

# Schéma parallélisé-randomisé de la décomposition canonique polyadique pour la localisation de sources de courant intracérébral

Abdelhak BOUDEHANE<sup>1</sup>, Laurent ALBERA<sup>2</sup>, Arthur TENENHAUS<sup>1</sup>, Laurent LE BRUSQUET<sup>1</sup>, Rémy BOYER<sup>3</sup>

<sup>1</sup>Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France.

<sup>2</sup>Université Rennes 1, Inserm, LTSI - UMR 1099, F-35000 Rennes, France.

<sup>3</sup>Centre de Recherche en Informatique, Signal et Automatique de Lille, Univ. Lille 1, Villeneuve-d'Ascq, France.

abdelhak.boudehane@gmail.com, laurent.albera@univ-rennes1.fr

arthur.tenenhaus@centralesupelec.fr, Laurent.Lebusquet@centralesupelec.fr, remy.boyer@univ-lille.fr

**Résumé** – Les avantages en terme d’interprétabilité et d’unicité du modèle Canonique Polyadique (CP) sont largement exploités dans de nombreux contextes applicatifs, tels que la localisation de sources de courant intracérébral. Toutefois, la complexité et la taille mémoire nécessaire pour le calcul de ces décompositions constituent un véritable challenge pour les tenseurs de grandes dimensions et d’ordre élevé. Dans cet article, nous présentons un schéma parallélisé et randomisé qui permet de réduire la complexité et la taille mémoire requise, tout en conservant la qualité de localisation.

**Abstract** – The advantages in terms of uniqueness and interpretability of the Canonical Polyadic (CP) model are widely exploited in various applications, such as the brain current source localization. However, the complexity and memory requirements for the computation of these decompositions remain a real challenge for high-order large-scale tensors. In this article, we present a parallelized-randomised scheme, allowing for the reduction of computational complexity and memory requirements while maintaining the localization quality.

## 1 Introduction

Les modèles tensoriels présentent de nombreux avantages pour le traitement de données et de signaux multidimensionnels. L’unicité de la décomposition du modèle CP (Canonique Polyadique) [1] facilite l’interprétation physique. En effet, cette décomposition est unique à des ambiguïtés triviales près [2]. De plus, ce modèle permet la prise en considération du caractère multidimensionnel des données brutes ou transformées, comme peuvent l’être les signaux EEG (ElectroEncephaloGraphy) après utilisation d’une transformée temps-échelle. Ceci n’est pas possible par les modèles matriciels usuels que l’on peut trouver dans la littérature [3, 4]. Cet avantage permet une meilleure séparation de sources et, donc, une meilleure localisation de ces dernières dans le contexte EEG, où l’objectif est de localiser les sources de courant intracérébral à partir de l’activité électrique de scalp. Toutefois, il convient de dire que, dans le cas des tenseurs d’ordre élevé et de grandes dimensions, la complexité de calcul et le stockage ( $N^Q$  où  $Q$  est l’ordre du tenseur et  $N$  est la taille du tenseur cubique) sont des goulots d’étranglement technologiques qui empêchent de traiter les données multidimensionnelles [5]. En effet, le nombre d’éléments du tenseur augmente exponentiellement en fonction de l’ordre, ce qui résulte en un accroissement exponentielle de la complexité de calcul et de la capacité de mémoire requise. Le problème est identifié par la communauté sous le terme générique de “fléau de la dimension”.

Ainsi, les algorithmes proposés dans la littérature, tels que ALS

(Alternating Least Squares) qui est un algorithme itératif, ou encore des approches basées sur la descente de gradient [6], font face à des problèmes de convergence lorsqu’il s’agit de tenseurs d’ordre élevé. Le schéma JIRAFE (Joint dImentionalitY Reduction And Factor rEtrieval) propose une solution à ce problème [7, 8]. Cependant, ce schéma basé sur le modèle TT (Tensor-Train) et dédié au traitement des tenseurs d’ordre élevé, rencontre aussi une limitation dans le cas des tenseurs de grande taille. [9] tels qu’en EEG. En effet, JIRAFE est composé de deux étapes essentielles : la réduction de la dimension et l’estimation des facteurs. Dans la première étape, JIRAFE utilise la représentation TT (Tensor-Train) [10] pour “casser” le tenseur CP d’ordre élevé en un train de tenseurs d’ordre 3 appelés les TT-coeurs. Cette étape consiste à appliquer une série de SVDs sur les différents “reshapings” du tenseur. Dans le cas des tenseurs de grandes dimensions, cette étape devient très coûteuse en terme de complexité et d’espace mémoire.

