)}) + \rho_1\mathbf{I}_R\big)^{-1} \qquad (5.51)\\ \tilde{\mathbf{C}}_{(it+1)} &= \left[\widehat{\mathbf{C}}_{(it+1)} + \frac{\mathbf{\Lambda}^{(1)}_{(it)}}{\rho_1}\right]_+ \qquad (5.52)\\ \tilde{\mathbf{W}}_{(it+1)} &= \left[\mathbf{US}\widehat{\mathbf{A}}^{(2)}_{(it+1)} + \frac{\mathbf{\Lambda}^{(2)}_{(it)}}{\rho_2}\right]_+ \qquad (5.53)\\ \tilde{\mathbf{X}}_{(it+1)} &= \left[\mathbf{V}\widehat{\mathbf{A}}_{(it+1)} + \frac{\mathbf{\Lambda}^{(3)}_{(it)}}{\rho_3}\right]_+ \qquad (5.54)\end{aligned}$$

avec, $\mathbf{z}_{(it)} = \mathsf{vec}\{\mathbf{M}_{(2)}(\widehat{\mathbf{C}}_{(it)}\odot\widehat{\mathbf{A}}_{(it+1)}) + \alpha\widehat{\mathbf{A}}_{(it+1)} - \mathbf{SU}^T\mathbf{\Lambda}^{(2)}_{(it)} + \rho_3\mathbf{SU}^T\tilde{\mathbf{W}}\}$ et $\mathbf{K}_{(it)} = (\widehat{\mathbf{C}}_{(it)}\odot\widehat{\mathbf{A}}_{(it+1)})^T(\widehat{\mathbf{C}}_{(it)}\odot\widehat{\mathbf{A}}_{(it+1)}) + \alpha\widehat{\mathbf{A}}^T_{(it+1)}\widehat{\mathbf{A}}_{(it+1)}$.

Comme précédemment, l'ordre d'estimation des variables peut être inter-changé d'une itération à l'autre. Nous pouvons noter que l'expression de la matrice $\widehat{\mathbf{A}}^{(2)}$ vectorisée est obtenue grâce à l'équation de Sylvester (1.13). Une fois la matrice $\widehat{\mathbf{A}}^{(2)}$ estimée sous forme vectorisée, il faut donc la reformer en matrice de taille $R\times R$.

Les matrices des multiplicateurs de Lagrange sont calculées comme

$$\begin{aligned}\mathbf{\Lambda}^{(1)}_{(it+1)} &= \mathbf{\Lambda}^{(1)}_{(it)} + \rho_1(\widehat{\mathbf{C}}_{(it+1)} - \tilde{\mathbf{C}}_{(it+1)}) \qquad (5.55)\\ \mathbf{\Lambda}^{(2)}_{(it+1)} &= \mathbf{\Lambda}^{(2)}_{it} + \rho_2(\mathbf{US}\widehat{\mathbf{A}}^{(2)}_{(it+1)} - \tilde{\mathbf{W}}_{(it+1)}) \qquad (5.56)\\ \mathbf{\Lambda}^{(3)}_{(it+1)} &= \mathbf{\Lambda}^{(3)}_{(it)} + \rho_3(\mathbf{V}\widehat{\mathbf{A}}_{(it+1)} - \tilde{\mathbf{X}}_{(it+1)}) \qquad (5.57)\end{aligned}$$

Les termes de pénalité ρ_1, ρ_2 et ρ_3 peuvent être laissés fixes à chaque itération. Néanmoins,

il est proposé dans [79] une méthode judicieuse pour calculer ces paramètres :

$$\rho_{q(it+1)} = \begin{cases} \rho_{q(it)}\tau^{(1)} \text{ si } ||\mathbf{P}_{(it)}^{(q)}||_F > \mu||\mathbf{Z}_{(it)}^{(q)}||_F \\ \rho_{q(it)}/\tau^{(2)} \text{ si } ||\mathbf{Z}_{(it)}^{(q)}||_F > \mu||\mathbf{P}_{(it)}^{(q)}||_F \\ \rho_{q(it)} \text{ sinon.} \end{cases}$$

Avec $q \in \{1,2,3\}$, $\mu = 8$, $\tau^{(1)} = 4$, $\tau^{(2)} = 2$,

$$\begin{cases} \mathbf{P}_{(it)}^{(1)} = \widehat{\mathbf{C}}_{(it)} - \tilde{\mathbf{C}}_{(it)}, \\ \mathbf{Z}_{(it)}^{(1)} = \rho_1(\tilde{\mathbf{C}}_{(it)} - \tilde{\mathbf{C}}_{(it-1)}), \\ \mathbf{P}_{(it)}^{(2)} = \mathbf{U}\mathbf{S}\widehat{\mathbf{A}}_{(it)}^{(2)} - \tilde{\mathbf{W}}_{(it)}, \\ \mathbf{Z}_{(it)}^{(2)} = \rho_2(\tilde{\mathbf{W}}_{(it)} - \tilde{\mathbf{W}}_{(it-1)}), \\ \mathbf{P}_{(it)}^{(3)} = \mathbf{V}\widehat{\mathbf{A}}_{(it)} - \tilde{\mathbf{X}}_{(it)}, \\ \mathbf{Z}_{(it)}^{(3)} = \rho_3(\tilde{\mathbf{X}}_{(it)} - \tilde{\mathbf{X}}_{(it-1)}). \end{cases}$$

Ainsi, moins les contraintes sont respectées, plus les valeurs de ρ_1, ρ_2 et ρ_3 sont augmentées et inversement dans le but d'éviter que les variables auxiliaires ne varient trop d'une itération à l'autre (lorsque les contraintes sont suffisamment respectées).

La complexité numérique de ces trois algorithmes est environ de $3R^3K$.

Notre but est d'intégrer les trois algorithmes de DCS précédemment présentés dans l'algorithme DIAG [22]. Nous appelons ainsi respectivement ces méthodes DIAG-ALS, DIAG-ALS+ et DIAG-ADMM.

5.3 Simulations numériques

Nous nous plaçons ici dans le cadre de la décomposition CP non-négative d'un tenseur d'ordre 3, de rang 5 et de dimensions $30 \times 30 \times 30$. Les colonnes des matrices facteurs sont générées de la manière suivante :

$$\forall q \in [1;3]_{\mathbb{N}},\ \forall r \in [2;5]_{\mathbb{N}}\ \mathbf{f}_r^{(q)} = \delta\mathbf{f}_1^{(q)} + (1-\delta)\mathbf{u}_r^{(q)}, \tag{5.58}$$

où la première colonne des matrices $\mathbf{F}^{(q)}$ et les vecteurs colonnes $\mathbf{u}_r^{(q)} \in \mathbb{R}^{I_q}$ sont générés à partir d'une loi uniforme sur $[0;1]$. Le coefficient $\delta \in [0;1]$ permet de régler le degré de colinéarité entre les différentes colonnes des matrices $\mathbf{F}^{(q)}$. Le tenseur à décomposer est construit de la manière suivante :

$$\mathcal{T} = \frac{\mathcal{I}_R \times_1 \mathbf{F}^{(1)} \times_2 \mathbf{F}^{(2)} \times_3 \mathbf{F}^{(3)}}{||\mathcal{I}_R \times_1 \mathbf{F}^{(1)} \times_2 \mathbf{F}^{(2)} \times_3 \mathbf{F}^{(3)}||_F} + \sigma\frac{\mathcal{E}}{||\mathcal{E}||_F} \tag{5.59}$$

avec $\mathcal{E}$ un tenseur généré selon une loi uniforme sur 0 et 1 représentant le bruit et σ un paramètre permettant de régler le RSB défini ici comme $-20\log_{10}(\sigma)$.

Pour mesurer l'erreur d'estimation des matrices facteurs, nous utilisons l'indice de performance suivant (après correction de l'indétermination d'échelle et de permutation) :

$$r_F = \frac{1}{3}\sum_{q=1}^{3}\frac{||\mathbf{F}^{(q)} - \widehat{\mathbf{F}}^{(q)}||_F}{||\mathbf{F}^{(q)}||_F}, \tag{5.60}$$

où $\widehat{\mathbf{F}}^{(q)}$ est l'estimation de la matrice $\mathbf{F}^{(q)}$.

Nous comparons les performances des méthodes proposées dans ce chapitre avec les algorithmes ADMoM [80], PROCO-ALS [87] et DIAG [22].

Pour l'étape de DCS de l'algorithme DIAG, nous utilisons les algorithmes JDTM [22] et JET-U [50]. Nous notons ces algorithmes DIAG-JDTM et DIAG-JET-U.

Nous rappelons que PROCO-ALS est un algorithme basé sur une compression du tenseur $\mathcal{T}$ par HOSVD. Nous choisissons de réduire par 2 les dimensions du tenseur initial, le tenseur obtenu suite à cette compression est alors de dimensions $15 \times 15 \times 15$.

Concernant ADMoM et PROCO-ALS les matrices $\widehat{\mathbf{F}}^{(1)}$, $\widehat{\mathbf{F}}^{(2)}$ et $\widehat{\mathbf{F}}^{(3)}$ sont initialisées aléatoirement à partir d'une loi uniforme sur $[0;1]$. Pour ADMoM, les autres paramètres sont initialisés comme dans [80].

Concernant DIAG-JDTM et DIAG-JET-U, la matrice diagonalisante est initialisée avec la matrice identité.

Nous intégrons l'algorithme JDJS dans l'algorithme DIAG [22] et l'algorithme JDJS2 dans SSD-CP [91] nous nommons respectivement ces algorithmes DIAG-JDJS et SSD-JDJS2. Concernant ces deux méthodes, les matrices $\mathbf{L}$ et $\mathbf{Q}$ sont initialisées avec la matrice identité.

Concernant DIAG-ALS, DIAG-ALS+ et DIAG-ADMM, les matrices $\widehat{\mathbf{A}}$ et $\widehat{\mathbf{A}}^{(2)}$ sont initialisées aléatoirement à partir d'une loi normale centrée et réduite. La matrice $\widehat{\mathbf{C}}$ est quant à elle initialisée à partir une loi uniforme sur $[0;1]$. Le coefficient α dans l'équation (5.32) est initialisé à 1 pour les trois méthodes alternées proposées. Ce dernier est divisé par 2 à chaque fois que

$$\frac{|\Psi(\widehat{\mathbf{A}}_{(it+1)}, \widehat{\mathbf{A}}^{(2)}_{(it+1)}, \widehat{\mathbf{C}}_{(it+1)}) - \Psi(\widehat{\mathbf{A}}_{(it)}, \widehat{\mathbf{A}}^{(2)}_{(it)}, \widehat{\mathbf{C}}_{(it)})|}{\Psi(\widehat{\mathbf{A}}_{(it+1)}, \widehat{\mathbf{A}}^{(2)}_{(it+1)}, \widehat{\mathbf{C}}_{(it+1)})} < 10^{-2}. \tag{5.61}$$

En effet, il arrive parfois que la fonction de coût (5.32) ne diminue plus car la contrainte $\widehat{\mathbf{A}}^{-T} = \widehat{\mathbf{A}}^{(2)}$ est trop forte. Plus spécifiquement, pour DIAG-ADMM, les termes de pénalité ρ_1, ρ_2, ρ_3 sont initialisés à 10^{-3}. Les matrices auxiliaires $\tilde{\mathbf{C}}$, $\tilde{\mathbf{W}}$, $\tilde{\mathbf{X}}$ et les matrices de multiplicateurs de Lagrange $\mathbf{\Lambda}^{(q)}$ sont initialisées à zéros.

Il est important de noter que pour les algorithmes DIAG et SSD-CP (peu importe l'algorithme de DCS utilisé), les matrices facteurs estimées ou les produits de Khatri-Rao des matrices facteurs estimés à l'issue de l'étape de DCS sont projetés en prenant leur valeur absolue.

Les fonctions de coût optimisées par les algorithmes ADMoM, PROCO-ALS, DIAG-ALS, DIAG-ALS+ et DIAG-ADMM peuvent ne plus évoluer pendant plusieurs itérations avant de décroître de nouveau. Nous proposons alors comme seul critère d'arrêt un nombre d'itérations suffisant pour atteindre un point stationnaire. Ainsi, le nombre d'itérations pour ADMoM, DIAG-ALS, DIAG-ALS+ et DIAG-ADMM est de 1000. Le nombre d'itérations pour PROCO-ALS est de 200. Le nombre d'itérations DIAG-JDTM et DIAG-JET-U est de 10. Enfin pour DIAG-JDJS et SSD-JDJS2, le nombre d'itérations est de 200 pour l'étape de symétrisation. DIAG-JDJS effectuera 10 itérations et SSD-JDJS2 effectuera 40 itérations pour l'étape de diagonalisation.

Nous présentons dans la suite deux études. La première dans le cas où les colonnes des matrices facteurs ne sont pas corrélées (*i.e.* $\delta = 0$). La seconde dans le cas où les colonnes des matrices facteurs ont un haut degré de corrélation ($\delta = 0,85$). Dans ces deux études,

nous faisons varier le RSB de 10 dB à 100 dB par pas de 10 dB. Pour chaque valeur du RSB, nous effectuons 200 simulations de Monte-Carlo et à chacune de ces simulations un nouveau tenseur est généré. Comme au chapitre précédent, nous considérons que plus la moyenne et l'écart type de r_F d'un algorithme sont petits, plus ce dernier est précis.

5.3.1 Scénario $\delta = 0$

La valeur médiane du critère r_F en fonction du RSB est tracée sur la figure 5.1. Nous pouvons observer que l'algorithme permettant d'obtenir ici la plus petite erreur est ADMoM pour tous les RSB. SSD-JDJS2 fournit des résultats similaires à ce dernier au-dessus de 40 dB. L'erreur médiane de PROCO-ALS est ici la même que ADMoM sous 80 dB. DIAG-ALS fournit quant à lui la moins bonne erreur médiane de 30 à 100 dB. Il est intéressant de remarquer que les deux autres algorithmes alternés proposés (DIAG-ALS+ et DIAG-ADMM) fournissent de meilleurs résultats que DIAG-ALS au-dessus de 30 dB, mais restent parmi les algorithmes fournissant la moins bonne estimation en médiane. DIAG-ALS+ a une erreur d'estimation comparable à celle de DIAG-JET-U, au-dessus de 20 dB. DIAG-JDJS et DIAG-JDTM procurent ici la même erreur d'estimation au-dessus de 20 dB, sous cette valeur DIAG-JDTM fournit une erreur médiane plus importante.

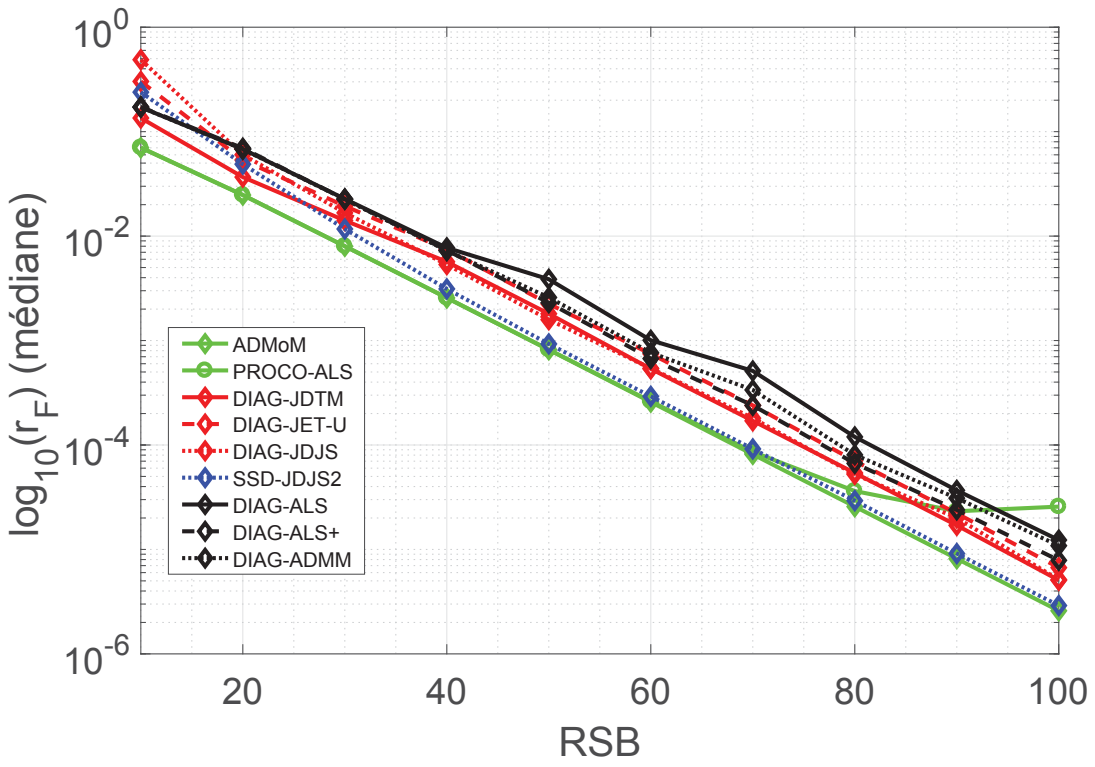

FIGURE 5.1: r_F médian en fonction du RSB ($\delta = 0$).

La moyenne du critère r_F en fonction du RSB est affichée sur la figure 5.2. DIAG-ALS est l'algorithme ayant la plus grande erreur en moyenne pour les valeurs du RSB supérieures à 30 dB. DIAG-ADMM et particulièrement DIAG-ALS+ fournissent de meilleurs

résultats que DIAG-ALS au-dessus de 30 dB. Toujours au-dessus de 30 dB, DIAG-ALS+ et PROCO-ALS ont quasiment la même erreur en moyenne. Au-dessus de 60 dB DIAG-JDJS fournit de moins bons résultats en moyenne que DIAG-JDTM et DIAG-JET-U. De 40 à 60 dB, il estime en moyenne aussi bien les matrices facteurs que DIAG-JDTM et DIAG-JET-U. Puis sous 40 dB, il fournit une mauvaise estimation des matrices facteurs. SSD-JDJS2 est l'algorithme le plus précis de 40 à 60 dB. Enfin, ADMoM fournit les meilleurs résultats pour les faibles RSB.

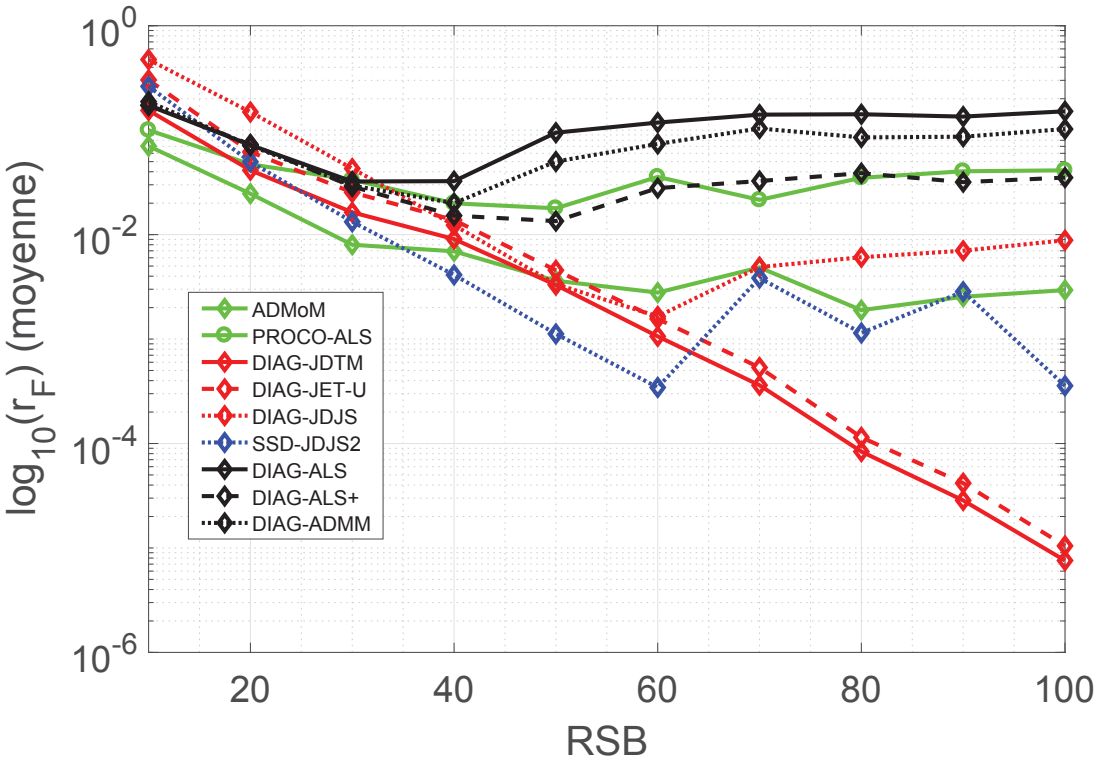

FIGURE 5.2: r_F moyen en fonction du RSB ($\delta = 0$).

L'écart type du critère r_F est donné à la figure 5.3. PROCO-ALS a globalement l'écart type le plus élevé pour toutes les valeurs du RSB. L'écart type de DIAG-ALS est le plus élevé au-dessus de 40 dB. Dans cette plage de valeurs, DIAG-ADMM a un écart type légèrement inférieur a celui de DIAG-ALS alors que DIAG-ALS+ a un écart type inférieur à ceux des deux méthodes alternées proposées mais reste parmi les algorithmes ayant l'écart type le plus élevé. Au-dessus de 30 dB, SSD-JDJS2 a un écart type largement inférieur à celui des autres algorithmes étudiés, DIAG-JDTM est le second algorithme à avoir le plus petit écart type suivi de près par DIAG-JET-U. DIAG-JDJS a un écart type légèrement supérieur que ADMoM au dessus de 70 dB, puis les tendances s'inversent entre 50 et 70 dB. ADMoM est l'algorithme fournissant le plus petit écart type pour les faibles RSB.