Dans cet article, nous proposons un schéma parallélisé et randomisé de JIRAFE, afin de diminuer la complexité de calcul (et donc le temps de calcul), ainsi que la taille de la mémoire requise. Les performances de l’approche proposée sont évaluées sur signaux simulés de manière réaliste dans le contexte de l’EEG et de l’épilepsie.

## 2 Tenseurs massifs

La décomposition CP du tenseur  $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_Q}$  de rang  $R$  et d’ordre  $Q$  s’écrit comme  $\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \dots \times_Q \mathbf{P}_Q$ ,

où  $\mathbf{P}_q \in \mathbb{R}^{N_q \times R}$  sont les matrices-facteurs à estimer. Donnons quelques notations utilisées dans la suite de l'article : les vecteurs, matrices et tenseurs sont représentés, respectivement, par  $\mathbf{x}$ ,  $\mathbf{X}$  et  $\mathcal{X}$ . Les symboles  $(\cdot)^T$ ,  $(\cdot)^{-T}$  et  $(\cdot)^\dagger$  indiquent, respectivement, la transposée, l'inverse transposée et la pseudo-inverse d'une matrice. La norme de Frobenius est définie par  $\|\cdot\|_F$ . Le produit de Khatri-Rao est noté par  $\odot$ . Le tenseur identité  $\mathcal{I}_{Q,R}$ , est un tenseur  $(R \times R \times \dots \times R)$  d'ordre  $Q$  avec des uns sur la diagonale et des zéros ailleurs. La matrice  $\mathbf{X}^{(k)}$  ( $N_k \times N_1 \dots N_{k-1} N_{k+1} \dots N_Q$ ) est le  $k$ ème dépliement du tenseur  $\mathcal{X}$ .

## 2.1 Schéma JIRAFE

JIRAFE exploite l'équivalence entre le modèle CP et le modèle TT (Tensor-Train) pour casser le tenseur d'ordre élevé en un train de tenseurs d'ordre 3 appelés les TT-coeurs. La décomposition TT d'un tenseur CP est donnée par

$$\mathcal{X} = \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \dots \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 \mathbf{G}_Q,$$

où  $\mathcal{G}_q = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T}$  (pour  $2 < q < Q-1$ ),  $\mathbf{G}_1 = \mathbf{P}_1 \mathbf{M}_1^{-1}$  et  $\mathbf{G}_Q = \mathbf{M}_{Q-1} \mathbf{P}_Q^T$  sont les TT-coeurs de tailles respectives  $R \times N_p \times R$ ,  $N_1 \times R$  et  $N_Q \times R$ . Les matrices  $\mathbf{M}_q$ , de taille  $R \times R$ , sont les matrices de passage pour  $1 \leq q \leq Q-1$ . Ainsi, le problème de décomposition tensorielle d'ordre élevé est réduit à une série de décompositions CP couplées d'ordre 3, où les facteurs peuvent être estimés en utilisant l'algorithme Tri-ALS (ALS d'ordre 3) et Bi-ALS (ALS d'ordre 3 avec un facteur pré-estimé).