La figure 5.4 représente la médiane de l'erreur de reconstruction du tenseur $\mathcal{T}$ en fonction du nombre d'itérations pour les algorithmes ADMoM et PROCO-ALS à 60 dB. Nous pouvons remarquer que cette erreur ne diminue plus significativement à partir de

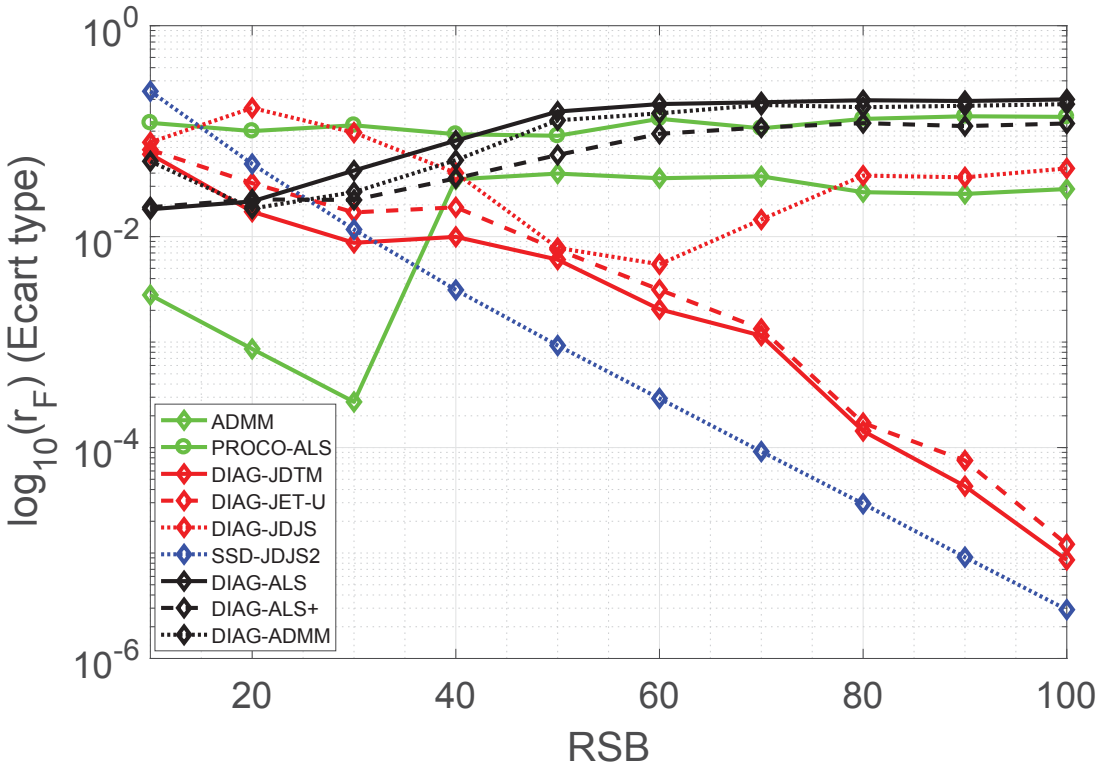

FIGURE 5.3: Écart type de r_F en fonction du RSB ($\delta = 0$).

400 itérations pour ADMoM et à partir de 100 itérations pour PROCO-ALS.

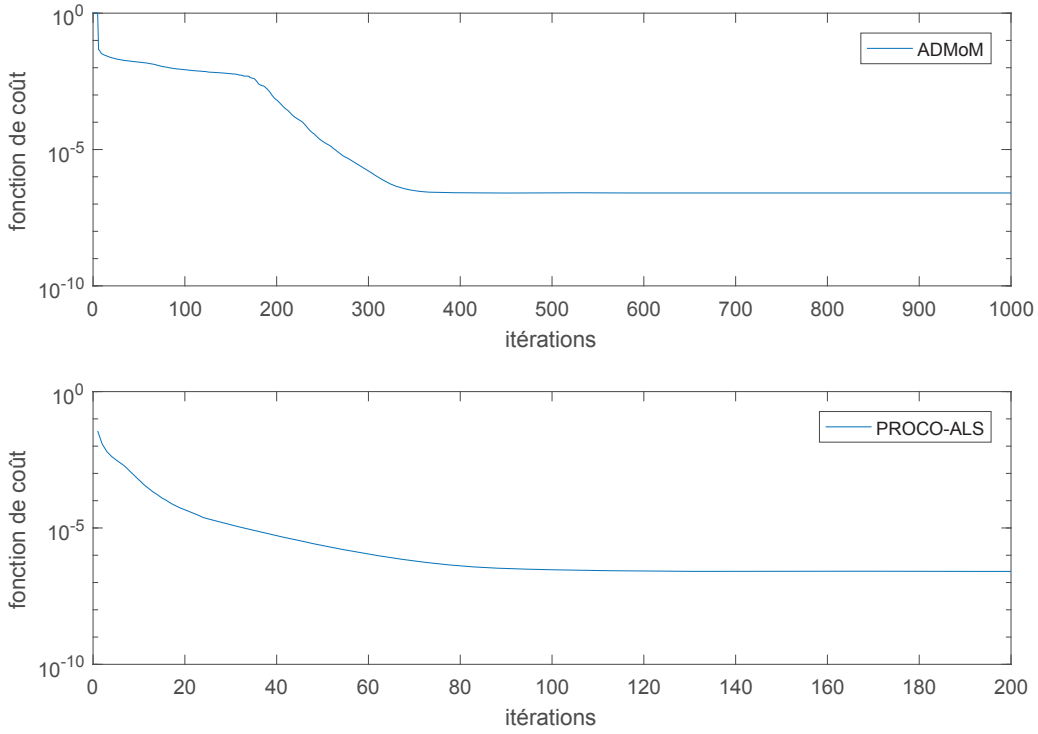

FIGURE 5.4: Évolution de la médiane de l'erreur de reconstruction $\Psi(\widehat{\mathbf{F}}^{(1)}, \widehat{\mathbf{F}}^{(2)}, \widehat{\mathbf{F}}^{(3)})$ du tenseur $\mathcal{T}$ en fonction des itérations avec $\delta = 0$ et à 60 dB.

La figure 5.5 représente la médiane de l'erreur de reconstruction du tenseur $\mathcal{M}$ en fonction du nombre d'itérations pour les algorithmes DIAG-ALS, DIAG-ALS+ et DIAG-ADMM à 60 dB. Cette erreur ne diminue quasiment plus à partir de 600 itérations pour DIAG-ALS, à partir de 100 itérations pour DIAG-ALS+ et à partir de 300 itérations pour DIAG-ADMM.

L'évolution du critère (3.63) pour l'algorithme JET-U et l'évolution du critère (3.14) pour l'algorithme JDTM au cours des itérations (à 60 dB) sont tracés sur la figure 5.6. Ces critères ne diminuent plus à partir de 4 itérations pour JDTM et à partir de 6 pour JET-U.

L'évolution du critère (5.7) pour l'étape de symétrisation de JDJS et l'évolution du critère (3.14) pour l'étape de diagonalisation de JDJS au cours des itérations (à 60 dB) sont tracés sur la figure 5.7. Ces critères ne diminuent plus à partir de 30 itérations pour l'étape de symétrisation. L'étape de diagonalisation de JDJS prend 4 itérations et celle de JDJS2 prend 6 itérations.

Pour calculer le coût calcul total de chacun des algorithmes proposés, nous multiplions le nombres d'itérations pour lequel les algorithmes ne font plus évoluer les fonctions de coût précédemment citées par la complexité numérique des processus itératifs de chacun des algorithmes. Il est important de noter que dans l'algorithme DIAG, le tenseur $\mathcal{M}$ est de dimensions $5 \times 5 \times 870$. Le coût de calcul total à 60 dB est donné dans le tableau

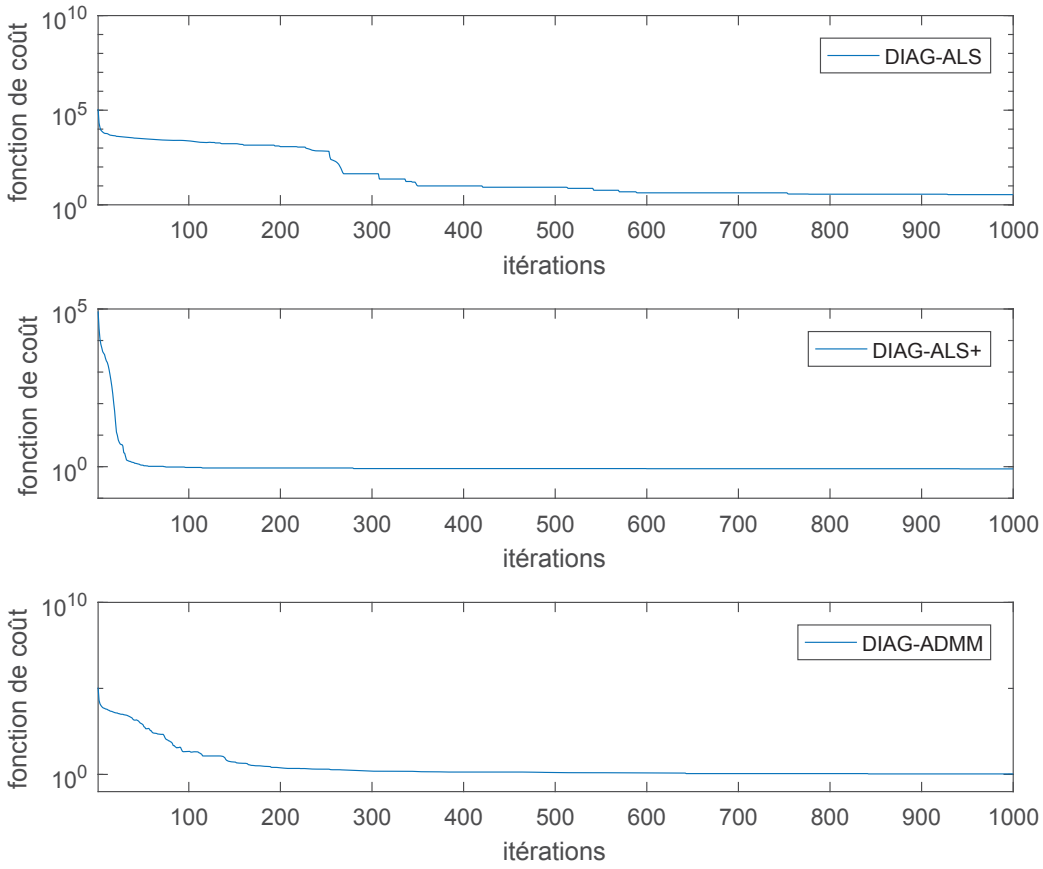

FIGURE 5.5: Évolution de la fonction de reconstruction $\Psi(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}})$ du tenseur $\mathcal{M}$ en fonction des itérations avec $\delta = 0$ et à 60 dB.

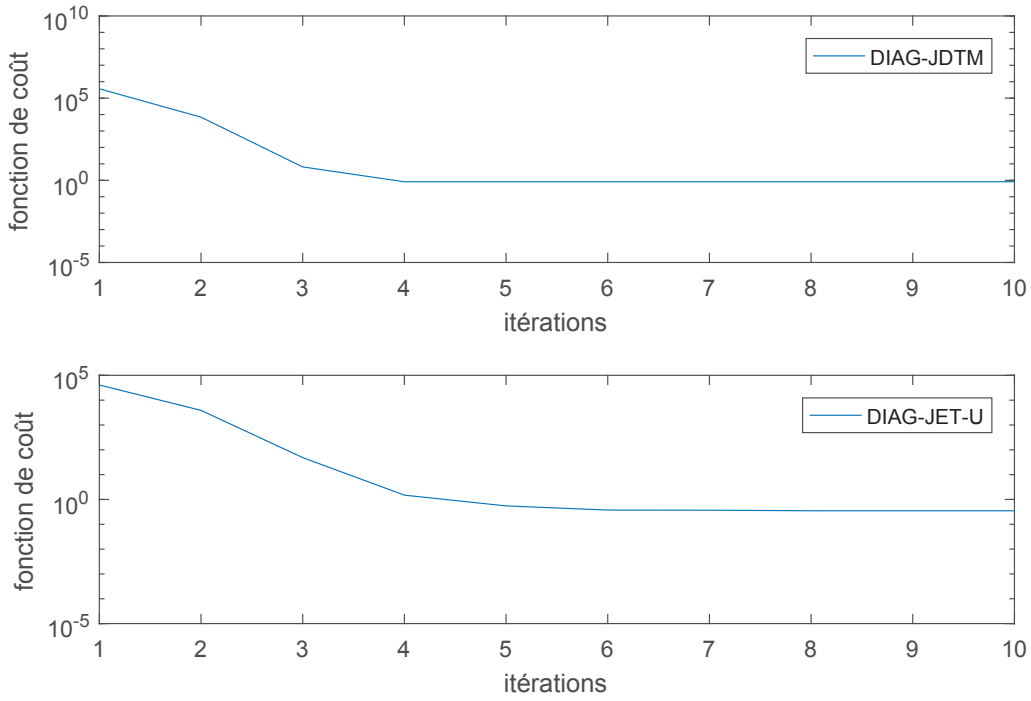

FIGURE 5.6: Évolution respective des critères C_{inverse} (3.14) et C_{triang} (3.63) pour les algorithmes JDTM et JET-U avec $\delta = 0$ et à 60 dB.

5.1. L'algorithme le moins coûteux est ici DIAG-JET-U suivi de DIAG-JDTM puis de PROCO-ALS. DIAG-JSJD et SSD-JDJS2 sont respectivement les troisième et quatrième algorithmes les moins coûteux. DIAG-ALS+ est moins coûteux que DIAG-ADMM. Enfin les algorithmes les plus coûteux sont respectivement DIAG-ALS et ADMoM qui ne bénéficient pas d'étape de réduction de dimensions.

Algorithmes	ADMoM	PROCO-ALS	ALS	ALS+	ADMM	JDTM	JET-U	JDJS	JDJS2
Coût calcul	$1,6.10^8$	$5,1.10^6$	$1,9.10^8$	$3,2.10^7$	$9,8.10^7$	$3,4.10^6$	$6,6.10^5$	$8,2.10^6$	$8,9.10^6$

TABLE 5.1: Tableau des coûts calcul totaux des différents algorithmes itératifs étudiés à 60 dB et pour $\delta = 0$.

Cette première étude nous permet de tirer les conclusions suivantes :

- Les méthodes alternées (DIAG-ALS, DIAG-ALS+ et DIAG-ADMM) ne sont pas les plus performantes pour résoudre le problème de DCS dans le cadre d'une décomposition CP où les colonnes des matrices facteurs ont un faible degré de corrélation. En effet, les algorithmes alternés proposés sont généralement moins précis pour l'estimation des matrices facteurs et plus coûteux que ceux avec lesquels ils sont comparés. Cependant, la prise en compte des contraintes de non-négativité offertes par le problème de décomposition CP non-négative permet d'améliorer la précision et de diminuer le coût calcul de ces derniers.

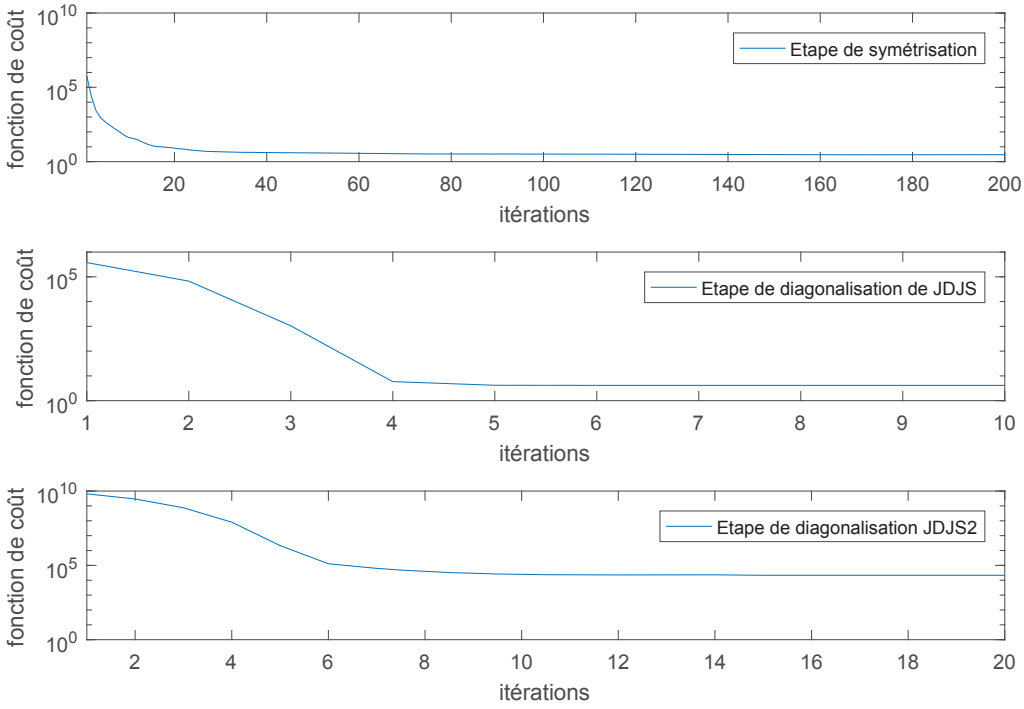

FIGURE 5.7: Évolution respective des critères C_{sym} (5.7) et C_{inverse} (3.14) pour l'étape de symétrisation et de diagonalisation de l'algorithme JDJS avec $\delta = 0$ et à 60 dB.

- Cette étude ne permet pas de montrer l'intérêt pratique de DIAG-JDJS, ce dernier a un coût de calcul assez faible mais ne procure pas les meilleurs résultats en termes de précision.

- SSD-JDJS2 fournit ici de bons résultats et plus particulièrement en termes de précision. Le niveau de performances atteint par SSD-JDJS2 comparé à DIAG-JDJS montre ici l'intérêt de forcer les matrices de l'ensemble à être parfaitement symétriques et définies positives (équation (5.24)).

5.3.2 Scénario $\delta = 0,85$

La valeur médiane de r_F pour ce scénario est affichée sur la figure 5.8. Nous pouvons remarquer qu'ici ADMoM et PROCO-ALS fournissent une mauvaise estimation des matrices facteurs pour toutes les valeurs du RSB. DIAG-JDJS a une erreur médiane inférieure à celle de DIAG-JET-U et DIAG-JDTM pour tous les RSB. SSD-JDJS2 fournit de meilleurs résultats que DIAG-JDJS au-dessus de 40 dB, puis à partir de cette valeur son erreur médiane devient plus importante que celle de DIAG-JDTM et DIAG-JET-U. DIAG-ALS, DIAG-ALS+ et DIAG-ADMM fournissent de moins bons résultats que DIAG-JET-U et DIAG-JDTM au-dessus de 60 dB. Inversement, sous cette valeur, les trois méthodes alternées proposées estiment le mieux les matrices facteurs en médiane. DIAG-ALS+ et DIAG-ADMM surpassent DIAG-ALS sous 40 dB.

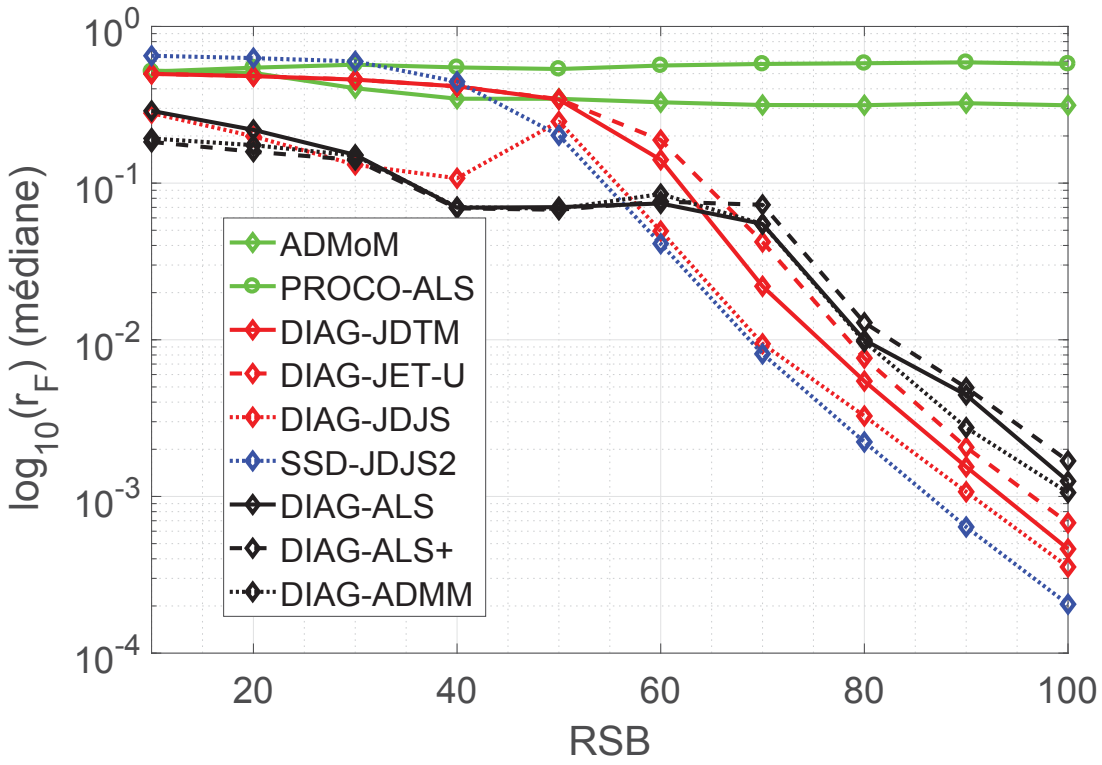

FIGURE 5.8: r_F médian en fonction du RSB ($\delta = 0,85$).

La valeur moyenne de r_F en fonction du RSB est tracée sur la figure 5.9. D'une manière générale, les algorithmes ont le même comportement que pour la valeur médiane de r_F.

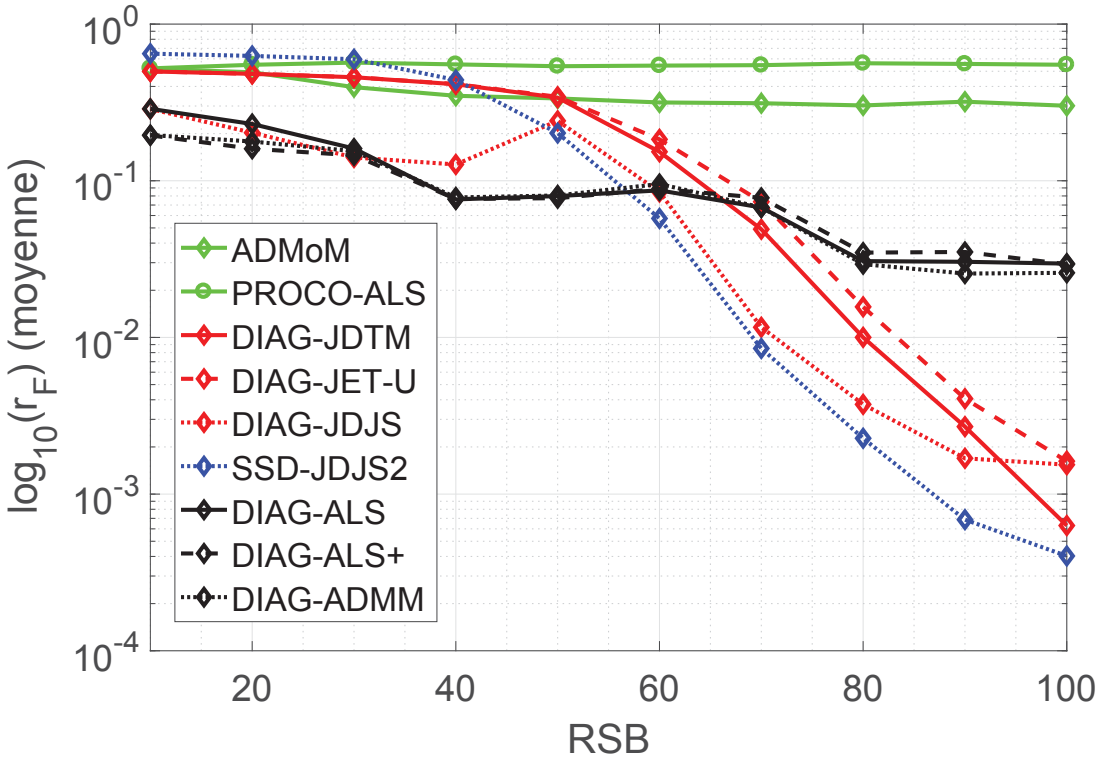

FIGURE 5.9: r_F moyen en fonction du RSB ($\delta = 0,85$).

L'écart type de r_F est tracée sur la figure 5.10. SSD-JDJS2 est l'algorithme ayant le plus petit écart type au-dessus de 50 dB. DIAG-JDJS atteint ici le même niveau de performances que ce dernier pour 70 et 80 dB, puis son écart type devient supérieur à celui de DIAG-JET-U et DIAG-JDTM sous 70 dB. Il est notable que bien que DIAG-JET-U et DIAG-JDTM aient un faible écart type sous 70 dB en comparaison avec la plupart des autres algorithmes, ces derniers n'ont pas une erreur moyenne suffisamment faible pour que cette valeur de l'écart type soit significative.

Ainsi les algorithmes alternés proposés procurent la meilleure précision sous 60 dB. DIAG-ALS+ et DIAG-ADMM sont les algorithmes les plus précis sous 40 dB. SSD-JDJS2 est quant à lui l'algorithme le plus précis à partir de 60 dB.

L'évolution de l'erreur de reconstruction du tenseur $\mathcal{T}$ est tracée sur la figure 5.11 pour ADMoM et PROCO-ALS à 60 dB. Nous pouvons remarquer que cette erreur n'évolue plus de manière significative à partir de 20 itérations pour ADMoM et d'environ 100 itérations pour PROCO-ALS. L'estimation des matrices facteurs étant peu satisfaisante pour ces deux méthodes, nous pouvons supposer que ces méthodes convergent rapidement vers un minimum local.

L'évolution de l'erreur de reconstruction du tenseur $\mathcal{M}$ en fonction du nombre d'itérations (à 60 dB) est tracée sur la figure 5.11 pour les algorithmes DIAG-ALS, DIAG-ALS+ et pour DIAG-ADMM. Nous pouvons remarquer que cette erreur n'évolue quasiment plus

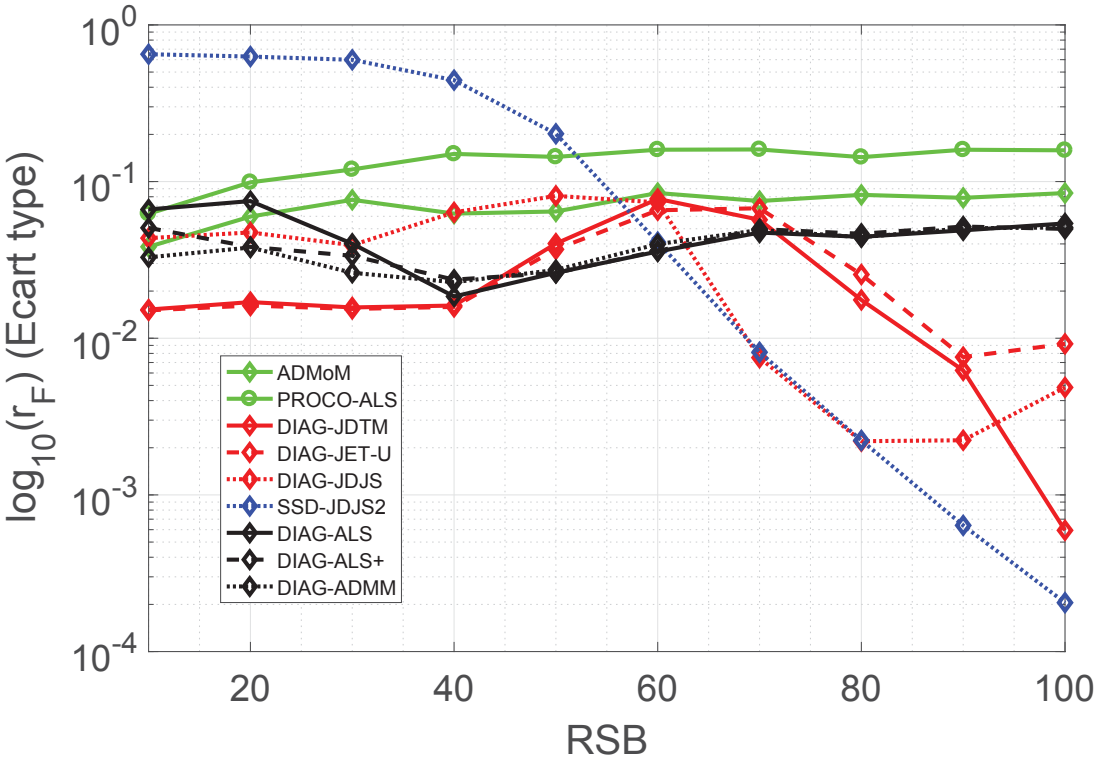

FIGURE 5.10: écart type de r_F en fonction du RSB ($\delta = 0,85$).

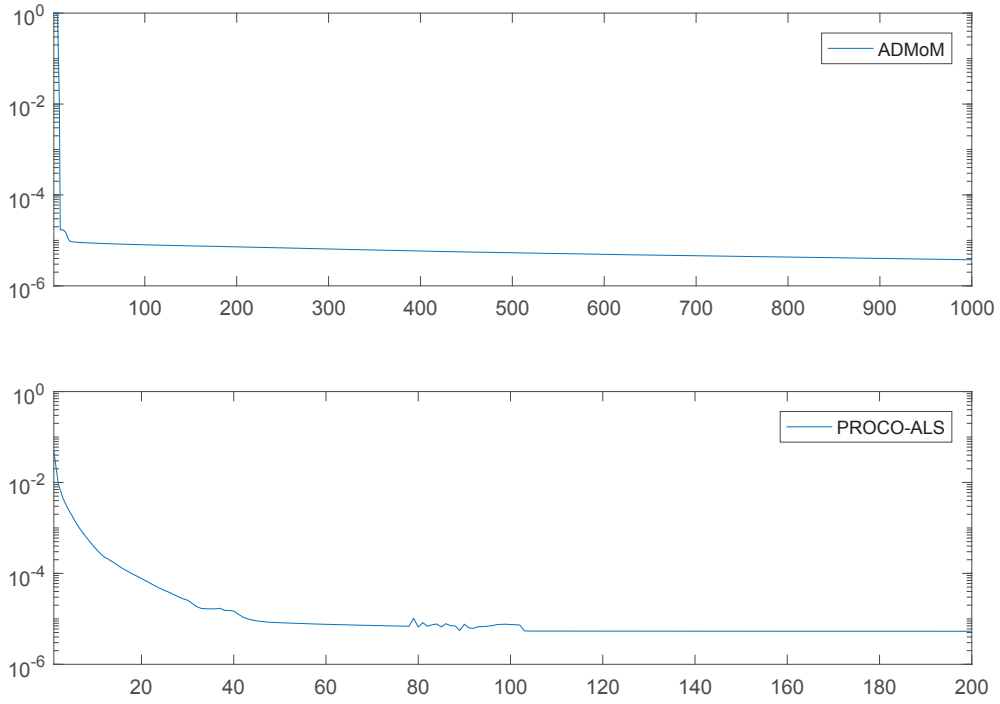

FIGURE 5.11: Évolution de la médiane de l'erreur de reconstruction $\Psi(\widehat{\mathbf{F}}^{(1)}, \widehat{\mathbf{F}}^{(2)}, \widehat{\mathbf{F}}^{(3)})$ du tenseur $\mathcal{T}$ en fonction des itérations avec $\delta = 0,85$ et à 60 dB.

pour les trois méthodes de DCS alternées à partir de 100 itérations.

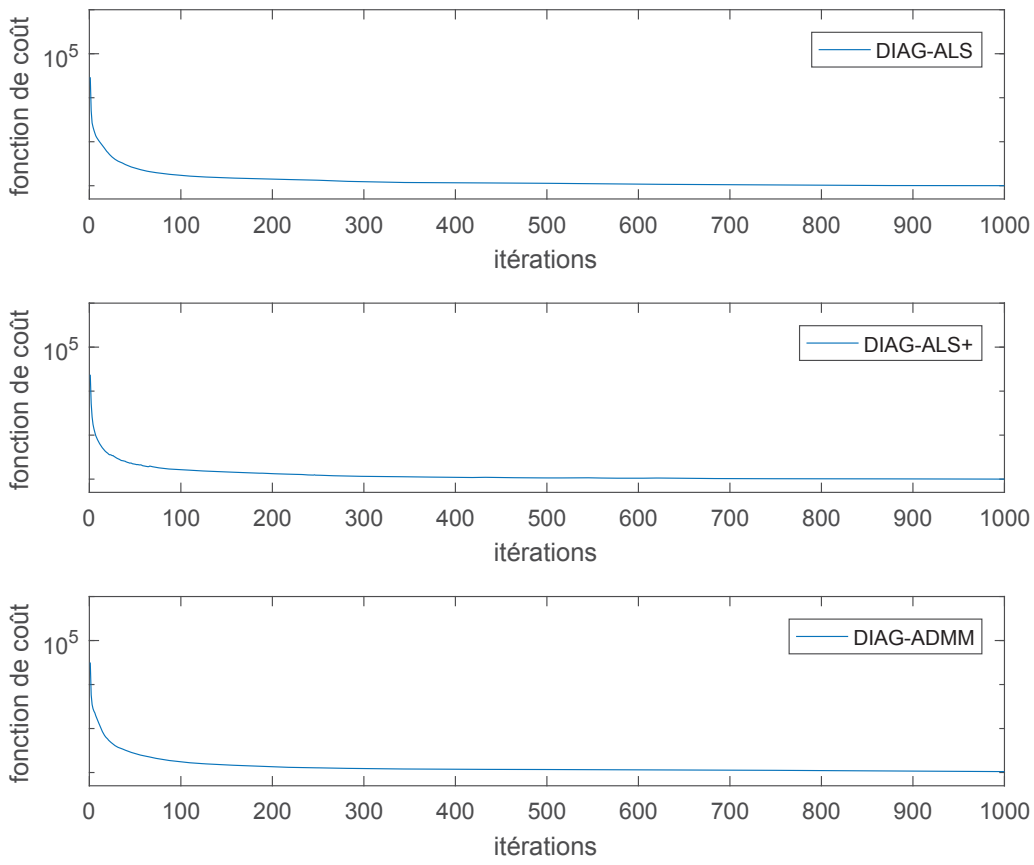

FIGURE 5.12: Évolution de la fonction de reconstruction du tenseur $\mathcal{M}$ $\Psi(\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^{(2)}, \widehat{\mathbf{C}})$ en fonction des itérations avec $\delta = 0,85$ et à 60 dB.

L'évolution du critère (3.14) pour l'algorithme JDTM et l'évolution du critère (3.63) pour l'algorithme JET-U au cours des itérations (à 60 dB) sont tracées sur la figure 5.6. Pour JDTM, le critère n'évolue plus à partir de 3 itérations. Pour JET-U le critère ne semble plus évoluer à partir de 9 itérations.

L'évolution du critère (5.7) pour l'étape de symétrisation de JDJS et de JDJS2 ainsi que l'évolution du critère (3.14) pour l'étape de diagonalisation de JDJS et de JDJS2 au cours des itérations (à 60 dB) sont tracées sur la figure 5.14. Le critère de symétrisation ne diminue plus à partir de 10 itérations. Le critère de diagonalisation n'évolue plus à partir de 2 itérations pour JDJS et à partir de 25 itérations pour JDJS2.

Le coût de calcul total de chacun des algorithmes comparés est présenté dans le tableau 5.2. L'algorithme le moins coûteux est DIAG-JET-U, le second est DIAG-JDTM, le troisième est DIAG-JDJS, le quatrième est PROCO-ALS, le cinquième est ADMoM et le sixième est SSD-JDJS2. Enfin les trois algorithmes les plus coûteux sont DIAG-ALS, DIAG-ALS+ et DIAG-ADMM.

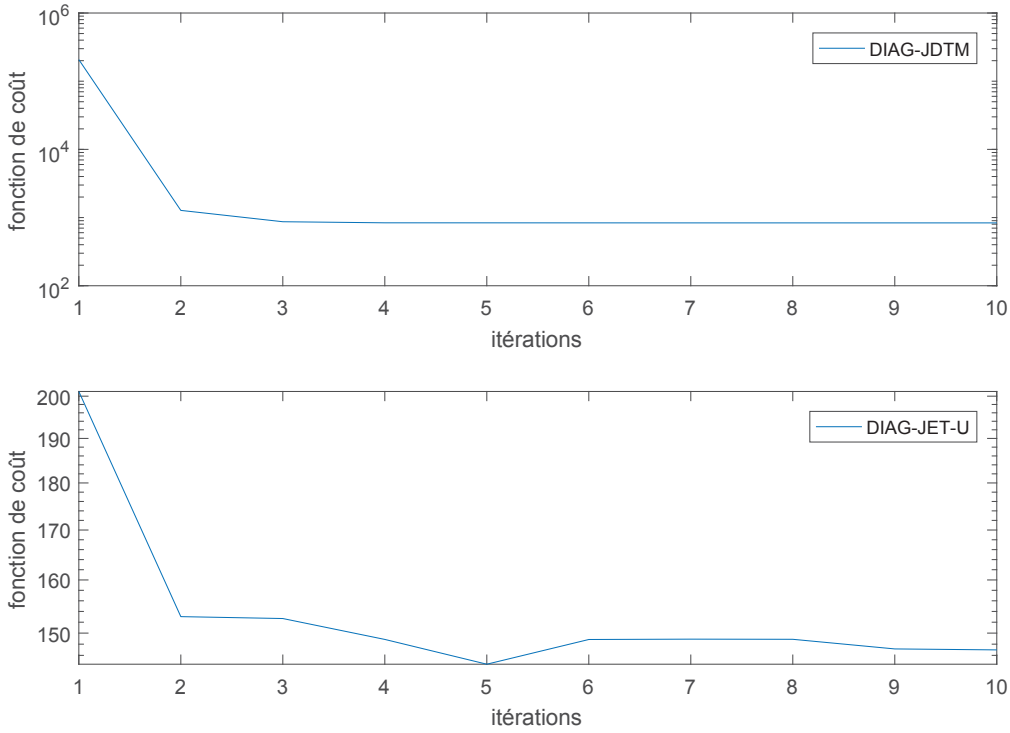

FIGURE 5.13: Évolution respective des critères C_{inverse} (3.14) et C_{triang} (3.63) pour les algorithmes JDTM et JET-U avec $\delta = 0,85$ et à 60 dB.

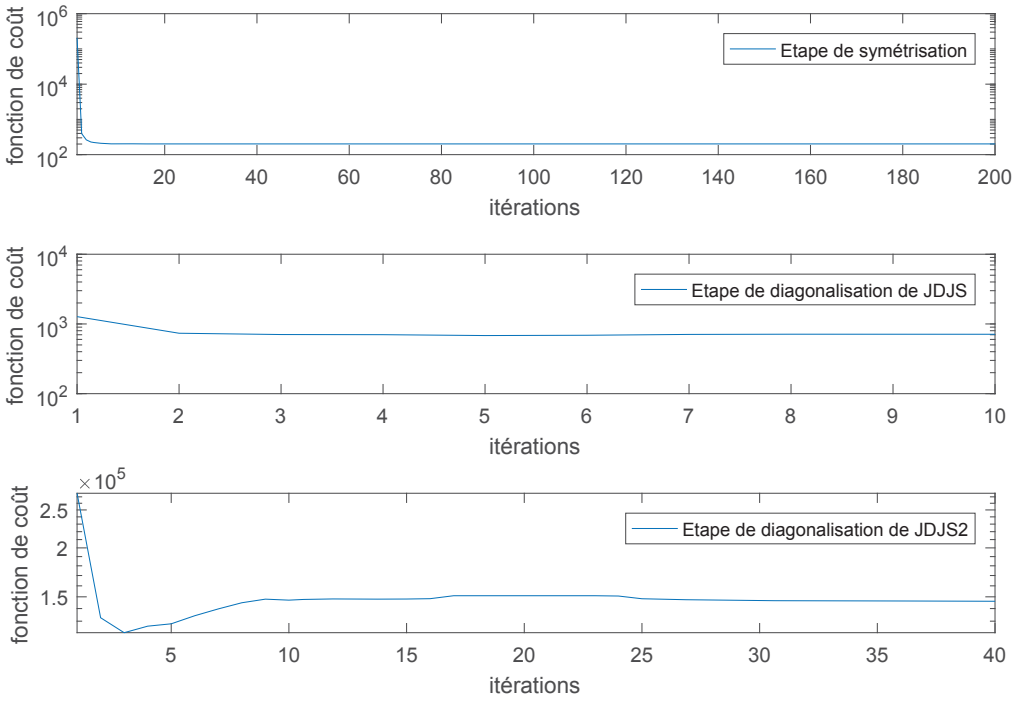

FIGURE 5.14: Évolution respective des critères C_{sym} (5.7) et C_{inverse} (3.14) pour l'étape de symétrisation et de diagonalisation de l'algorithme JDJS avec $\delta = 0,85$ et à 60 dB.

Algorithmes	ADMoM	PROCO-ALS	ALS	ALS+	ADMM	JDTM	JET-U	JDJS	JDJS2
Coût calcul	$8,1.10^6$	$5,1.10^6$	$3,2.10^7$	$3,2.10^7$	$3,2.10^7$	$2,6.10^6$	$9,8.10^5$	$3,0.10^6$	$1,3.10^7$

TABLE 5.2: Tableau des coûts calcul totaux des différents algorithmes itératifs étudiés à 60 dB et pour $\delta = 0,85$.

Ainsi, nous pouvons observer que les méthodes alternées effectuant l'étape de DCS dans l'algorithme DIAG ont un réel intérêt lorsque les colonnes des matrices facteurs du problème de décomposition CP ont toutes un haut degré de corrélation. En effet, elles permettent d'obtenir des résultats plus précis que les autres approches lorsque le niveau de bruit est élevé. La prise en compte des contraintes de non-négativité permet de rendre ce type de méthode plus précise lorsque le RSB est très faible. Cependant, ces algorithmes sont les plus coûteux des algorithmes étudiés.

DIAG-JDJS procure de bons résultats en termes de précision et de coût calcul dans ce scénario en comparaison avec les algorithmes de la littérature.

Enfin SSD-JDJS2 est quant à lui le plus précis des algorithmes étudiés lorsque le RSB est élevé, il est cependant peu précis pour les faibles RSB. Le coût de calcul de cet algorithme est plus important que celui des algorithmes basés sur des mises à jour multiplicatives (DIAG-JET-U et DIAG-JDTM) mais bien plus faible que celui des méthodes alternées proposées. Une fois encore les résultats fournis par SSD-JDJS2 montrent l'intérêt de forcer la contrainte de positivité.

5.4 Bilan du chapitre

Dans ce chapitre, nous avons proposé deux types de méthodes pour résoudre le problème de DCS sous contraintes de non-négativité. Ces méthodes sont particulièrement intéressantes lorsque les colonnes des matrices facteurs du problème de décomposition CP ont un haut degré de corrélation.

Dans ce cas de figure, les méthodes basées sur la mise à jour alternée des matrices à estimer donnent des résultats intéressants lorsque le RSB est très faible. D'une manière générale, nos résultats montrent que le niveau de performances est amélioré par la prise en compte de l'ensemble des contraintes de non-négativité. En effet, DIAG-ALS+ et DIAG-ADMM sont plus précis que DIAG-ALS lorsque les colonnes des matrices facteurs ont un degré de corrélation nul et plus robustes au bruit que DIAG-ALS dans le cas de facteurs avec un haut degré de corrélation.

Les deux méthodes basées sur la mise à jour multiplicative de la matrice diagonalisante ne prennent en compte que la contrainte de positivité sur les valeurs propres de l'ensemble de matrices à diagonaliser. L'une des méthodes proposées (DIAG-JDJS) ne prend pas forcément en compte la contrainte lorsque le niveau de bruit ne permet pas de suffisamment symétriser l'ensemble de matrices. L'autre méthode (SSD-JDJS2) procure de très bons résultats de manière générale et plus particulièrement lorsque les colonnes des matrices facteurs du problème de décomposition CP ont un haut degré de corrélation.

Ainsi, dans le cas où les colonnes des matrices facteurs du problème de décomposition CP ont un haut degré de corrélation, les algorithmes alternés proposés sont plus adaptés lorsque le RSB est faible alors que les algorithmes basés sur une étape de symétrisation sont plus adaptés pour des RSB moyens et élevés.

Chapitre 6

Applications en séparation de sources

Dans ce chapitre, nous allons mettre en évidence l'intérêt des méthodes que nous avons développées à travers deux applications classiques de décomposition CP.

La première application concerne un dispositif de télécommunications numériques et des signaux à valeurs complexes. Nous comparerons ainsi dans cette application les algorithmes SJDTE et JAPAM-5 présentés au chapitre 4 avec des algorithmes de la littérature.

La seconde application est la spectroscopie de fluorescence. Cette technique permet de connaitre les espèces chimiques présentes dans une solution suite à une excitation lumineuse. Lorsque les espèces chimiques sont faiblement concentrées, les signaux physiques mesurés peuvent être modélisés par une décomposition CP non-négative. Cela nous donnera donc l'occasion de tester les algorithmes présentés dans le chapitre 5.

6.1 Application à la séparation de signaux de télécommunications numériques

Les tenseurs à valeurs complexes et les décompositions CP sont fréquemment utilisés pour les problèmes de séparations de sources en télécommunications numériques [12–17]. En particulier, dans [17], Sidiropoulos *et al* ont montré comment un système MIMO (Multiple Inputs Multiple Outputs) de DS-CDMA (pour *Direct sequence Code Division Multiple Access*) peut être modélisé par le biais d'une décomposition CP. Un récepteur spécifique a alors été proposé dans le but d'estimer les signaux sources de manière déterministe. En effet, en l'absence d'interférence inter-symbole, l'enveloppe complexe des signaux sources apparait comme les colonnes d'une des matrices facteurs de la décomposition CP. Depuis ces premiers travaux, plusieurs approches déterministes faisant intervenir des tenseurs ont été appliquées avec succès aux dispositifs de DS-CDMA [138–141].