## 2.2 Limitation

Le schéma de JIRAFE, présenté dans la section précédente, a montré de bonnes performances en terme de réduction de calcul. La TT-SVD [10] (algorithme utilisé pour estimer les TT-coeurs) représente la complexité dominante de JIRAFE. Cet étape consiste à appliquer une série de SVDs sur les différents remodelages du tenseur, pour extraire les différents modes. Dans le cas des tenseurs de grande taille, cette étape devient coûteuse en terme de complexité et d'espace mémoire. La complexité de la TT-SVD pour un tenseur cubique de rang  $R$ , d'ordre 4 et de taille  $N \times N \times N \times N$  est donnée par

$$k(\text{TT-SVD}) = O(RN^Q) + O(R^2 N^{Q-1}) + \dots + O(R^2 N^2). \quad (1)$$

L'équation (1) montre la dépendance de la complexité de la TT-SVD aux dimensions  $N$  du tenseur. Par conséquent, la complexité de calcul de JIRAFE dans le cas des tenseurs de grandes dimensions et d'ordre élevé devient intraitable. De plus, la complexité du Tri-ALS utilisé pour estimer un facteur  $\mathbf{P}_q$  à partir du TT-coeur  $\mathcal{G}_q$ , est donnée par  $O(3R^3 N_q)$ . Ainsi, cette complexité augmente linéairement avec les dimensions.

## 3 Stratégie de parallélisation

Dans cette section, nous proposons des solutions afin de réduire la complexité de calcul et l'espace mémoire requis par JIRAFE, tout en maintenant le même niveau de précision.

## 3.1 Diviser pour régner

La stratégie split and merge [11] (diviser et fusionner) consiste à "diviser" la matrice de grande taille en sous-matrices de taille raisonnable. La décomposition SVD des sous-matrices se fait séparément, *i.e.*, en parallèle ou en séquentiel. Ensuite, l'étape "fusionner" permet de remonter à la SVD de la matrice de grande taille à partir des SVDs des sous-matrices. Dans notre contexte, nous utilisons cette stratégie pour paralléliser les étapes SVDs de la TT-SVD afin de réduire le temps de calcul. L'algorithme 1 décrit les étapes de cette stratégie appliquée sur le 1er remodelage du tenseur. Il est à noter que split

### Algorithme 1 Split and merge parallélisée

---

**Entrées :** Déploiements du tenseur  $\mathcal{X}$  et le rang  $R$   
**Sorties :** Les matrices  $\mathbf{U}_x$ ,  $\mathbf{S}_x$  et  $\mathbf{V}_x$

- 1: Diviser  $\mathbf{X}^{(1)}$  en sous matrices  $\mathbf{X}^{(1)} = [\mathbf{X}_1^{(1)T} \mathbf{X}_2^{(1)T} \dots \mathbf{X}_J^{(1)T}]^T$
- 2: **parfor**  $j \in \{1, \dots, J\}$   
 Appliquer la SVD sur  $\mathbf{X}_j^{(1)T}$  en parallèle  $[\tilde{\mathbf{U}}_j, \mathbf{S}_j, \mathbf{V}_j^T] = \mathbf{X}_j^{(1)T}$   
**end parfor**
- 3: Former les nouvelles matrices  $\mathbf{U} = \text{diag}(\tilde{\mathbf{U}}_1, \dots, \tilde{\mathbf{U}}_J)$  et  $\mathbf{Y} = [\mathbf{S}_1 \mathbf{V}_1^T \dots \mathbf{S}_J \mathbf{V}_J^T]^T$
- 4: Appliquer la SVD sur  $\mathbf{Y}$   $\mathbf{Y} = \mathbf{U}_y \mathbf{S}_y \mathbf{V}_y^T$
- 5: La décomposition SVD de  $\mathbf{X}^{(1)}$   $\mathbf{U}_x = \tilde{\mathbf{U}} \mathbf{U}_y$ ,  $\mathbf{S}_x = \mathbf{S}_y$ ,  $\mathbf{V}_x = \mathbf{V}_y$ .

---

and merge permet la parallélisation des SVDs et, donc, la distribution de la complexité sur les calculateurs et la réduction du temps de calcul. Afin de réduire la complexité de calcul nécessaire pour effectuer les SVDs sur chaque calculateur, nous remplaçons les SVDs des étapes 2 et 4 de l'algorithme 1 par la SVD randomisée (RSVD) [12]. En utilisant un nombre suffisant d'itérations de puissance [12, algorithm 4.4], nous arrivons à réduire la complexité des étapes SVD tout en gardant le même niveau de précision.