6.1.1 Modélisation du dispositif DS-CDMA à l'aide de la décomposition CP

Les dispositifs DS-CDMA sont des systèmes de codage de transmissions basés sur une technique d'étalement de spectre par multiplication. Un des intérêts de ces dispositifs est qu'ils permettent le multiplexage des signaux. Ils permettent également de rendre les

signaux confidentiels en les codant. Enfin, ils procurent une bonne résistance aux interférences et ont une faible consommation d'énergie.

Le principe est le suivant : R utilisateurs transmettent en même temps des signaux sources $\mathbf{s}_r$ constitués de P symboles à une période T_s. Nous représentons ces signaux sur les colonnes de la matrice $\mathbf{S} \in \mathbb{C}^{P\times R}$. Avant émission, chaque signal $\mathbf{s}_r$ est modulé à l'aide d'un second signal numérique $\mathbf{c}_r \in \mathbb{C}^Q$ appelé code et qui contient une séquence de Q valeurs émises à une période T_c telle que $T_s = \omega T_c$ où ω est un entier supérieur à 1. Il est important de tenir compte de la différence de débit entre les signaux sources et codes pour effectuer la multiplication et donc sur-échantillonner le signal d'entré d'un facteur ω, ce qui revient à étaler son spectre. L'opération de modulation entre les signaux $\mathbf{s}_r$ et $\mathbf{c}_r$ s'écrit donc :

$$\mathbf{x}_r = \mathbf{s}_r \otimes \mathbf{c}_r. \tag{6.1}$$

Les signaux générés par l'antenne émettrice peuvent être alors rangés dans une matrice $\mathbf{X} \in \mathbb{C}^{PQ\times R}$ définie comme :

$$\mathbf{X} = \mathbf{S} \odot \mathbf{C} \tag{6.2}$$

où $\mathbf{C} \in \mathbb{C}^{Q\times R}$ est la matrice contenant les codes $\mathbf{c}_r$ sur ses colonnes. A la réception, les signaux reçus peuvent être modélisés (en négligeant les interférences inter-symbole et autres phénomènes perturbateurs) comme une combinaison linéaire des signaux émis, les paramètres de la combinaison modélisent les gains des canaux de transmission. Les gains de transmission (n, r) sont ici représentés par une matrice $\mathbf{H} \in \mathbb{C}^{N\times R}$. L'ensemble des signaux reçus peuvent alors être rangés dans une matrice $\mathbf{T}_{(1)}$ vérifiant :

$$\mathbf{T}_{(1)} = \mathbf{H}(\mathbf{S} \odot \mathbf{C})^T. \tag{6.3}$$

L'équation (6.3) est ainsi la décomposition CP de rang R d'un tenseur d'ordre 3 dont $\mathbf{T}_{(1)}$ est la matrice de dépliement dans le premier mode. Le dispositif DS-CDMA est résumé dans le schéma 6.1 de manière simplifiée. En pratique les signaux reçus sont perturbés par un bruit. Les signaux émis, le code et le gain de chaque canal peuvent alors être estimés en décomposant le tenseur suivant :

$$\mathcal{T} = \mathcal{I}_R \times_1 \mathbf{H} \times_2 \mathbf{C} \times_3 \mathbf{S} + \mathcal{E} \tag{6.4}$$

où $\mathcal{E} \in \mathbb{C}^{N\times Q\times P}$ représente le bruit.

Ainsi la décomposition CP du tenseur $\mathcal{T}$ fournit une estimation des signaux sources, des codes utilisés et des gains de transmissions (à une permutation et à un facteur d'échelle près).

6.1.2 Simulations numériques

Notre but est d'évaluer les performances de nos méthodes de DCS pour effectuer la décomposition CP de ce type de tenseurs. Ainsi nous comparons ici les performances des algorithmes suivants : ALS, ALS+ELS, DIAG-JDTM, DIAG-JET-U, DIAG-JAPAM-5 et DIAG-SJDTE. Les quatre derniers algorithmes sont quatre versions différentes de l'algorithme DIAG [22] utilisant respectivement les algorithmes JDTM, JET-U, JAPAM-5 et SJDTE pour l'étape de DCS.

L'intérêt des méthodes tensorielles étant notamment de pouvoir traiter des cas sous-déterminés, nous simulons donc un système de transmission DS-CDMA avec $R = 7$, $P =$

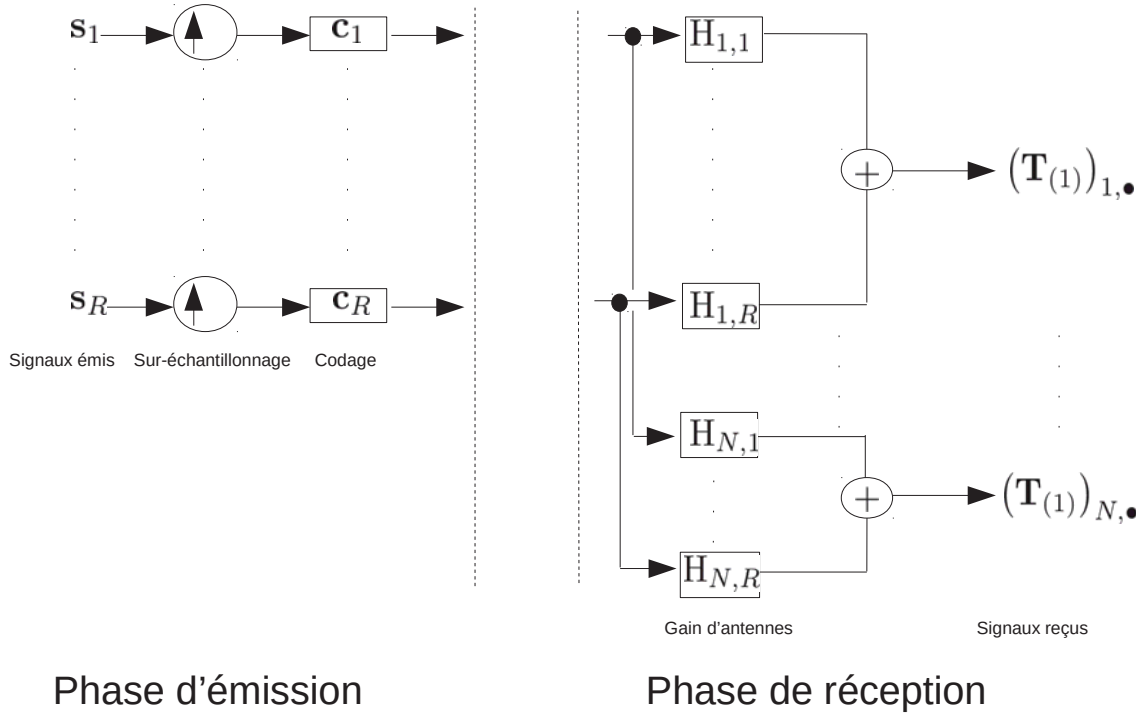

FIGURE 6.1: Schéma simplifié du dispositif DS-CDMA.

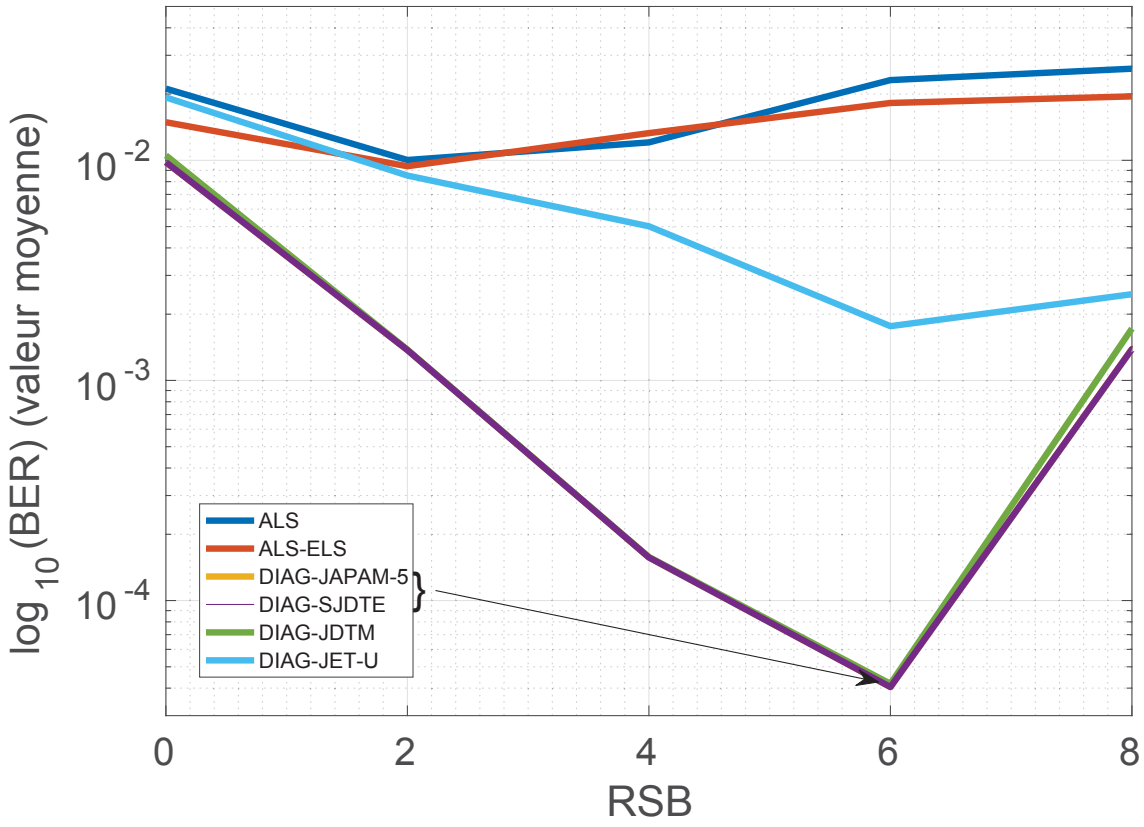

FIGURE 6.2: Évolution de la valeur moyenne du BER en fonction du RSB.

512, $Q = 16$ et $N = 5$ de sorte que nous ayons moins d'antennes réceptrices (5) que d'antennes émettrices (7). Les colonnes de la matrice $\mathbf{S}$ sont construites à partir de la modulation QPSK d'un signal binaire aléatoire de taille $2P$, la matrice $\mathbf{C}$ est générée à partir d'un code de Hadamard et la matrice $\mathbf{H}$ est construite selon une loi normale centrée et réduite.

Nous faisons varier le RSB de 0 dB à 8 dB par pas de 2 dB. Pour chaque RSB, nous construisons 200 tenseurs résultants du dispositif DS-CDMA précédemment décrit et nous analysons les performances des différents algorithmes étudiés en moyennant les résultats obtenus sur ces différents tenseurs.

Notre premier critère de performances est le Bit Error Rate (BER) ou taux d'erreur binaire que nous calculons après démodulation des colonnes de la matrice source estimée pour chaque algorithme. Notre second critère de performances est le critère r_F présenté dans la section simulations numériques du chapitre 5 à l'équation (5.60). Nous rappelons que ce critère permet d'évaluer l'erreur d'estimation des trois matrices facteurs. Enfin, nous étudierons le nombre d'itérations dont les différents algorithmes de DCS ont besoin pour atteindre le critère d'arrêt $|S(\mathbf{B}_{it+1}) - S(\mathbf{B}_{it})| < 10^{-6}$. Où $S(\mathbf{B})$ est la fonction définie dans (4.72).

L'évolution de la valeur moyenne du BER est disponible sur la figure 6.2. Nous pouvons observer que DIAG-JDTM, DIAG-SJDTE et DIAG-JAPAM-5 surpassent largement les

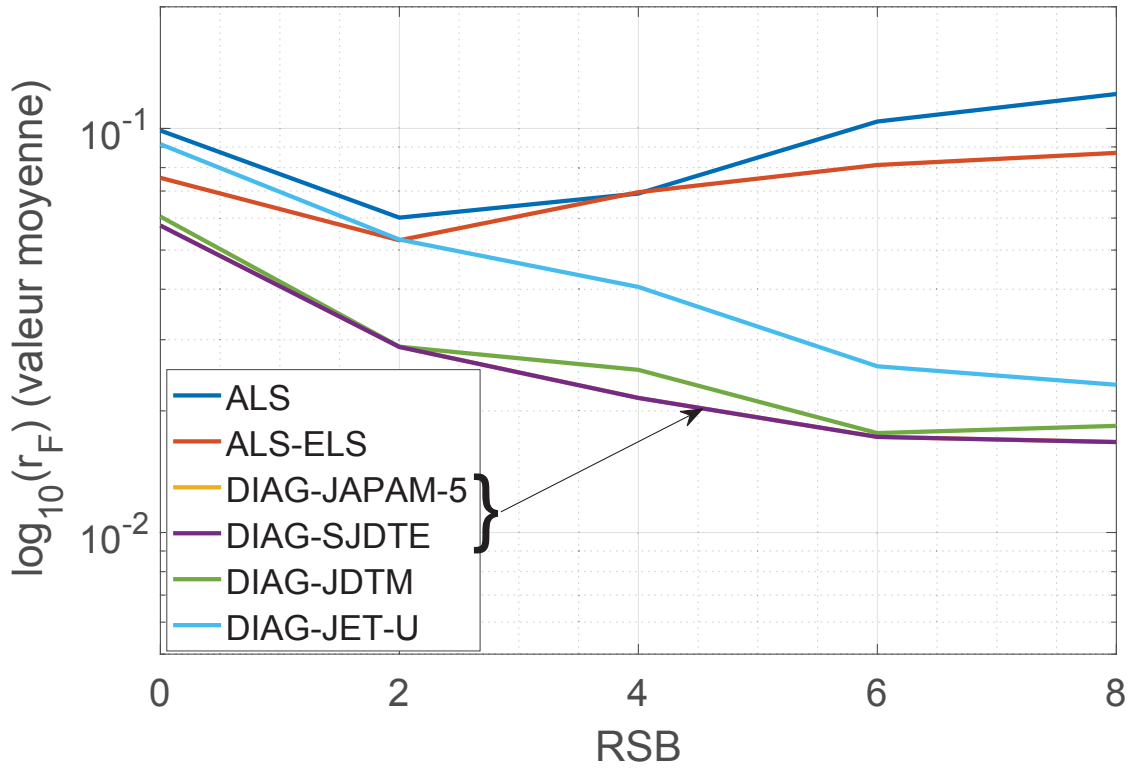

FIGURE 6.3: Évolution de la valeur moyenne du r_F en fonction du RSB.

autres algorithmes étudiés. Par ailleurs, DIAG-SJDTE et DIAG-JAPAM-5 atteignent le même niveau de performances et sont légèrement plus performants que DIAG-JDTM pour 8 dB.

La valeur moyenne de r_F est affichée sur la figure 6.3. Comme pour le BER, DIAG-SJDTE et DIAG-JAPAM-5 sont les algorithmes les plus performants, suivis de près par DIAG-JDTM.

Nous pouvons observer le nombre d'itérations des algorithmes de DCS sur la figure 6.4. JET-U a besoin d'environ dix fois moins d'itérations que JDTM pour atteindre le critère d'arrêt prédéfini. JAPAM-5 et SJDTE ont besoin d'environs deux fois moins d'itérations que JET-U pour pour atteindre le critère d'arrêt.

Ainsi dans cette étude, nous démontrons que les algorithmes SJDTE et JAPAM-5 peuvent être utilisés avec intérêt pour la séparation de signaux DS-CDMA via l'algorithme DIAG. Nos deux algorithmes sont légèrement plus précis que les algorithmes les plus précis de la littérature, mais ils se démarquent surtout sur le nombre d'itérations dont ils ont besoin pour atteindre le critère d'arrêt choisi.

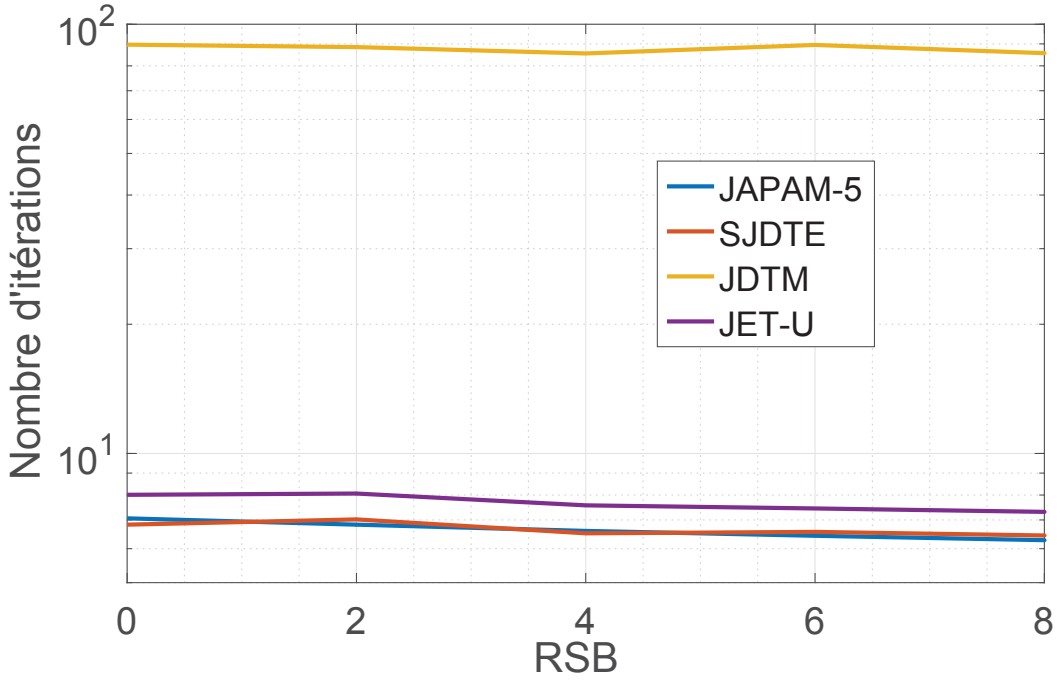

FIGURE 6.4: Évolution de la valeur moyenne du nombre d'itérations en fonction du RSB pour l'étape de DCS.

6.2 Application au dé-mélange de spectres de fluorescence

Couplée à des méthodes de séparation de sources, la spectroscopie de fluorescence est une technique permettant de connaitre les espèces chimiques présentes dans différentes solutions ainsi que l'évolution de leurs concentrations relatives.

Lorsqu'une molécule (ou un atome) est excitée par une source lumineuse (photons), celle-ci tend à revenir à un état d'excitation stable. Lors de l'excitation, la molécule absorbe donc des photons, puis elle en émet lors de sa relaxation après un délais très court. Ce phénomène s'appelle la fluorescence. La fluorescence est un des processus (radiatif) de relaxation. La molécule (ou l'atome) peut aussi revenir à son état fondamental sans aucune transition radiative (processus non-radiatif). Les molécules (ou atomes) ayant une relaxation radiative sont appelées fluorophores. Les fluorophores présents dans une solution peuvent alors être identifiés grâce aux spectres lumineux qui les excitent et aux spectres lumineux qu'ils émettent par fluorescence.

6.2.1 Absorption d'un photon par une molécule ou un atome

Un photon est un quanta d'énergie lumineuse (ou électromagnétique) qui est défini par son énergie $E = h\upsilon$, avec h la constante de Planck ($6,62 \times 10^{-34} J.s$) et υ la fréquence de l'onde électromagnétique définit par $\upsilon = c/\lambda$ (c la vitesse de la lumière dans le vide égale à $3 \times 10^8 m.s^{-1}$ et λ la longueur d'onde). Un atome, à un état stable S_0 (dit fondamental), voit ses électrons répartis sur ses orbitales atomiques (régions sur lesquelles il y a de

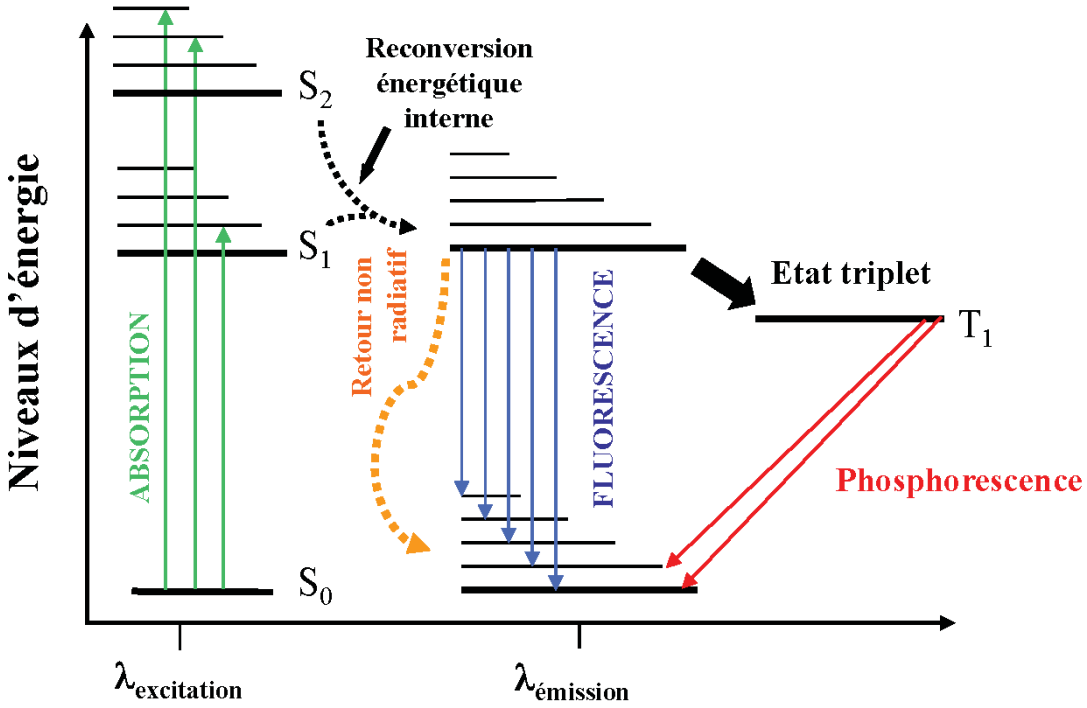

FIGURE 6.5: Diagramme de Jablonski.

fortes probabilités que se situent les électrons d'un atome) selon les lois de la mécanique quantique. Lorsque que l'atome est irradié par un photon suffisamment énergétique (voir conditions dans [21]), il peut passer à un état excité S_1, un électron de cet atome se situera donc sur une orbitale inoccupée de plus haute énergie. Les niveaux d'énergie d'un atome sont quantifiés et notés S_n. La mise en relation des électrons des atomes pour former des molécules s'appelle liaison chimique. Ces liaisons chimiques sont en fait une combinaison d'orbitales atomiques définissant une orbitale moléculaire. On peut alors généraliser les notions vues précédemment aux molécules.