## 3.2 Parallélisation du Tri-ALS

Dans le contexte de tenseurs de grandes dimensions, les dimensions  $N_p$  sont grandes. Cela veut dire qu'une dimension de chaque TT-coeur est grande alors que les 2 autres sont réduites. Afin de réduire le temps écoulé par le Tri-ALS pour estimer les facteurs, nous proposons une idée inspirée du Grid-PARAFAC [13], qui consiste à diviser le TT-coeur, selon la plus grande dimension, en  $L$  petits sous-tenseurs (voir Fig.1). Ces sous-tenseurs, dont la décomposition CP s'écrit comme

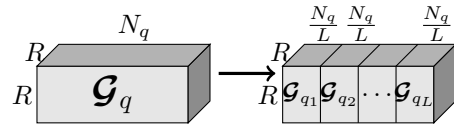


FIGURE 1 – Diviser un TT-core en  $L$  sous-tenseurs

$\mathcal{G}_{q_l} = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_{q_l} \times_3 \mathbf{M}_q^{-T}$  (avec  $\mathbf{P}_{q_l}$  le  $l$ -ème sous facteur), partagent les mêmes matrices de passage. De ce fait, nous n'avons besoin que d'un seul sous-tenseur  $\mathcal{G}_{q_l}$  pour estimer les matrices de passage, ainsi que le sous-facteur  $\mathbf{P}_{q_l}$  correspondant. Les sous-facteurs restants, dont la concaténation donne le facteur  $\mathbf{P}_q$ , peuvent être calculés exactement comme  $\mathbf{P}_{q_l} = \mathbf{G}_{q_l}^{(2)} [(\mathbf{M}_q^{-T} \odot \mathbf{M}_{q-1}^T)^T]^\dagger$ . Le choix

du sous-tenseur  $\mathcal{G}_{q_l}$  dépend de l'application. Dans notre cas, le critère permettant de choisir le sous-tenseur est la norme de Frobeinus, *i.e.*, nous estimons les matrices de passage à partir du sous-tenseur ayant la plus grande norme de Frobeinus. Ce choix se justifie par la nature du signal de sources (impulsion) où le pic correspond à la plus grande valeur du signal comparé à l'activité du fond. Par conséquent, choisir le sous-tenseur  $\mathcal{G}_{2_l}$  ayant la plus grande norme revient à choisir la partie du signal la moins affectée par l'activité du fond.  $l'$  est donc choisi tel que  $l' = \operatorname{argmax}_l \|\mathcal{G}_{q_l}\|_F$ . Les différentes étapes du schéma R-P-JIRAFE (Randomized-Parallelized JIRAFE), proposé dans cet article, sont résumés dans l'algorithme 2. L'étape 1 consiste

---

### Algorithme 2 R-P-JIRAFE

---

**Entrées :** Tenseur  $\mathcal{X}$  et son rang  $R$   
**Sorties :** Les matrices facteurs  $\mathbf{P}_q$   
1: Estimer les TT-coeurs  $[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q] = \text{P-TT-SVD}(\mathcal{X}, R)$   
2: **for**  $q \in \{2, \dots, Q-1\}$  Choisir  $l'$  et estimer les matrices  $\mathbf{M}_q$   
 $[\mathbf{M}_{q-1}, \mathbf{P}_{q_l}, \mathbf{M}_q] = \text{Tri-ALS}(\mathcal{G}_{q_l})$   
**for**  $l \neq l'$  Estimer les sous-facteurs restants  
 $\mathbf{P}_{q_l} = \mathbf{G}_{q_l}^{(2)} [(\mathbf{M}_q^{-T} \odot \mathbf{M}_{q-1})^T]^\dagger$   
**end for**  
**end for**  
3: Déterminer  $\mathbf{P}_1$  et  $\mathbf{P}_Q$  :  $\mathbf{P}_1 = \mathbf{G}_1 \mathbf{M}_1$  et  $\mathbf{P}_Q = \mathbf{G}_Q^T \mathbf{M}_{Q-1}^{-T}$