Le diagramme de Jablonski (figure 6.5) tiré de [142] résume le phénomène de fluorescence. Il introduit aussi la notion de phosphorescence qui est un retour à l'état initial plus long et plus complexe que la fluorescence, nous ne nous intéresserons pas à ce phénomène ici. Ainsi la fluorescence est le retour à un état stable d'un atome ou d'une molécule par émission d'énergie lumineuse lorsque ce dernier ou cette dernière a été excité par une source lumineuse (sans passer par un état triplet).

6.2.2 La spectroscopie de fluorescence

Nous avons donc vu dans la section précédente qu'après excitation, une molécule ou un atome émet une radiation électromagnétique (lumière). La longueur d'onde de cette

radiation caractérise une liaison entre atomes ou molécules. Sur la figure 6.6, nous pou-

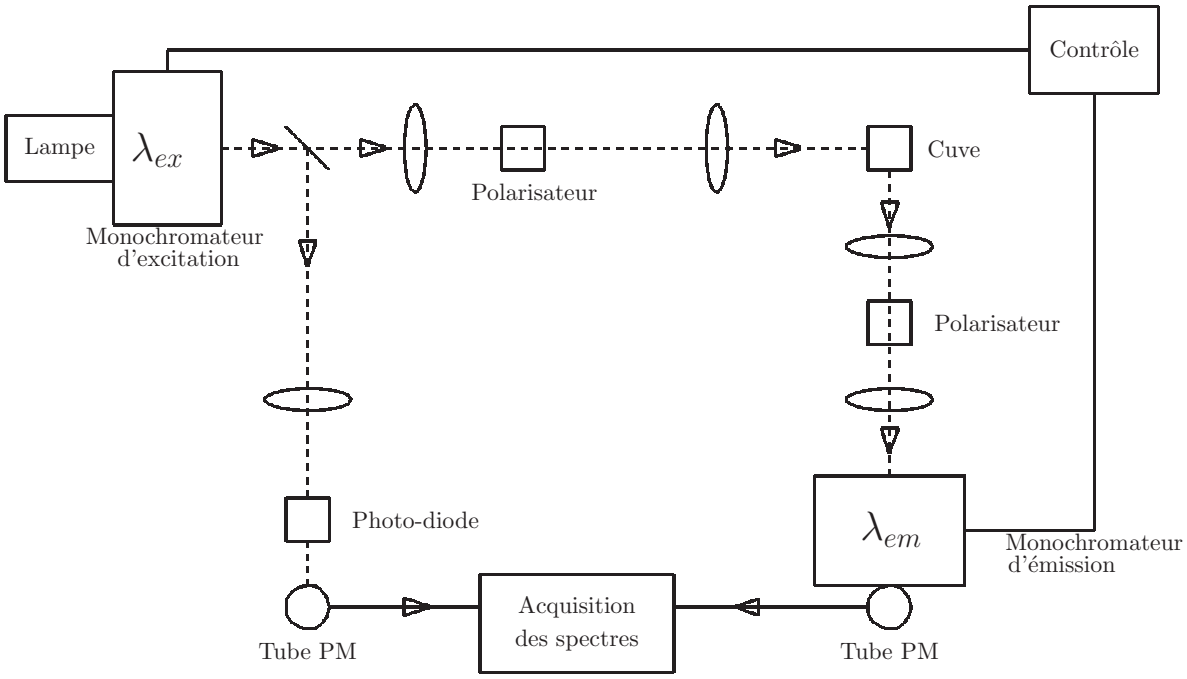

FIGURE 6.6: Schéma du spectrofluorimètre.

vons observer le principe du spectrofluorimètre. La source lumineuse est tout d'abord filtrée grâce au monochromateur d'excitation. Ce filtre permet de sélectionner les longueurs d'ondes destinées à exciter la solution à étudier. En sortie du filtre d'excitation, la lumière est polarisée et concentrée sur la solution à étudier et vient donc exciter les fluorophores présents dans celle-ci. La lumière de fluorescence émise par la solution à plusieurs longueurs d'ondes est alors à son tour polarisée et concentrée sur le monochromateur d'émission. Les spectres (intensité lumineuse mesurée en fonction de la longueur d'onde) d'émission et d'excitation sont alors acquis grâce à une photo-diode.

Grâce au monochromateur d'excitation, on peut mesurer un spectre d'émission à une longueur d'onde d'excitation donnée λ_e. Et inversement, grâce au monochromateur d'émission, on peut mesurer un spectre d'excitation à une longueur d'onde d'émission de fluorescence donnée λ_f. De plus, la fonction monochromatrice des filtres d'excitation et d'émission est primordiale car elle donne une condition indispensable pour utiliser la loi de Beer-Lambert qui ne s'applique que sur des faisceaux ne possédant qu'une seule longueur d'onde et qui permet d'obtenir l'intensité de fluorescence $I(\lambda_f, \lambda_e)$ mesurée pour un couple de longueurs d'onde fixé. Pour une solution ne comportant qu'un seul fluorophore, faiblement concentré, l'approximation linéaire de la loi de Beer-Lambert s'écrit

$$I(\lambda_f, \lambda_e) = I_0 \beta(\lambda_f) \gamma(\lambda_e) c, \tag{6.5}$$

avec I_0 une constante, β le signal proportionnel au spectre d'émission de fluorescence du fluorophore, γ le signal proportionnel à son spectre d'excitation et c sa concentration dans la solution étudiée. Le signal mesuré pour un ensemble de couples (λ_f, λ_e) est un signal bi-dimensionnel appelé Matrice d'Émission-Excitation de Fluorescence (MEEF).

Pour un mélange de R fluorophores (faiblement concentrés), on mesure une combinaison linéaire des spectres :

$$I(\lambda_f, \lambda_e) = I_0 \sum_{r=1}^{R} \beta_r(\lambda_f)\gamma_r(\lambda_e)c_r, \tag{6.6}$$

avec r l'indice du fluorophore considéré.

Enfin, on peut généraliser cette formule pour K solutions correspondant à K mélanges de R fluorophores :

$$\forall k \in [1; K]_{\mathbb{N}},\ I(\lambda_f, \lambda_e, k) = I_0 \sum_{r=1}^{R} \beta_r(\lambda_f)\gamma_r(\lambda_e)c_{k,r}, \tag{6.7}$$

Les monochromateurs permettent de parcourir la plage de longueurs d'onde d'émission et d'excitation désirée avec un pas arbitraire, ainsi le signal mesuré est donc un signal discret et tri-dimensionnel que nous modélisons à l'aide d'un tenseur $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$. Les indices $i \in [1; I]_{\mathbb{N}}$ correspondent aux longueurs d'onde d'émission λ_f sélectionnées et les indices $j \in [1; J]_{\mathbb{N}}$ correspondent aux longueurs d'onde d'excitation λ_e sélectionnées. Ainsi en prenant en compte l'équation (6.7), nous pouvons observer que l'expression du tenseur $\mathcal{T}$ est équivalente à l'expression du modèle de décomposition CP. Nous avons donc :

$$\forall (i, j, k) \in [1; I]_{\mathbb{N}} \times [1; J]_{\mathbb{N}} \times [1; K]_{\mathbb{N}},\ T_{i,j,k} = \sum_{r=1}^{R} A_{i,r}B_{j,r}C_{k,r}, \tag{6.8}$$

où la matrice $\mathbf{A} \in \mathbb{R}^{I\times R}$ contient sur ses colonnes les spectres d'émission de chaque fluorophore, la matrice $\mathbf{B} \in \mathbb{R}^{J\times R}$ contient sur ses colonnes les spectres d'excitation de chaque fluorophore et la matrice $\mathbf{C} \in \mathbb{R}^{K\times R}$ contient sur ses colonnes l'évolution relative des concentrations de chaque fluorophore dans les différentes solutions étudiées. Chaque MEEF correspond alors à une tranche frontale du tenseur. Les spectres d'émission et d'excitation, et les concentrations ayant des valeurs non-négatives, nous avons ici affaire à un problème de décomposition CP non-négative.

6.2.3 Application sur des données réelles

Les données sur lesquels nous allons effectuer notre étude ont été présentées dans [143, 144] et sont disponibles sur le site internet `http://www.models.life.ku.dk/Fluorescence`. Nous utilisons le jeu de données numéro 11 de ce site contenant 51 mélanges de quatre fluorophores : Le catéchol, l'hydroquinone, l'indole et la tyrosine. Ce jeu de données contient les MEEF de 51 mélanges, mesurées pour 136 longueurs d'onde d'émission (de 230 à 500 nanomètre (nm) tous les 2 nm) et pour 19 longueurs d'onde d'excitation (de 230 à 320 nm tous les 5 nm). Le jeu de données numéro 0 contenant les solutions pures, nous avons accès aux spectres des fluorophores que nous devons estimer. De même, les concentrations utilisées pour fabriquer les mélanges sont connues. Ce jeu de données est considéré comme compliqué car trois de ses composés (le catéchol, l'indole et la tyrosine) ont des spectres d'excitation très proches, de plus tous les composés ne sont pas forcément présents dans les 51 mélanges.

Lors de l'acquisition des MEEF, des spectres de diffusion non-linéaires (non pris en compte par le modèle de décomposition CP) apparaissent. Dans le but de les supprimer, nous utilisons la méthode proposée dans [145], puis nous appliquons un filtre passe-bas à toutes les MEEF. Pour supprimer les problèmes de non-linéarité persistant, nous retirons les trois premières longueurs d'onde d'excitation. Nous obtenons donc un tenseur de dimensions $16 \times 136 \times 51$ et de rang utile 4.

Dans cette étude, nous comparons les performances de DIAG-JDJS, SSD-JDJS2, DIAG-ALS, DIAG-ALS+ et DIAG-ADMM avec celles de ADMoM, PROCO-ALS et DIAG-JDTM. Dans PROCO-ALS, le tenseur de données est compressé de telle sorte que le tenseur obtenu ait ses dimensions toutes égales à 16. Il est important de noter que pour les algorithmes DIAG et SSD-CP, les matrices facteurs ou les produits de Khatri-Rao des matrices facteurs estimées sont projetés en prenant leur valeur absolue.

Le critère de performances utilisé est le critère r_F défini dans (5.60) après correction de l'indétermination d'échelle et de permutation.

Les algorithmes DIAG-ALS, DIAG-ALS+, DIAG-ADMM, ADMoM et PROCO-ALS étant sensibles à l'initialisation des matrices facteurs, nous les appliquons 100 fois au jeu de données à traiter avec à chaque fois une initialisation aléatoire.

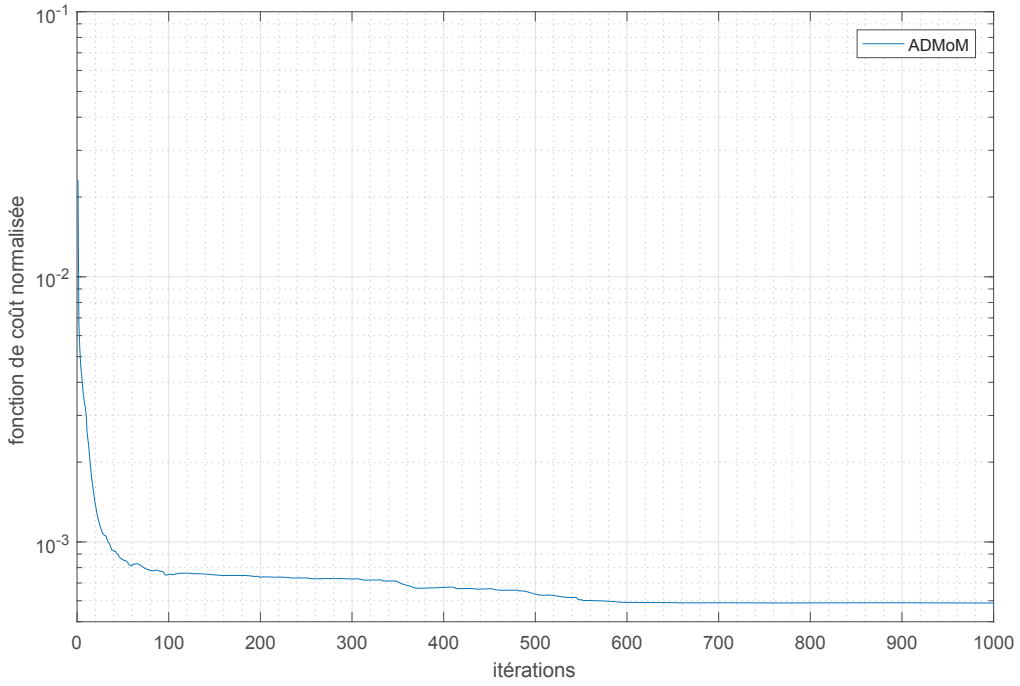

FIGURE 6.7: Évolution de la médiane de l'erreur de reconstruction normalisée du tenseur en fonction du nombre d'itérations pour ADMoM.

Les différentes fonctions de coût affichées sur les figures 6.7, 6.8, 6.9, 6.10, 6.11, 6.12 et 6.13 sont normalisées par la norme de Frobenius au carré du tenseur contenant les données à traiter.

L'évolution de la médiane de l'erreur de reconstruction normalisée du tenseur est affichée sur la figure 6.7 pour ADMoM. Nous pouvons voir que cette fonction de coût ne

décroit plus après 600 itérations.

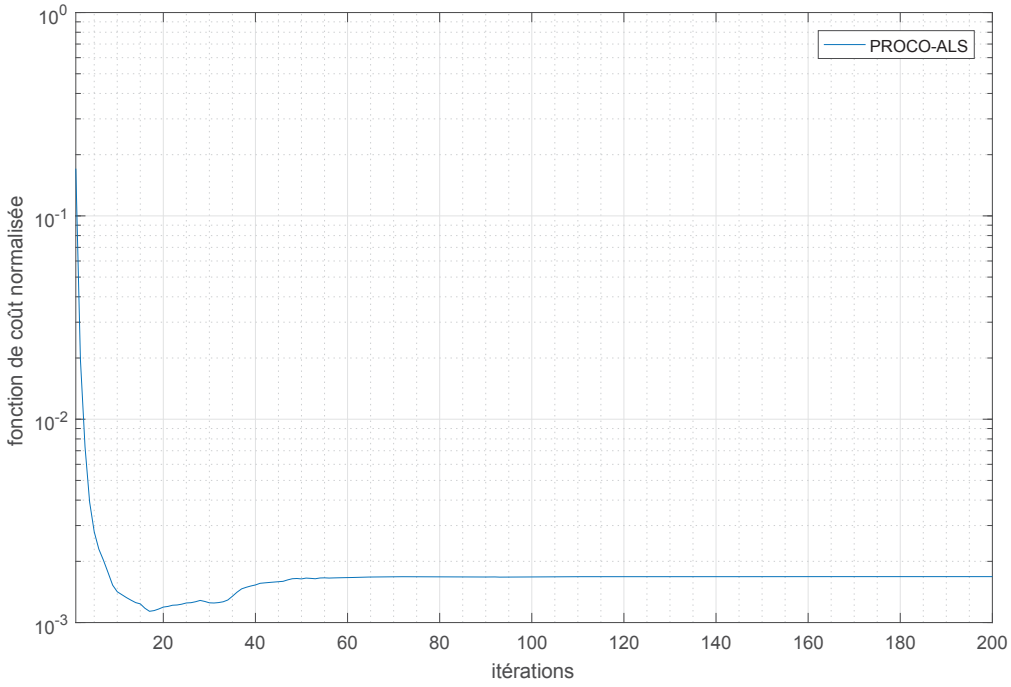

FIGURE 6.8: Évolution de la médiane de l'erreur de reconstruction normalisée du tenseur en fonction du nombre d'itérations PROCO-ALS.

L'évolution de la médiane de l'erreur de reconstruction du tenseur est affichée sur la figure 6.8 pour PROCO-ALS. Nous pouvons voir que cette fonction de coût n'évolue plus après 60 itérations.

L'évolution du critère C_{inverse} (3.14) est affichée sur la figure 6.9 pour DIAG-JDTM. Nous pouvons voir que cette fonction de coût ne décroît plus après 3 itérations.

L'évolution du critère C_{sym} (5.7) et du critère C_{inverse} (3.14) est affichée sur la figure 6.10 pour DIAG-JDJS et SSD-JDJS2. Nous pouvons voir que l'étape de symétrisation prend 3 itérations et que l'étape de diagonalisation prend 30 itérations pour JDJS2 (soit un total de 33 itérations). JDJS ne permet pas de stabiliser complètement la fonction de coût après décroissance.

L'évolution de la médiane de l'erreur de reconstruction de l'ensemble de matrice à diagonaliser est affichée sur la figure 6.11 pour DIAG-ALS. Nous pouvons voir que cette fonction de coût n'évolue plus après environ 150 itérations.

L'évolution de la médiane de l'erreur de reconstruction de l'ensemble de matrice à diagonaliser est affichée sur la figure 6.12 pour DIAG-ALS+. Nous pouvons voir que cette fonction de coût n'évolue plus après environ 750 itérations.

Enfin, l'évolution de la médiane de l'erreur de reconstruction de l'ensemble de matrice à diagonaliser est affichée sur la figure 6.13 pour DIAG-ADMM. Nous pouvons voir que cette fonction de coût n'évolue quasiment plus après environ 550 itérations.

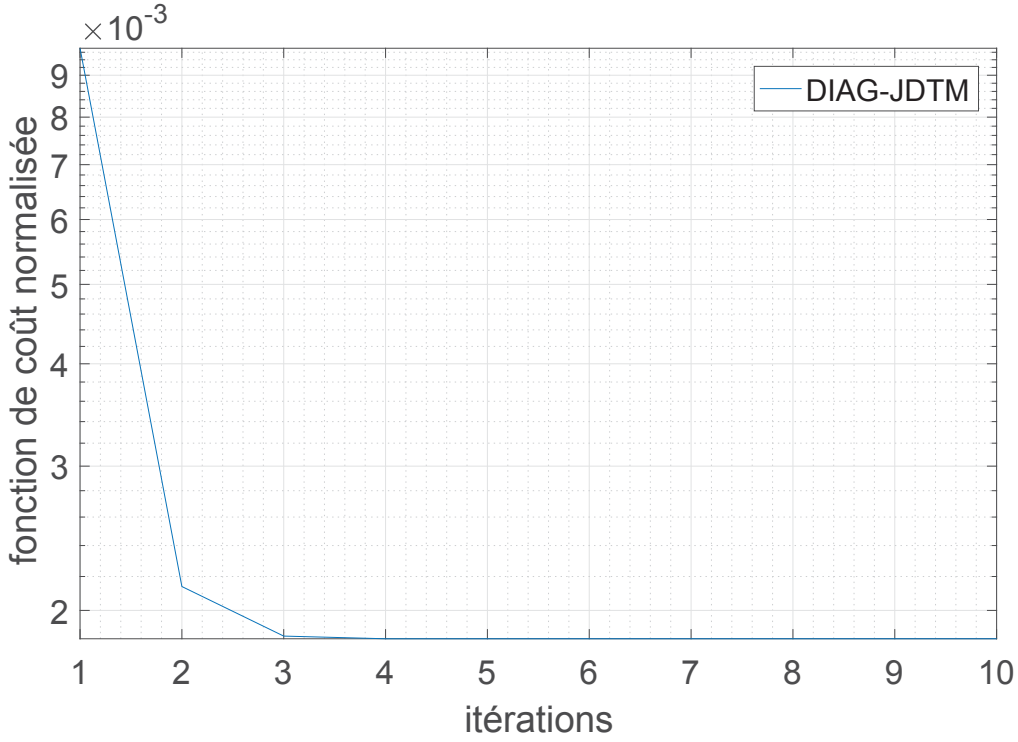

FIGURE 6.9: Évolution du critère C_{inverse} (3.14) normalisée en fonction du nombre d'itérations pour DIAG-JDTM.

Algorithmes	ADMoM	PROCO-ALS	DIAG-JDTM	DIAG-JDJS	SSD-JDJS2	DIAG-ALS	DIAG-ALS+	DIAG-ADMM
r_F moyen	0.2129	0.2318	0.1715	0.4608	0.1502	0.5367	0.5356	0.4848
r_F médian	0.1762	0.1486	0.1715	0.4608	0.1502	0.6172	0.6084	0.4866
r_F écart type	0.1299	0.1040	0	0	0	0.1327	0.1099	0.0866

TABLE 6.1: Tableau des valeurs du critères r_F.

Les valeurs moyennes et médianes du critère r_F ainsi que l'écart type sont donnés dans le tableau 6.1. Nous rappelons que les algorithmes DIAG-JDTM, DIAG-JDJS et SSD-JDJS2 étant toujours initialisés avec la matrice identité, leur moyenne est égale à leur médiane et leur écart type est égal à 0.

Le meilleur algorithme en moyenne est SSD-JDJS2 suivi par DIAG-JDTM puis par ADMoM et PROCO-ALS. DIAG-JDJS est légèrement plus précis que DIAG-ADMM en moyenne. Enfin les algorithmes estimant le moins bien les matrices facteurs en moyenne sont DIAG-ALS+ et DIAG-ALS.

Concernant la valeur médiane de r_F, l'algorithme atteignant le meilleur niveau de performances est cette fois-ci PROCO-ALS suivi par SSD-JDJS2. DIAG-ADMM et DIAG-JDJS donnent ici de moins bons résultats que ADMoM et DIAG-JDTM, mais ils fournissent de meilleurs résultats que DIAG-ALS et DIAG-ALS+. L'algorithme le moins précis est toujours DIAG-ALS.

En observant l'écart-type du critère r_F, nous pouvons voir que DIAG-ADMM est moins sensible à l'initialisation par rapport aux autres algorithmes étudiés. PROCO-ALS et DIAG-ALS+ sont aussi peu sensibles à l'initialisation. Au contraire, ADMoM et plus particulièrement DIAG-ALS sont assez sensibles à l'initialisation. Il est important de noter

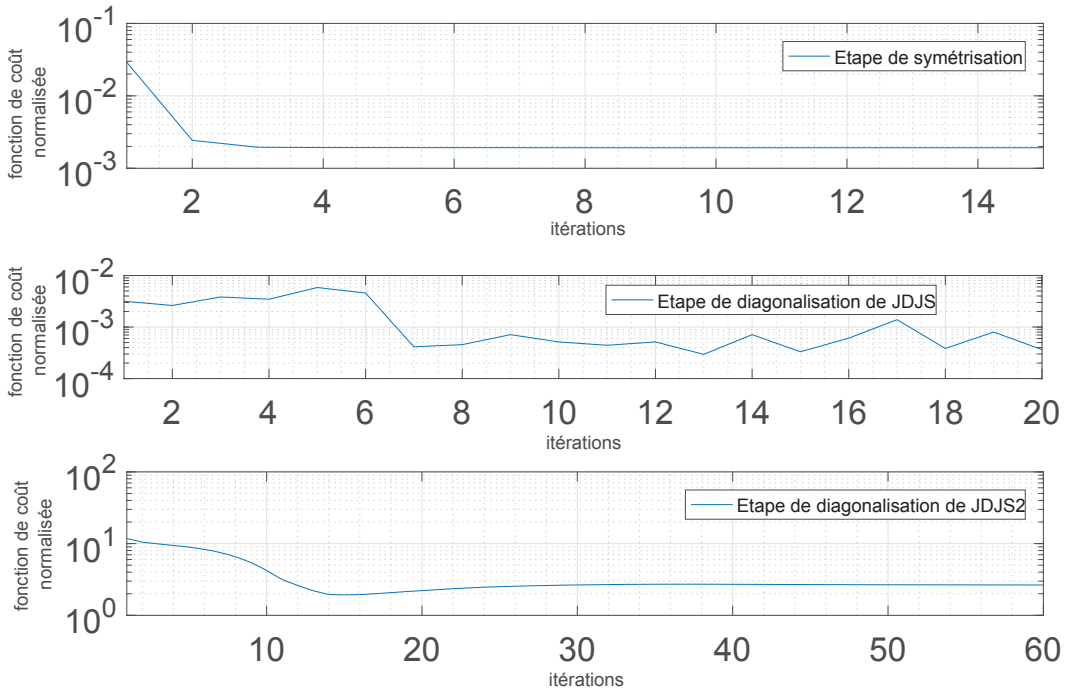

FIGURE 6.10: Évolution du critère de symétrisation C_{sym} (5.7) normalisée, puis du critère C_{inverse} (3.14) normalisée pour JDJS et JDJS2 en fonction du nombre d'itérations .

que sur les 100 initialisations différentes de PROCO-ALS, 45 d'entre elles fournissent des résultats moins précis que celui obtenu par SSD-JDJS2. De même, sur les 100 initialisations différentes de ADMoM, 52 d'entre elles fournissent un résultat moins précis que SSD-JDJS2.

Cette étude sur ce jeu de données ne nous permet pas de faire des conclusions générales sur les performances de nos algorithmes. Toutefois, nous avons montré que dans ce cas, l'algorithme SSD-JDJS2 améliore les performances obtenues par DIAG-JDTM et par la moyenne des réalisations de ADMoM et PROCO-ALS. Bien que PROCO-ALS soit plus précis en médiane, ce dernier est sensible à l'initialisation choisie et a besoin de beaucoup plus d'itérations que SSD-JDJS2 pour converger. De plus, l'initialisation qui permet d'obtenir la meilleure estimation des matrices facteurs n'est pas forcément celle qui fait le plus décroître l'erreur de reconstruction du tenseur, l'initialisation de PROCO-ALS ou de ADMoM est donc difficile à choisir en pratique. Enfin, les résultats fournis par DIAG-ALS, DIAG-ADMM et DIAG-ALS+ permettent de confirmer que la prise en compte des contraintes de non-négativité augmente la précision des algorithmes et diminue leur sensibilité à l'initialisation. Cependant, ces trois méthodes ne donnent pas des résultats assez précis ici pour être exploités.

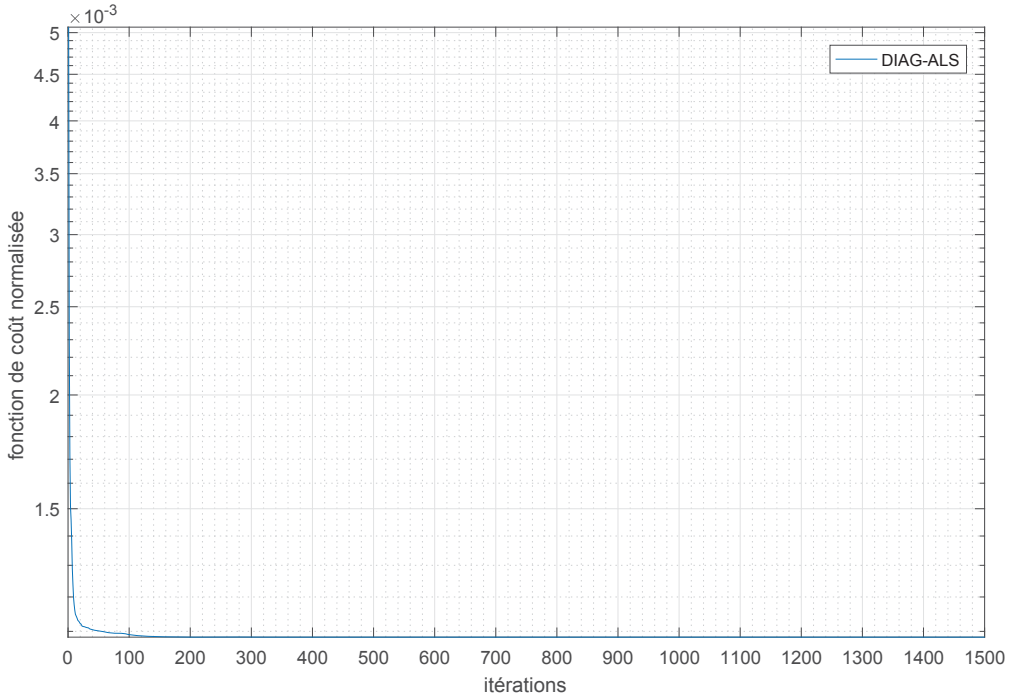

FIGURE 6.11: Évolution de la médiane de l'erreur de reconstruction de l'ensemble de matrices de DCS normalisée en fonction du nombre d'itérations pour DIAG-ALS.

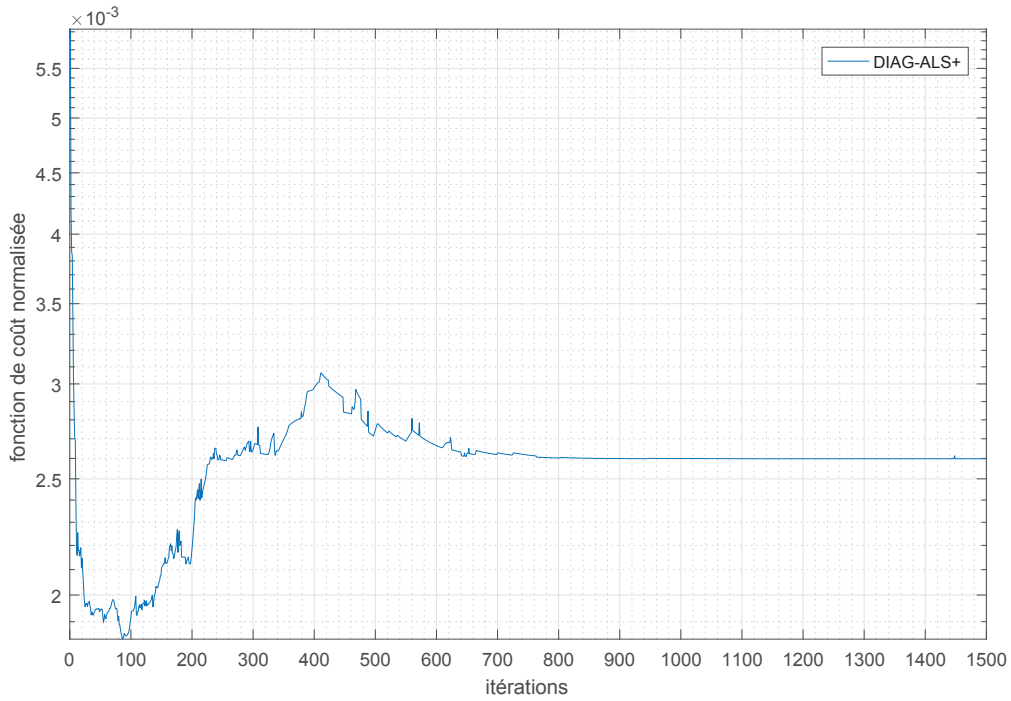

FIGURE 6.12: Évolution de la médiane de l'erreur de reconstruction de l'ensemble de matrices de DCS normalisée en fonction du nombre d'itérations pour DIAG-ALS+.

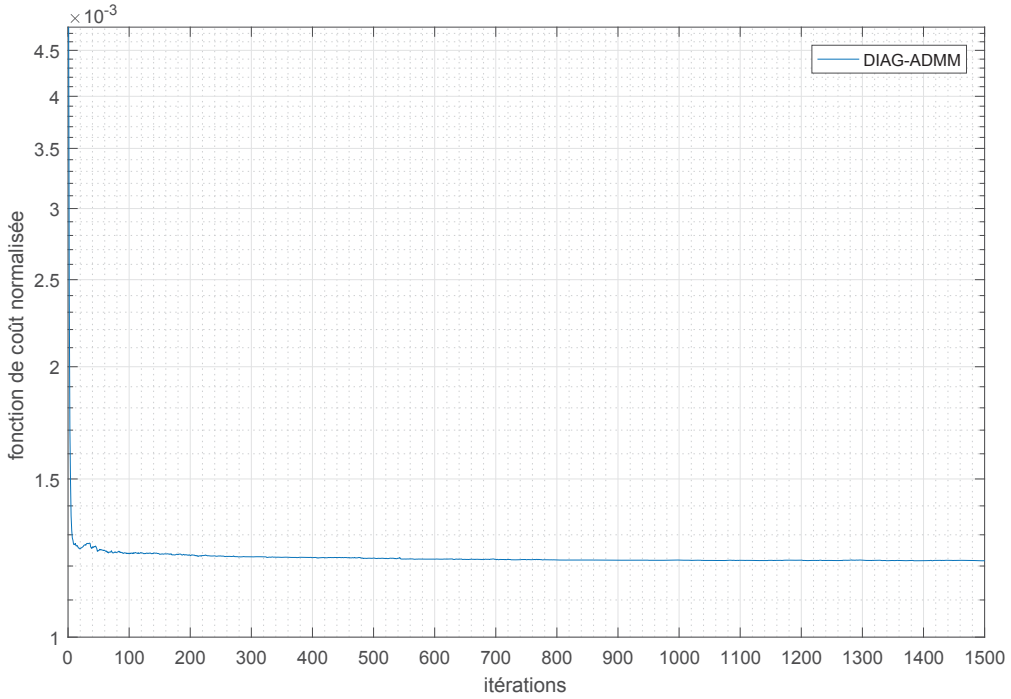

FIGURE 6.13: Évolution de la médiane de l'erreur de reconstruction de l'ensemble de matrices de DCS normalisée en fonction du nombre d'itérations pour DIAG-ADMM.

Conclusion et perspectives

Dans cette thèse, nous avons proposé plusieurs méthodes de diagonalisation conjointe de matrices par similitude.

Les algorithmes de diagonalisation conjointe par similitude sont habituellement développés en deux versions, l'une pour traiter des données réelles et l'autre pour traiter des données complexes. Dans le cas de données complexes, une catégorie d'algorithmes de la littérature fournit généralement des résultats précis de manière peu couteuse lorsque le niveau de bruit est faible alors que l'autre catégorie d'algorithmes de la littérature est plus robuste au bruit mais a un coût calcul bien plus élevé. Nous proposons donc, dans un premier temps, deux nouvelles classes de méthodes permettant de traiter des données réelles et complexes sans modifications. Celles-ci sont basées sur l'hypothèse que nous sommes proches de la solution recherchée. Cette hypothèse permet d'estimer simultanément les paramètres inconnus de notre problème de manière analytique.

La première classe utilise un développement de Taylor et nous a permis de développer deux algorithmes. Le premier (JDTE) estime la matrice diagonalisante de manière globale tandis que l'autre (SJDTE) estime la matrice diagonalisante par le biais d'un balayage par paire.

Les algorithmes de la seconde classe ont une structure commune et estiment la matrice diagonalisante à l'aide d'un balayage par paire. Leur différence réside dans la paramétrisation de la matrice de mise à jour à l'aide de différentes factorisations matricielles : classiques, peu connues (décomposition polaire algébrique) ou nouvelles (factorisation QR algébrique). De plus, nous montrons que nous pouvons passer d'une paramétrisation à l'autre en ne changeant qu'un seul paramètre. Les décompositions polaire et QR que nous proposons ne sont pas celles classiquement utilisées pour traiter des données complexes, nous utilisons leur version dite algébrique. Celles-ci nous permettent d'estimer directement les paramètres complexes sans passer par $\mathbb{R}^2$.

Nos simulations numériques montrent que plusieurs de ces algorithmes améliorent les résultats obtenus par les algorithmes de la littérature en termes de précision et de coût calcul en particulier pour des données complexes. Cependant, ils sont sensibles à la taille des matrices de l'ensemble à diagonaliser. Cette sensibilité permet toutefois de faire des conclusions intéressantes. En effet, il apparait que la stratégie de balayage par paire est plus robuste aux erreurs d'approximation pour la première classe d'algorithmes et que la décomposition polaire est plus robuste aux erreurs d'approximation que la décomposition QR, elle même plus robuste que la décomposition LU pour la seconde classe. Ce problème de sensibilité aux matrices de grandes tailles est résolu en initialisant nos algorithmes avec une décomposition en valeurs propres généralisée. Nous appliquons enfin un algorithme de chacune des deux classes au problème de séparation de signaux DS-CDMA. Nous montrons

alors que nos algorithmes permettent de retrouver les signaux sources en peu d'itérations et de manière précise.

La diagonalisation conjointe par similitude étant généralement utilisée pour résoudre le problème de décomposition CP, nous proposons une autre catégorie de méthodes tirant avantage des contraintes du problème de décomposition CP non-négative, suite à une réduction de dimensions par SVD. Nos travaux sur les méthodes de diagonalisation conjointe par similitude prenant en compte les contraintes du problème de décomposition CP non-négative sont précurseurs.

Notre première idée permet seulement d'imposer une contrainte de positivité. Pour imposer la positivité sur les valeurs propres de matrices conjointement diagonalisables dans la même base de vecteurs propres, nous avons d'abord développé un algorithme de symétrisation conjointe de matrices par similitude. Cet algorithme transforme l'ensemble de matrices initial en un ensemble de matrices approximativement symétriques définies positives et nous permet d'utiliser des méthodes existantes de diagonalisation conjointe par congruence conservant la positivité des valeurs diagonales par construction. Ainsi l'algorithme de diagonalisation conjointe par similitude que nous proposons (JDJS) est basé sur une étape de symétrisation suivie d'une étape de diagonalisation. Nous avons aussi proposé une seconde version de cet algorithme (JDJS2) dans laquelle les matrices sont contraintes à être parfaitement définies positives suite à l'étape de symétrisation. Nous avons intégré JDJS à l'algorithme DIAG et JDJS2 à l'algorithme SSD-CP dans le but de résoudre le problème de décomposition CP non-négative. La prise en compte des contraintes de non-négativité est généralement utile dans les cas difficiles (surestimation du rang, facteurs corrélés...). Ainsi, nous montrons que nos algorithmes sont particulièrement efficaces lorsque les colonnes des matrices facteurs de la décomposition CP ont un haut degré de corrélation pour des niveaux faibles et moyens de bruit. De plus, nous avons réussi à montrer qu'une des deux méthodes (JDJS2) permet d'obtenir de bons résultats sur un jeu de données de spectroscopie de fluorescence issue de mesures expérimentales (signaux non-synthétiques).

Les algorithmes précédents permettent de prendre en compte seulement les contraintes sur les valeurs propres du problème de diagonalisation conjointe par similitude, or il existe aussi des contraintes sur la matrice de vecteurs propres. De plus, ces algorithmes permettent d'imposer seulement une contrainte de positivité et non pas de non-négativité. Pour prendre en compte les contraintes de non-négativité à la fois sur les valeurs propres et les vecteurs propres, nous avons proposé des méthodes alternées résolvant le problème de diagonalisation conjointe par similitude : un algorithme de type ALS projeté, un algorithme de type ADMM et un algorithme de type ALS sans contraintes. Ce dernier algorithme nous permet de démontrer l'intérêt de la prise en compte de la non-négativité et aussi d'apprécier le comportement des méthodes de type alternées en comparaison avec les méthodes classiques basées sur des mises à jour multiplicatives. Les simulations numériques ont montré que la prise en compte de l'ensemble des contraintes améliore les performances pour les méthodes de type alterné. De plus, ces trois algorithmes permettent d'obtenir de meilleurs résultats que ceux obtenus par les algorithmes avec lesquels ils sont comparés lorsque le niveau de bruit est important et que les colonnes des matrices facteurs de la décomposition CP ont un haut degré de corrélation.

Les travaux présentés dans cette thèse ouvrent plusieurs perspectives. Nous avons développé des méthodes de type ALS projeté ou ADMM pour prendre en compte les contraintes de non-négativité offertes par le problème de décomposition CP non-négative. Or, il existe bien d'autres méthodes pour mettre en œuvre ce type de contraintes. Les méthodes basées sur des mises à jour multiplicatives ayant de meilleurs propriétés de convergence que les méthodes alternées, il serait judicieux de développer des méthodes de diagonalisation conjointe par similitude avec mises à jour multiplicatives prenant en compte la totalité des contraintes du problème de décomposition CP non-négative. Par ailleurs, nous avons vu dans le chapitre 2 que la SVD n'est pas le seul moyen de réduire les dimensions d'un tenseur, il pourrait aussi être intéressant de développer des algorithmes sous contraintes suite à une compression par HOSVD.

Le problème de diagonalisation conjointe par similitude n'est pas seulement utilisé pour le problème de décomposition CP (il peut être aussi utilisé pour de l'analyse en composante indépendante par exemple). Appréhender les contraintes offertes par les autres problèmes dans lesquels la diagonalisation conjointe par similitude intervient pourrait permettre de mettre en œuvre de nouveaux algorithmes.

Les méthodes que nous proposons pour conserver la positivité de la diagonale des matrices sont basées sur une étape de symétrisation. Il serait astucieux de trouver une fonction de coût propre au problème de diagonalisation conjointe par similitude dans le but de contraindre les valeurs propres estimées à être non-négatives sans passer par une étape de symétrisation. Inversement, on pourrait aussi chercher d'autres applications à notre algorithme de symétrisation.

Enfin, les conditions de fonctionnement des algorithmes DIAG et SSD-CP étant plus contraignantes que les conditions d'autres algorithmes de décomposition CP, il serait intéressant de trouver une nouvelle manière de transformer le problème de décomposition CP en un problème de diagonalisation conjointe imposant des conditions moins fortes.

Annexe A

Annexe

A.1 Démonstrations des propositions 8 et 9

Démonstration de la proposition 8. L'équation (4.54) est équivalente à

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

Où $\mathbf{\Lambda}$ est une matrice diagonale inversible. Ainsi, nous devons trouver une condition nécessaire et suffisante pour laquelle le système suivant a une solution pour q et r :

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

Nous rappelons que $\tilde{X}_{2,2} \neq 0$, $\tilde{X}_{1,1} \neq 0$ et $\det(\tilde{\mathbf{X}}) = \tilde{X}_{1,1}\tilde{X}_{2,2} - \tilde{X}_{2,1}\tilde{X}_{1,2} = 1$.

Nous pouvons distinguer quatre cas :
cas 1 $\tilde{X}_{1,2} = \tilde{X}_{2,1} = 0$
Dans ce cas la matrice $\tilde{\mathbf{X}}$ est déjà une matrice diagonale, ainsi nous avons la solution triviale $r = q = 0$.

cas 2 $\tilde{X}_{1,2} = 0$ et $\tilde{X}_{2,1} \neq 0$
Dans ce cas, le système (A.2) devient :

$$\begin{cases} r = q \\ \tilde{X}_{2,1}q^2 - q\tilde{X}_{2,2} + \tilde{X}_{2,1} = 0 \\ 1+q^2 \neq 0. \end{cases} \tag{A.3}$$

Le polynôme en q de ce système a toujours une solution dans $\mathbb{C}$. La condition $q^2 \neq -1$ est assurée alors par $\tilde{X}_{2,2} \neq 0$.

cas 3 $\tilde{X}_{2,1} = 0$ et $\tilde{X}_{1,2} \neq 0$
Nous avons directement $q = 0$ et $r = \frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}}$.

cas 4 $\tilde{X}_{2,1} \neq 0$ et $\tilde{X}_{1,2} \neq 0$
Dans ce cas, le système (A.2) devient :

$$\begin{cases} r = \frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}} + q \\ \tilde{X}_{2,1}\tilde{X}_{1,1}q^2 - q + \tilde{X}_{2,1}\tilde{X}_{1,1} = 0 \\ 1 + q^2 \neq 0. \end{cases} \tag{A.4}$$

Le polynôme en q de ce système a toujours une solution dans $\mathbb{C}$ et la troisième équation implique $q^2 \neq -1$.

Ainsi, peu importe le cas considéré, la matrice $\tilde{\mathbf{X}}$ pourra toujours être paramétrée à l'aide d'une factorisation QR algébrique.

□

Démonstration de la proposition 9. Nous utilisons le même raisonnement que pour la proposition 8 :
Ainsi nous cherchons à montrer qu'il existe toujours h et q tels que la matrice

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

soit une matrice diagonale et inversible. Nous cherchons donc sous quelles conditions le système suivant à une solution :

$$\begin{cases} \tilde{X}_{1,2}qh + \tilde{X}_{1,1}q - \tilde{X}_{1,1}h + \tilde{X}_{1,2} = 0 \\ -\tilde{X}_{2,1}qh - \tilde{X}_{2,2}q - \tilde{X}_{2,2}h + \tilde{X}_{2,1} = 0 \\ 1 + q^2 \neq 0 \\ 1 - h^2 \neq 0. \end{cases} \tag{A.5}$$

Cette fois-ci encore, nous distinguons quatre cas :
case 1 $\tilde{X}_{1,2} = \tilde{X}_{2,1} = 0$
Ce cas est trivial puisque $\tilde{\mathbf{X}}$ est déjà une matrice diagonale inversible. Nous avons donc $q = h = 0$ pour solution.

case 2 $\tilde{X}_{1,2} = 0$ et $\tilde{X}_{2,1} \neq 0$
Nous pouvons facilement vérifier dans ce cas que $q = h$ avec q vérifiant :

$$-\tilde{X}_{2,1}q^2 - 2\tilde{X}_{2,2}q + \tilde{X}_{2,1} = 0. \tag{A.6}$$

Ce polynôme a toujours une solution dans $\mathbb{C}$. La condition $q^2 \neq -1$ implique $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}})^2 \neq -1$. La condition $h^2 \neq 1$ est assurée par $\tilde{X}_{2,2} \neq 0$.

case 3 $\tilde{X}_{1,2} \neq 0$ et $\tilde{X}_{2,1} = 0$
Il est facile de vérifier ici que $q = -h$ avec q une solution de

$$-\tilde{X}_{1,2}q^2 + 2\tilde{X}_{1,1}q + \tilde{X}_{1,2} = 0. \tag{A.7}$$

Ce polynôme a toujours une solution dans $\mathbb{C}$. Ici la condition $q^2 \neq -1$ est équivalente à $(\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}})^2 \neq -1$. La condition $h^2 \neq 1$ est assurée par $\tilde{X}_{1,1} \neq 0$.

case 4 $\tilde{X}_{1,2} \neq 0$ et $\tilde{X}_{2,1} \neq 0$
Nous avons ici :

$$\begin{cases} \alpha q - \beta h + \gamma = 0 \\ \beta h^2 + 2h - \beta = 0 \\ q^2 \neq -1 \\ h^2 \neq 1. \end{cases} \tag{A.8}$$

Avec

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

Nous devons ici étudier quatre sous-cas en fonction des valeurs de α et β :

a) $\alpha = 0$ et $\beta = 0$

Ce cas est impossible car $\tilde{X}_{2,1} \neq 0$ et $\tilde{X}_{1,2} \neq 0$.

b) $\alpha \neq 0$ et $\beta = 0$

Ici, une solution est donnée par $q = -\frac{\gamma}{\alpha} = -\frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}}$ et $h = 0$. Ainsi si $(\frac{\tilde{X}_{2,1}}{\tilde{X}_{1,1}})^2 \neq -1$ alors $q^2 \neq -1$.

c) $\alpha = 0$ et $\beta \neq 0$

Ici, une solution est donnée par $q = 0$ et $h = \frac{\gamma}{\beta} = \frac{\tilde{X}_{1,2}}{\tilde{X}_{1,1}} = \frac{\tilde{X}_{2,1}}{\tilde{X}_{2,2}}$. Ici $h^2 = 1$ est incompatible avec $\det\{\tilde{\mathbf{X}}\} = 1$.

d) $\alpha \neq 0$ and $\beta \neq 0$

Ici, il est clair que $h \neq 1$ et $h \neq -1$ (sinon la seconde équation dans (A.8) serait fausse). La première équation de (A.8) donne $h = \frac{\alpha q + \gamma}{\beta}$. En reportant cette expression dans la deuxième équation nous avons :

$$\alpha^2 q^2 + (2\alpha\gamma + 2\alpha)q + \gamma^2 + 2\gamma - \beta^2 = 0 \tag{A.9}$$

Ce polynôme a au moins une solution. Après développement, nous pouvons montrer que :

$$q^2 \neq -1 \Leftrightarrow \alpha \neq \pm i(1+\gamma) \tag{A.10}$$

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

□

A.2 Simulation numérique des méthodes du chapitre 4 dans le cas de données réelles

Dans le but de montrer que les méthodes proposées dans le chapitre 4 fonctionnent sur des données réelles, nous proposons une simulation numérique identique à celle du scénario 1 du chapitre 4 où les matrices à diagonaliser contiennent cette fois-ci des nombres réels.

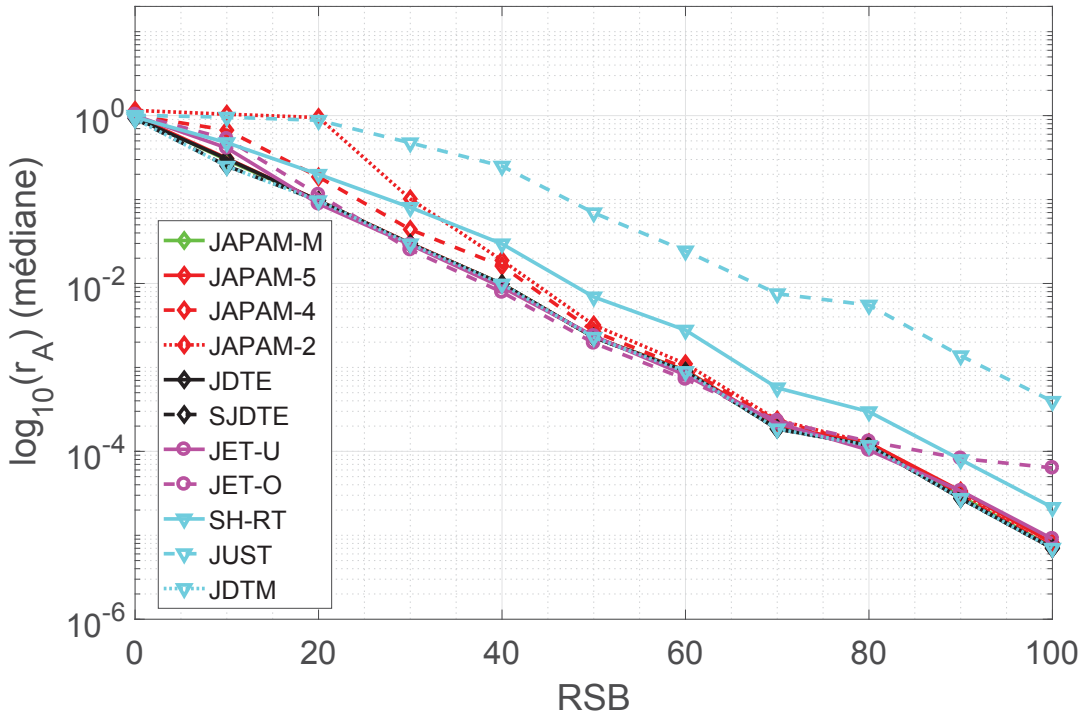

FIGURE A.1: r_A médian en fonction du RSB.

La médiane du critère r_A en fonction du RSB est tracée sur la figure A.1. Les algorithmes de l'état de l'art fournissant les meilleurs résultats sont JET-U et JDTM. JUST est globalement le moins performant en médiane de tous les algorithmes comparés. Tous les algorithmes que nous proposons sont aussi performants que JDTM et JET-U au-dessus de 60 dB. Sous 60 dB, JAPAM-4 et JAPAM-2 donnent de moins bons résultats mais les autres algorithmes proposés restent en compétition avec JET-U, JET-O et JDTM. JAPAM-2 est meilleur que SH-RT au dessus de 30 dB, puis devient moins performant que ce dernier sous cette valeur.

La moyenne du critère r_A en fonction est disponible sur la figure A.2. JUST reste le moins performant de tous les algorithmes de cette étude. JAPAM-M, JAPAM-5, JET-O, JET-U et JDTM sont fournissent la meilleure estimation moyenne et ont des performances quasiment équivalentes. JAPAM-4 et SH-RT estiment moins bien la matrice diagonalisante que les algorithmes précédemment cités et fournissent des résultats similaires entre eux. JAPAM-2 est meilleur que SH-RT au-dessus de 20 dB.

Ainsi, grâce à cette simulation nous avons montré que les algorithmes proposés au chapitre 4 fonctionnent sur des données réelles et fournissent des résultats équivalents à ceux des meilleurs algorithmes existants.

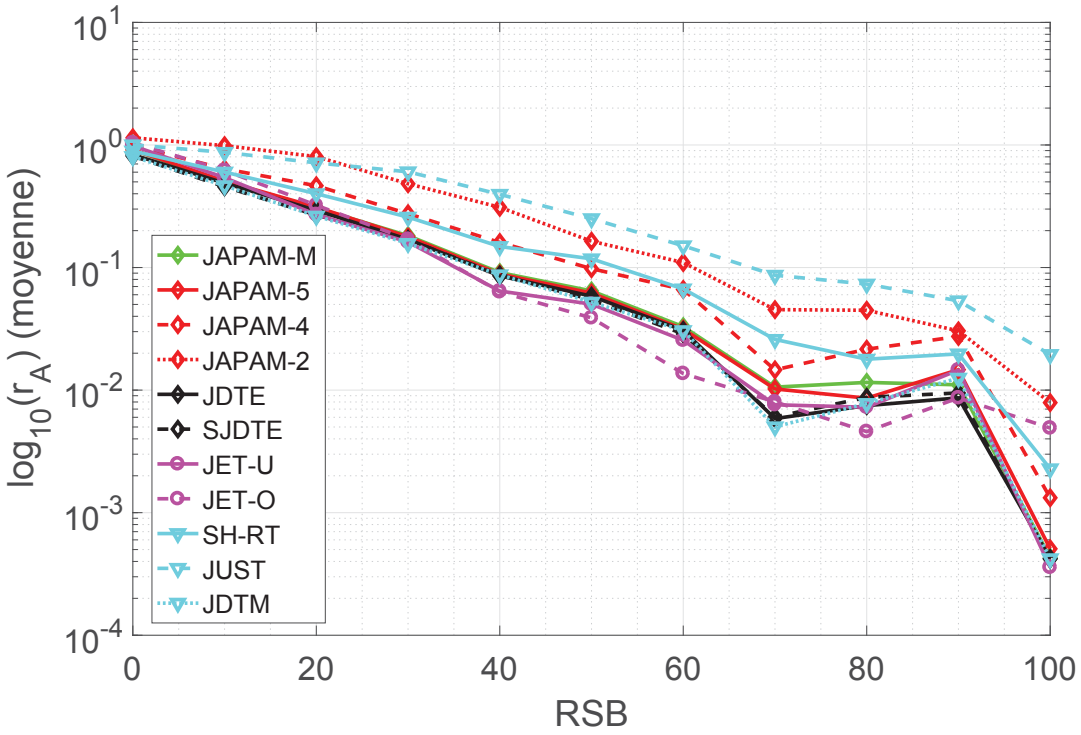

FIGURE A.2: r_A moyen en fonction du RSB.

A.3 Méthode de symétrisation à l'aide d'une matrice symétrique

Nous proposons ici de symétriser les matrices $\mathbf{M}^{(k)}$ issues du problème de DCS à l'aide d'une matrice symétrique de déterminant 1 de la forme :

$$\mathbf{H}^{(i,j)} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & h_1^{(i,j)} & \ddots & h_2^{(i,j)} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & h_2^{(i,j)} & \ddots & h_3^{(i,j)} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}, \tag{A.11}$$

avec $h_1^{(i,j)}h_3^{(i,j)} - h_2^{(i,j)2} = 1$. Pour se faire, nous effectuons une décomposition polaire sur la matrice $\mathbf{L}^{(i,j)}$ de l'équation (5.5) du chapitre 5. Nous pouvons donc écrire $\mathbf{L}^{(i,j)} = \mathbf{H}^{(i,j)}\mathbf{Q}^{(i,j)}$, où $\mathbf{Q}^{(i,j)}$ est une matrice de rotation de Givens définie dans la proposition 3 du chapitre 3. Il est important de noter qu'une transformation telle que (5.6) par une matrice orthogonale n'affecte pas la symétrie des matrices de l'ensemble. Il n'est donc pas nécessaire d'estimer la matrice $\mathbf{Q}^{(i,j)}$. Cependant, il est toujours nécessaire d'estimer une matrice diagonale comme explicité au chapitre 5. De plus, comme $\mathbf{L}^{(i,j)}$ ne dépend que d'un seul paramètre, nous devons chercher les liens entre $h_1^{(i,j)}$, $h_2^{(i,j)}$ et $h_3^{(i,j)}$. Nous avons

en effet le système d'équations suivant :

$$\begin{cases} h_1^{(i,j)}c + h_2^{(i,j)}s = 0, \\ -h_1^{(i,j)}s + h_2^{(i,j)}c = 0, \\ -h_2^{(i,j)}s + \frac{1+h_2^{(i,j)2}}{h_1^{(i,j)}}c = 0. \end{cases} \tag{A.12}$$

Les deux premières équations nous permettent d'établir :

$$\begin{cases} c = \frac{h_1^{(i,j)}}{h_1^{(i,j)2}+h_2^{(i,j)2}}, \\ s = \frac{h_2^{(i,j)}}{h_1^{(i,j)2}+h_2^{(i,j)2}}, \\ -h_2^{(i,j)}s + \frac{1+h_2^{(i,j)2}}{h_1^{(i,j)}}c = 0. \end{cases} \tag{A.13}$$

En injectant les expressions de c et s dans la troisième équation de (A.12), nous obtenons :

$$h_1^{(i,j)} = \sqrt{1 - h_2^{(i,j)2}}. \tag{A.14}$$

Grâce à cette dernière équation et au fait que le déterminant de $\mathbf{H}^{(i,j)}$ est égal à 1, nous pouvons chercher la matrice $\mathbf{H}^{(i,j)}$ de la forme :

$$\mathbf{H}^{(i,j)} = \sqrt{1 - h_2^{(i,j)2}} \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & \frac{h_2^{(i,j)}}{\sqrt{1-h_2^{(i,j)2}}} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \frac{h_2^{(i,j)}}{\sqrt{1-h_2^{(i,j)2}}} & \ddots & \frac{1+h_2^{(i,j)2}}{1-h_2^{(i,j)2}} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \tag{A.15}$$

En posant $h^{(i,j)} = \frac{h_2^{(i,j)}}{\sqrt{1-h_2^{(i,j)2}}}$, nous avons finalement :

$$\mathbf{H}^{(i,j)} = \frac{1}{\sqrt{1 + h^{(i,j)2}}} \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & h^{(i,j)} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & h^{(i,j)} & \ddots & 1 + h^{(i,j)2} & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \tag{A.16}$$

Le critère (5.7) devient alors :

$$C_{\text{sym}}(\mathbf{H}^{(i,j)}) = \sum_{k=1}^{K} \left(\sum_{\substack{p \neq i \\ p \neq j}} \left((N_{p,i}^{(k)'} - N_{i,p}^{(k)'})^2 + (N_{p,j}^{(k)'} - N_{j,p}^{(k)'})^2 \right) + (N_{i,j}^{(k)'} - N_{j,i}^{(k)'})^2 \right), \tag{A.17}$$

où $\mathbf{N}^{(k)'} = \mathbf{H}^{(i,j)}\mathbf{N}^{(k)}\mathbf{H}^{(i,j)-1}$.

En développant chacun des termes de (A.17), nous obtenons pour le premier terme :

$$(N_{p,i}^{(k)'} - N_{i,p}^{(k)'})^2 = \frac{\left[2N_{i,p}^{(k)}h^{(i,j)2} - (N_{p,j}^{(k)} + N_{j,p}^{(k)})h^{(i,j)} + (N_{i,p}^{(k)} - N_{p,i}^{(k)})\right]^2}{(1+h^{(i,j)2})^2}$$
$$= \frac{P^{(k,p,i,j)}(h^{(i,j)})}{(1+h^{(i,j)2})^2}. \tag{A.18}$$

Pour le second terme, nous obtenons :

$$(N_{j,p}^{(k)'} - N_{p,j}^{(k)'})^2 = \frac{\left[-2N_{p,j}^{(k)}h^{(i,j)2} - (N_{i,p}^{(k)} + N_{p,i}^{(k)})h^{(i,j)} + (N_{j,p}^{(k)} - N_{p,j}^{(k)})\right]^2}{(1+h^{(i,j)2})^2}$$
$$= \frac{Q^{(k,p,i,j)}(h^{(i,j)})}{(1+h^{(i,j)2})^2}. \tag{A.19}$$

Et enfin pour le dernier terme, nous avons :

$$(N_{i,j}^{(k)'} - N_{j,i}^{(k)'})^2 = \frac{1}{(1+h^{(i,j)2})^2}\big[4N_{i,j}^{(k)}h^{(i,j)4} - 2(N_{i,i}^{(k)} + N_{j,j}^{(k)})h^{(i,j)3} + (5N_{i,j}^{(k)} - N_{j,i}^{(k)})h^{(i,j)2} +$$
$$2(N_{i,i}^{(k)} - N_{j,j}^{(k)})h^{(i,j)} + (N_{i,j}^{(k)} - N_{j,i}^{(k)})\big]^2$$
$$= \frac{R^{(k,i,j)}(h^{(i,j)})}{(1+h^{(i,j)2})^2}. \tag{A.20}$$

En posant

$$S^{(i,j)}(h^{(i,j)}) = \sum_{k=1}^{K} R^{(k,i,j)}(h^{(i,j)}) + \sum_{\substack{p\neq i\\ p\neq j}} \big(P^{(k,p,i,j)}(h^{(i,j)}) + Q^{(k,p,i,j)}(h^{(i,j)})\big), \tag{A.21}$$

nous pouvons réécrire (A.17) de la manière suivante :

$$C_{\text{sym}}(\mathbf{H}^{(i,j)}) = \frac{S^{(i,j)}(h^{(i,j)})}{(1+h^{(i,j)2})^2}. \tag{A.22}$$

Pour minimiser ce critère, nous annulons $\frac{\partial C_{\text{sym}}(\mathbf{H}^{(i,j)})}{\partial h^{(i,j)}}$. Cela est équivalent à annuler le polynôme de degré 9 suivant :

$$\frac{\partial S^{(i,j)}(h^{(i,j)})}{\partial h^{(i,j)}}(1+h^{(i,j)2}) - 4h^{(i,j)}S^{(i,j)}(h^{(i,j)}). \tag{A.23}$$

Remarque :

1. Pour symétriser un ensemble de matrices, il aussi possible de minimiser le critère suivant :

$$C'_{\text{sym}}(\mathbf{H}^{(i,j)}) = \sum_{k=1}^{K}(N_{i,j}^{(k)'} - N_{j,i}^{(k)'})^2. \tag{A.24}$$

Pour se faire, il faut annuler le polynôme de degré 9

$$\frac{\partial R^{(i,j)}(h^{(i,j)})}{\partial h^{(i,j)}}(1+h^{(i,j)2}) - 4h^{(i,j)}R^{(i,j)}(h^{(i,j)}). \tag{A.25}$$

2. Nous pouvons remarquer qu'avec cette paramétrisation, la matrice $\mathbf{H}^{(i,j)}$ tend vers la matrice identité lorsque $h^{(i,j)}$ tend vers 0. Cela permet de proposer des algorithmes moins coûteux en approximant les termes de plus hauts degrés de $C_{\text{sym}}(\mathbf{H}^{(i,j)})$ ou $C'_{\text{sym}}(\mathbf{H}^{(i,j)})$.

3. Pour symétriser un ensemble de matrice à l'aide d'une matrice symétrique, il est aussi possible d'appliquer une décomposition polaire sur les matrices $\mathbf{X}^{(i,j)}$ définie à la page 60. La matrice de mise à jour symétrique sera alors une matrice de rotation hyperbolique définie comme à la page 67 et l'estimation de ses paramètres sera identique à l'estimation des paramètres de la matrice symétrique de l'algorithme JDTM [22].

Bibliographie

[1] J. Delabrouille, J.-F. Cardoso, and G. Patanchon, "Multidetector multicomponent spectral matching and applications for cosmic microwave background data analysis," *Monthly Notices of the Royal Astronomical Society*, vol. 346, no. 4, pp. 1089–1102, 2003.

[2] S. Moussaoui, H. Hauksdottir, F. Schmidt, C. Jutten, J. Chanussot, D. Brie, S. Doute, and J.A. Benediktsson, "On the decomposition of mars hyperspectral data by ica and bayesian positive source separation," *Neurocomputing*, vol. 71, pp. 2294–2208, 2008.

[3] G. Chabriel and J. Barrere, "An instantaneous formulation of mixtures for blind separation of propagating waves," *IEEE Transactions on Signal Processing*, vol. 54, no. 1, pp. 49–58, Jan 2006.

[4] Z. Koldovsky and P. Tichavsky, "Time-domain blind separation of audio sources on the basis of a complete ica decomposition of an observation space," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 406–416, Feb 2011.

[5] M. Congedo, C. Gouy-Pailler, and C. Jutten, "On the blind source separation of human electroencephalogram by approximate joint diagonalization of second order statistics," *Clinical Neurophysiology*, vol. 119, no. 12, pp. 2677 – 2686, 2008.

[6] L. De Lathauwer, D. Callaerts, B. De Moor, and J. Vandewalle, "Fetal electrocardiogram extraction by source subspace separation," in *IEEE Workshop on Higher Order Statistics*, Girona, Spain, June 12-14 1995, pp. 134–138.

[7] J. Herault, C. Jutten, and B. Ans, "Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé," in *GRETSI 85, Dixième colloque sur le Traitement du Signal et des Images*, Nice, France, Septembre 1985, pp. 1017–1022.

[8] T. He, G. Clifford, and L. Tarassanko, "Application of ica in removing artefacts from the ECG," *Neural Computating & Applications*, vol. 15, no. 2, pp. 105–116, Apr. 2006.

[9] 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," *Bulletin of the Polish Academy of Sciences - Technical sciences*, vol. 60, no. 3, pp. 407–418, 2012.

[10] C. Paulus and J. I. Mars, "New multicomponent filters for geophysical data processing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 8, pp. 2260–2270, Aug 2006.

[11] N.Thirion, *Séparation d'ondes en prospection sismique*, Ph.D. thesis, Institut national polytechnique, 1995.

[12] 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," *Signal Process.*, vol. 86, no. 8, pp. 1950–1967, Aug. 2006.

[13] M. Castella and E. Moreau, "Generalized identifiability conditions for blind convolutive mimo separation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2846–2852, July 2009.

[14] P. Chevalier, L. Albera, P. Comon, and A. Ferreol, "Comparative performance analysis of eight blind source separation methods on radiocommunications signals," in *IJCNN 04, 2004 IEEE International Joint Conference on Neural Networks*, Budapest, Hungary, July 25-29 2004.

[15] A. Ferreol, P. Chevalier, and L. Albera, "Second order blind separation of first and second order cyclostationary sources — Application to AM, FSK, CPFSK and deterministic sources," *IEEE Transactions On Signal Processing*, vol. 52, pp. 845–861, Apr. 2004.

[16] D. Nion and L. De Lathauwer, "An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA," *Signal Processing, Elsevier*, vol. 88, no. 3, pp. 749–755, Mar. 2008.

[17] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," *IEEE Transactions On Signal Processing*, vol. 48, no. 8, pp. 810–823, March 2000.

[18] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis with applications in the chemical sciences*, John Wiley & Sons Ltd, England, 2004.

[19] R. Bro, *Multi-way Analysis in the Food Industry : Models, Algorithms, and Applications*, Ph.D. thesis, University of Amsterdam, 1998.

[20] R. Bro, "Parafac, tutorial and applications," *Chemom. Intel. Lab. Syst.*, vol. 38, pp. 149–171, 1997.

[21] X. Luciani, *Analyse numérique des spectres de fluorescence 3D issus de mélanges non linéaires*, Ph.D. thesis, 2007, Thèse de doctorat dirigée par Bois, André Sciences. Traitement du signal Toulon 2007.

[22] X. Luciani and L. Albera, "Canonical polyadic decomposition based on joint eigenvalue decomposition," *Chemometrics and Intelligent Laboratory Systems*, vol. 132, no. 0, pp. 152 – 167, 2014.

[23] R. A. Harshman, "Foundation of PARAFAC procedure : Models and conditions for an 'explanatory' multi-mode factor analysis," *UCLA working papers in Phonetics*, , no. 16, pp. 1–84, 1970.

[24] C. J. Appellof and E. R. Davidson, "Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents," *Analytical Chemistry*, vol. 53, no. 13, pp. 2053–2056, 1981.

[25] M. Alex O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles : Tensorfaces," in *Computer Vision — ECCV 2002*, Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, Eds., Berlin, Heidelberg, 2002, pp. 447–460, Springer Berlin Heidelberg.

[26] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and Bülent Y., "Modeling and multiway analysis of chatroom tensors," in *Intelligence and Security Informatics*, Paul Kantor, Gheorghe Muresan, Fred Roberts, Daniel D. Zeng, Fei-Yue Wang, Hsinchun Chen, and Ralph C. Merkle, Eds., Berlin, Heidelberg, 2005, pp. 256–268, Springer Berlin Heidelberg.

[27] C.F. Beckmann and S.M. Smith, "Tensorial extensions of independent component analysis for multisubject fmri analysis," *NeuroImage*, vol. 25, no. 1, pp. 294 – 311, 2005.

[28] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[29] 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," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, March 2015.

[30] F. L. Hitchcock, "Multiple invariants and generalized rank of a p-way matrix or tensor," *J. Math. and Phys.*, vol. 7, no. 1, pp. 39–79, 1927.

[31] L. R. Tucker, "Implications of factor analysis of three-way matrices for measurement of change," in *Problems in Measuring Change*, C. W. HARRIS, Ed., pp. 122–137. University of Wisconsin Press, 1963.

[32] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep 1966.

[33] J. Carroll and J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition," *Psychometrika*, , no. 9, pp. 267–283, 1970.

[34] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, Mar. 2000.

[35] B. Savas and L. Eldén, "Handwritten digit classification using higher order singular value decomposition," *Pattern Recognition*, vol. 40, no. 3, pp. 993 – 1003, 2007.

[36] P. Comon and C. Jutten, "Handbook of blind source separation," Academic Press, March 2010.

[37] M. Rajih and P. Comon, "Alternating least squares identification of underdetermined mixtures based on the characteristic function," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, May 2006, vol. 3, pp. III–III.

[38] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Computation of the canonical decomposition by means of a simultaneous Schur decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 295–327, 2004.

[39] F. Roemer and M. Haardt, "A closed-form solution for multilinear parafac decompositions," in *SAM 08, Fifth IEEE Sensor Array and Multichannel Signal Processing Workshop*, july 2008, pp. 487–491.

[40] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other thales," *Journal of Chemometrics*, vol. 23, april 2009.

[41] Q. Zhang, H. Wang, R. J. Plemmons, and V. Pau'l Pauca, "Tensor methods for hyperspectral data analysis : a space object material identification study," *J. Opt. Soc. Am. A*, vol. 25, no. 12, pp. 3001–3012, Dec 2008.

[42] A. Cichocki, R. Zdunek, A. Phan, and S. Amari, Nonnegative matrix and tensor factorization *Unsupervised Adaptive Filtering, Vol. I, Blind Source Separation*, Wiley, november 2009.

[43] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals," *IEE Proceesings-F*, vol. 140, no. 6, pp. 362–370, Dec. 1993.

[44] M. Wax and J. Sheinvald, "A least-squares approach to joint diagonalization," *IEEE Signal Processing Letters*, vol. 4, no. 2, pp. 52–53, Feb 1997.

[45] G. Hellbourg, T. Trainini, R. Weber, E. Moreau, C. Capdessus, and A. J. Boonstrd, "Rfi subspace estimation techniques for new generation radio telescopes," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug 2012, pp. 200–204.

[46] X. Huang, H. C. Wu, and J. C. Principe, "Robust blind beamforming algorithm using joint multiple matrix diagonalization," *IEEE Sensors Journal*, vol. 7, no. 1, pp. 130–136, Jan 2007.

[47] P. Comon, E. Moreau, and L. Rota, "Blind separation of convolutive mixtures : A contrast-based joint diagonalization approach," in *in 3rd Int. Conf. Independent Component Analysis, (San Diego*, 2001, pp. 686–691.

[48] X. Liu, "Joint estimation of angle and doppler frequency in mimo radar," in *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, June 2013, pp. 231–235.

[49] B. Afsari and P. S. Krishnaprasad, *Some Gradient Based Joint Diagonalization Methods for ICA*, pp. 437–444, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[50] X. Luciani and L. Albera, "Joint eigenvalue decomposition of non-defective matrices based on the lu factorization with application to ica," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4594–4608, Sept 2015.

[51] E. Moreau and T. Adali, *Blind Identification and Separation of Complex-valued Signals*, FOCUS Digital Signal and Image Processing Series. 2013.

[52] C. G. J. Jacobi, "Uber ein leichtes verfahren die in der theorie der sacularstorungen vorkommenden gleichungen numerisch aufzulosen," *Journal fur die reine und angewandte Mathematik*, vol. 1846, no. 30, pp. 51–94, January 1846.

[53] J. F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM Journal Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161–164, 1996.

[54] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1999.

[55] R. A. Horn and C. R. Johnson, *Matrix Analysis 2nd edition*, Cambridge University Press, New York, 2012.

[56] R. A. Horn, "The hadamard product," vol. 40, pp. 87–169, 5 1990.

[57] J. W. Brewer, "Kronecker products and matrix calculus in system theory," vol. 25, pp. 772 – 781, 10 1978.

[58] C. G. Khatri and C. R. Rao, "Solutions to some functional equations and their applications to characterization of probability distributions," *Sankhya : Indian J. Statistics*, vol. Series A, 30, pp. 167–180, 1968.

[59] I. Kaplansky, "Algebraic polar decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 11, no. 2, pp. 213–217, 1990.

[60] P. Mccullagh, *Tensor Methods in Statistics*, Chapman and Hall, Monographs on Statistics and Applied Probability, 1987.

[61] L. De Lathauwer, *Signal processing based on multilinear algebra*, Ph.D. thesis, Katholieke Universiteit, 1997.

[62] P. Comon, "Tensors : A brief introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 44–53, May 2014.

[63] C. Andersson and R. Bro, "Improving the speed of multi-way algorithms. part i : Tucker3," vol. 42, pp. 93–103, 08 1998.

[64] J. B. Kruskal, "Three-way arrays : Rank and uniqueness of trilinear decompositions," *Linear Algebra and Applications*, vol. 18, pp. 95–138, 1977.

[65] R. A. Harshman and M. E. Lundy, "The PARAFAC model for three-way factor analysis and multidimensional scaling," in *Research Methods for Multi-mode Data Analysis*, H. G. LAW, C. W. SNYDER, J. A. HATTIE, and R. P. MCDONALD, Eds., Praeger, New York, 1984, vol. 16, pp. 122–215.

[66] N. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of n-way arrays," *Journal of Chemometrics*, vol. 14, pp. 229–239, 2000.

[67] A. Stegeman and N. D. Sidiropoulos, "On kruskal's uniqueness condition for the candecomp/parafac decomposition," *Linear Algebra and its Applications*, vol. 420, no. 2, pp. 540 – 552, 2007.

[68] P. Comon, J.M.F. ten Berge, L. De Lathauwer, and J. Castaing, "Generic and typical ranks of multi-way arrays," *Linear Algebra and its Applications*, vol. 430, no. 11, pp. 2997 – 3007, 2009.

[69] R. Bro and H. A. Kiers, "A new efficient method for determinating the number of components in parafac models," *Chemometrics*, vol. 17, pp. 274–286, 2003.

[70] F. J. Király and A. Ziehe, "Approximate rank-detecting factorization of low-rank tensors," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3938–3942.

[71] K. G. Gauss, *Theory of Motion of the Heavenly Bodies*, Dover, New York, 1963.

[72] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the parafac model," *Computational Statistics & Data Analysis*, vol. 50, no. 7, pp. 1700–1734, 2006.

[73] W. S. Rayens and B. C. Mitchell, "Two-factor degeneracies and a stabilization of parafac," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, pp. 173–181, 1997.

[74] P. Paatero, "Construction and analysis of degenerate parafac models," *Journal of Chemometrics*, vol. 14, no. 3, pp. 285–299, 2000.

[75] B. C. Mitchell and D. S. Burdick, "Slowly converging parafac sequences : Swamps and two-factor degeneracies," *Journal of Chemometrics*, vol. 8, no. 2, pp. 155–168, 1994.

[76] H.A.L. Kiers, "A three-step algorithm for candecomp/parafac analysis of large data sets with multicollinearity," *Journal of Chemometrics*, vol. 12, no. 3, pp. 155–171, 1998.

[77] M. Rajih, P. Comon, and R. Harshman, "Enhanced line search : A novel method to accelerate PARAFAC," *SIAM Journal on Matrix Analysis and Applications*, 2008.

[78] P. Comon, X. Luciani, and A. L. F. De Almeida, "Tensor Decompositions, Alternating Least Squares and other Tales," *Journal of Chemometrics*, vol. 23, pp. 393–405, Aug. 2009.

[79] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[80] A. P. Liavas and N. D. Sidiropoulos, "Parallel algorithms for constrained tensor factorization via the alternating direction method of multipliers," *CoRR*, vol. abs/1409.2383, 2014.

[81] P. L. Combettes and J. C. Pesquet, *Proximal Splitting Methods in Signal Processing*, pp. 185–212, Springer New York, New York, NY, 2011.

[82] A. Franc, *Etude algébrique des multi-tableaux : apport de l'algèbre tensorielle*, Ph.D. thesis, University of Montpellier II, 1992.

[83] J.-P. Royer, P. Comon, and N. Thirion-Moreau, "Computing the nonnegative 3-way tensor factorization using tikhonov regularization," in *International Conference on Acoustic Speech and Signal Processing (ICASSP'2011)*, Prague, Czech Republic, May 2009, pp. 2732–2735.

[84] J.-P. Royer, N. Thirion-Moreau, and P. Comon, "Computing the polyadic decomposition of nonnegative third order tensors," *Signal Processing*, vol. 91, no. 9, pp. 2159 – 2171, 2011.

[85] J. Coloigner, A. Karfoul, L. Albera, and P. Comon, "Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors," 06 2014.

[86] J. Coloigner, L. Albera, A. Kachenoura, and L. Senhadji, "Canonical polyadic decomposition of semi-symmetric and semi-nonnegative three-way arrays by means of an exponential change of variable using line search and trust region strategies," 07 2012.

[87] J. Cohen, R. C. Farias, and P. Comon, "Fast decomposition of large nonnegative tensors," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 862–866, July 2015.

[88] M. A. Veganzones, J. E. Cohen, R. Cabral Farias, J. Chanussot, and P. Comon, "Nonnegative tensor cp decomposition of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2577–2588, May 2016.

[89] F. Roemer and M. Haardt, "A semi-algebraic framework for approximate cp decompositions via simultaneous matrix diagonalizations (secsi)," *Signal Processing*, vol. 93, no. 9, pp. 2722 – 2738, 2013.

[90] L. De Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM Journal on matrix Analysis and Applications*, vol. 28, no. 3, pp. 642–666, 2006.

[91] S. Hajipour, L. Albera, M. B. Shamsollahi, and I. Merlet, "Canonical polyadic decomposition of complex-valued multi-way arrays based on simultaneous Schur decomposition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 4178–4182.

[92] E. Moreau, "A generalization of joint-diagonalization criteria for source separation," *IEEE on Signal Processing*, vol. 49, no. 3, pp. 530–541, Mar. 2001.

[93] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, february 1997.

[94] G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor, "Joint Matrices Decompositions and Blind Source Separation. A Survey of Methods, Identification and Applications," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 34–43, May 2014.

[95] A. Mesloub, K. Abed-Meraim, and A. Belouchrani, "A new algorithm for complex non-orthogonal joint diagonalization based on shear and givens rotations," *IEEE Transactions on Signal Processing*, vol. 62, no. 8, pp. 1913–1925, April 2014.

[96] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Computation of the canonical decomposition by means of a simultaneous generalized schur decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, pp. 295–327, 2001.

[97] B. Afsari, "Sensitivity analysis for the problem of matrix joint diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, pp. 1148–1171, September 2008.

[98] M. Kleinsteuber and H. Shen, "Uniqueness analysis of non-unitary matrix joint diagonalization," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1786–1796, April 2013.

[99] B. Chabriel, J. Barrere, N. Thirion-Moreau, and E. Moreau, "Algebraic joint zero-diagonalization and blind sources separation," *IEEE Transactions On Signal Processing*, vol. 56, no. 3, pp. 980–989, 2008.

[100] H. Ghennioui, F. el Mostafa, N. Thirion-Moreau, A. Adib, and E. Moreau, "A nonunitary joint block diagonalization algorithm for blind separation of convolutive mixtures of sources," *IEEE Signal Processing Letters*, vol. 14, no. 11, pp. 860–863, November 2007.

[101] Y. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1982.

[102] A. Yeredor, "On using exact joint diagonalization for noniterative approximate joint diagonalization," *IEEE Signal Process. Lett.*, vol. 12, no. 9, pp. 645–648, 2005.

[103] G. Chabriel and J. Barrere, "A direct algorithm for nonorthogonal approximate joint diagonalization," *Trans. Sig. Proc.*, vol. 60, no. 1, pp. 39–47, jan 2012.

[104] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, Jul 2002.

[105] T. Trainini and E. Moreau, "A least squares algorithm for global joint decomposition of complex matrix sets," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2011, pp. 313–316.

[106] P. Tichavsky and A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Transactions On Signal Processing*, vol. 57, no. 3, pp. 878–891, March 2009.

[107] A. J. van der Veen, "Joint diagonalization via subspace fitting techniques," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, 2001, vol. 5, pp. 2773–2776 vol.5.

[108] L. Wang, L. Albera, L. Senhadji, and J. C. Pesquet, "An alternating direction method of multipliers for constrained joint diagonalization by congruence," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2015, pp. 197–200.

[109] B. Afsari, "Simple lu and qr based non-orthogonal matrix joint diagonalization," *Independent Component Analysis and Blind Signal Separation*, pp. 1–7, 2006.

[110] M. Sørensen, S. Icart, P. Comon, and L. Deneire, "Gradient based approximate joint diagonalization by orthogonal transforms," in *2008 16th European Signal Processing Conference*, Aug 2008, pp. 1–5.

[111] L. Wang, L. Albera, A. Kachenoura, H. Shu, and L. Senhadji, "Canonical polyadic decomposition of third-order semi-nonnegative semi-symmetric tensors using lu and qr matrix factorizations," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 150, Oct 2014.

[112] L. Wang, L. Albera, H. Z. Shu, and L. Senhadji, "A new jacobi-like nonnegative joint diagonalization by congruence," 09 2013.

[113] B. N. Flury, "Common principal components in k groups," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 892–898, 1984.

[114] D. T. Pham, "Joint approximate diagonalization of positive definite hermitian matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 22, no. 4, pp. 1136–1152, 2001.

[115] K. Todros and J. Tabrikian, "Qml-based joint diagonalization of positive-definite hermitian matrices," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4656–4673, Sept 2010.

[116] J-F Cardoso and Beate H Laheld, "Equivariant adaptive source separation," *IEEE Transactions on signal processing*, vol. 44, no. 12, pp. 3017–3030, 1996.

[117] Andreas Ziehe, Pavel Laskov, Guido Nolte, and Klaus-Robert MÃžller, "A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation," *Journal of Machine Learning Research*, vol. 5, no. Jul, pp. 777–800, 2004.

[118] X. F. Xu, D. Z. Feng, and W. X. Zheng, "A fast algorithm for nonunitary joint diagonalization and its application to blind source separation," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3457–3463, July 2011.

[119] T. Trainini and E. Moreau, "A coordinate descent algorithm for complex joint diagonalization under hermitian and transpose congruences," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 4974–4983, Oct 2014.

[120] X. Guo, S. Zhu, S. Miron, and D. Brie, "Approximate joint diagonalization by nonorthogonal nonparametric jacobi transformations," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 3774–3777.

[121] A. Souloumiac, "Nonorthogonal joint diagonalization by combining givens and hyperbolic rotations," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2222–2231, June 2009.

[122] 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, Dec 2014.

[123] G. H. Golub and C. F. Van Loan, *Matrix computations, second edition*, The Johns Hopkins University Press, Baltimore, MD, 1989.

[124] S. Banerjee and A. Roy, *Linear algebra and matrix analysis for statistics*, Chapman & Hall/CRC texts in statistical science. CRC Press, Hoboken, NJ, 2014.

[125] M. Sørensen, P. Comon, S. Icart, and L. Deneire, "Approximate tensor diagonalization by invertible transforms," in *2009 17th European Signal Processing Conference*, Aug 2009, pp. 500–504.

[126] V. Maurandi, E. Moreau, and C. De Luigi, "Jacobi like algorithm for non-orthogonal joint diagonalization of hermitian matrices," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2014)*, Florence, Italy, 2014.

[127] V. Maurandi, C. De Luigi, and E. Moreau, "Fast jacobi like algorithms for joint diagonalization of complex symmetric matrices," in *21st European Signal Processing Conference (EUSIPCO'2013)*, Marrakech, Marocco, 2013.

[128] 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 *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2014, pp. 1–6.

[129] A. J. Van der Veen, P. B. Ober, and E. F. Deprettere, "Azimuth and elevation computation in high resolution doa estimation," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1828–1832, July 1992.

[130] A. N. Lemma, A. J. Van der Veen, and E. F. Deprettere, "Analysis of joint angle-frequency estimation using ESPRIT," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1264–1283, May 2003.

[131] M. Haardt and J.A. Nossek, "Simultaneous schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems," *IEEE Transactions on Signal Processing*, vol. 46, no. 1, pp. 161–169, January 1998.

[132] E. R. Balda, S. A. Cheema, A. Weiss, A. Yeredor, and M. Haardt, "Perturbation analysis of joint eigenvalue decomposition algorithms," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 3101–3105.

[133] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices," in *ICASSP 2006, 2006 IEEE International Conference on Acoustics Speech and Signal Processing*, May 2006, vol. 4, pp. 1137–1140.

[134] R. Iferroudjene, K. Abed-Meraim, and A. Belouchrani, "A new jacobi-like method for joint diagonalization of arbitrary non-defective matrices," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 363–373, 2009.

[135] X. L. Li and X. D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solution," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1803–1814, May 2007.

[136] B. Li, *"Generalizations of diagonal dominance in matrix theory"*, Ph.D. thesis, University of Regina, 1997.

[137] D. Choudhury and R.A. Horn, "A complex orthogonal-symmetric analog of the polar decomposition," *SIAM J. Algebraic Discrete Methods*, vol. 8, pp. 219–225, 1987.

[138] D. Nion and L. De Lathauwer, "A Block Component Model based Blind DS-CDMA Receiver," *IEEE Trans. Signal Proc.*, vol. 56, no. 11, pp. 5567–5579, 2008.

[139] A. de Almeida, G. Favier, and J.C. Mota, "Space-time spreading mimo-cdma downlink systems using constrained tensor modeling," *Signal Processing*, vol. 88, no. 10, pp. 2403 – 2416, 2008.

[140] M. da Costa, A. de Almeida, G. Favier, and J. Romano, "Tensor Coding for CDMA-MIMO Wireless Communication Systems," in *19th European Signal Processing Conference (EUSIPCO)*, Barcelone, Spain, Aug. 29 - Sept. 2 2011, pp. 101–105.

[141] A. de Almeida, A. Kibangou, S. Miron, and D. Araùjo, "Joint data and connection topology recovery in collaborative wireless sensor networks," in *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, May 2013, pp. 5303–5307.

[142] D. Maurel, "Diagramme de jablonski," .

[143] R. Bro, Å. Rinnan, and N. M. Faber, "Standard error of prediction for multilinear pls - 2. practical implementation in fluorescence spectroscopy," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 1, pp. 69–76, 2005.

[144] Å. Rinnan, *Application of PARAFAC on Spectral Data*, Ph.D. thesis, Royal Veterinary and Agricultural University, 2004.

[145] R. G Zepp, W. Sheldon, and M. A. Moran, "Dissolved organic fluorophores in southeastern us coastal waters : Correction method for eliminating rayleigh and raman scattering peaks in excitation-emission matrices," vol. 89, pp. 15–36, 10 2004.

Résumé

Cette thèse présente de nouveaux algorithmes de diagonalisation conjointe par similitude. Ces algorithmes permettent, entre autres, de résoudre le problème de décomposition canonique polyadique de tenseurs. Cette décomposition est particulièrement utilisée dans les problèmes de séparation de sources. L'utilisation de la diagonalisation conjointe par similitude permet de palier certains problèmes dont les autres types de méthode de décomposition canonique polyadique souffrent, tels que le taux de convergence, la sensibilité à la surestimation du nombre de facteurs et la sensibilité aux facteurs corrélés. Les algorithmes de diagonalisation conjointe par similitude traitant des données complexes donnent soit de bons résultats lorsque le niveau de bruit est faible, soit sont plus robustes au bruit mais ont un coût calcul élevé. Nous proposons donc en premier lieu des algorithmes de diagonalisation conjointe par similitude traitant les données réelles et complexes de la même manière. Par ailleurs, dans plusieurs applications, les matrices facteurs de la décomposition canonique polyadique contiennent des éléments exclusivement non-négatifs. Prendre en compte cette contrainte de non-négativité permet de rendre les algorithmes de décomposition canonique polyadique plus robustes à la surestimation du nombre de facteurs ou lorsque ces derniers ont un haut degré de corrélation. Nous proposons donc aussi des algorithmes de diagonalisation conjointe par similitude exploitant cette contrainte. Les simulations numériques proposées montrent que le premier type d'algorithmes développés améliore l'estimation des paramètres inconnus et diminue le coût de calcul. Les simulations numériques montrent aussi que les algorithmes avec contrainte de non-négativité améliorent l'estimation des matrices facteurs lorsque leurs colonnes ont un haut degré de corrélation. Enfin, nos résultats sont validés à travers deux applications de séparation de sources en télécommunications numériques et en spectroscopie de fluorescence.

Abstract

This thesis introduces new joint eigenvalue decomposition algorithms. These algorithms allow amongst others to solve the canonical polyadic decomposition problem. This decomposition is widely used for blind source separation. Using the joint eigenvalue decomposition to solve the canonical polyadic decomposition problem allows to avoid some problems whose the others canonical polyadic decomposition algorithms generally suffer, such as the convergence rate, the overfactoring sensibility and the correlated factors sensibility. The joint eigenvalue decomposition algorithms dealing with complex data give either good results when the noise power is low, or they are robust to the noise power but have a high numerical cost. Therefore, we first propose algorithms equally dealing with real and complex. Moreover, in some applications, factor matrices of the canonical polyadic decomposition contain only nonnegative values. Taking this constraint into account makes the algorithms more robust to the overfactoring and to the correlated factors. Therefore, we also offer joint eigenvalue decomposition algorithms taking advantage of this nonnegativity constraint. Suggested numerical simulations show that the first developed algorithms improve the estimation accuracy and reduce the numerical cost in the case of complex data. Our numerical simulations also highlight the fact that our nonnegative joint eigenvalue decomposition algorithms improve the factor matrices estimation when their columns have a high correlation degree. Eventually, we successfully applied our algorithms to two blind source separation problems : one concerning numerical telecommunications and the other concerning fluorescence spectroscopy.