---

à estimer les TT-coeurs en utilisant la TT-SVD parallélisée (P-TT-SVD), *i.e.*, en utilisant la stratégie split and merge parallèle et en remplaçant les SVDs par les RSVDs comme décrit dans la section précédente. À l'étape 2, nous choisissons le  $l'$ -ème sous-tenseur de chaque TT-coeur pour estimer les matrices de passage et, donc, déterminer les sous-facteurs de manière exacte. Enfin, nous calculons le premier et le dernier facteurs.

## 4 Simulations

Dans cette partie, nous évaluons les performances de notre algorithme sur un problème de localisation de sources de courant intracérébral épileptogènes [14] à partir de signaux EEG réalistes [15]. Le problème de localisation de sources est un problème inverse mal-posé du fait d'un nombre d'électrodes (256) inférieur au nombre de dipôles corticaux (10000). L'objectif est de localiser les sources épileptogènes en utilisant les données EEG et la matrice "lead-field". Cette matrice modélise le transfert entre l'activité électrique corticale et l'activité mesurée par les électrodes EEG. La matrice spatio-temporelle des signaux EEG s'écrit comme  $\mathbf{Z} = \mathbf{G}\mathbf{S}$ , où  $\mathbf{G}$  représente la matrice lead-field et  $\mathbf{S}$  représente les sources de courant. Afin d'améliorer la séparation des sources, nous appliquons une transformée en ondelettes sur la matrice  $\mathbf{Z}$ , en utilisant l'ondelette de Morlet. Le résultat est un tenseur  $\mathcal{Y}$  d'ordre 3 dont les directions représentent l'espace, le temps et les échelles. On simule, par ailleurs, la répétition des biomarqueurs EEG, pointes épileptiques, segments temporels utilisés pour la localisation des générateurs épileptogènes. Par conséquent, on obtient un tenseur  $\mathcal{X}$  d'ordre 4 de taille  $N_1 \times N_2 \times N_3 \times N_4$ . Pour approcher la pratique, on se place dans un scénario où 3 patchs corticaux épileptogènes sont présents : deux patchs proches l'un

de l'autre, avec une forte corrélations entre leurs activités, et un patch distant avec une activité moins corrélée avec les deux autres patchs. On ajoute un bruit blanc Gaussien pour simuler le bruit généré par les instruments de mesure et un bruit modélisant l'activité de fond [15]. Les paramètres de la matrice EEG sont  $N_1 = 91$  électrodes et  $N_2 = 200$  échantillons temporels avec une fréquence d'échantillonnage  $f_s = 256\text{Hz}$ . Pour former notre tenseur, nous appliquons une transformée en ondelette avec  $N_3 = 60$  échelles avec un nombre de répétition des pointes épileptiques  $N_4 = 50$ . Le rang du tenseur est égal au nombre de patchs présents ( $R = 3$ ). Afin d'évaluer les performances de notre algorithme, nous utilisons comme critères le DLE (Distance of Localization Error) [16], ainsi que le temps écoulé par notre algorithme, sur un ordinateur ayant les caractéristiques suivantes : *Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz (8 CPUs), 2.1GHz*. Nous comparons les résultats de simulation de notre algorithme à ceux de la version classique de JIRAFE et au NLS (Nonlinear Least Squares) d'ordre 4, proposé par la boîte à outils "TensorLab" [17]. Il est à noter que les algorithmes mentionnés précédemment sont appliqués au tenseur  $\mathcal{X}$  d'ordre 4. Nous comparons aussi ces 3 algorithmes au NLS d'ordre 3 appliqué au tenseur  $\mathcal{Y}$  (d'ordre 3) afin de visualiser l'impact de l'ajout de la dimensions répétition sur la localisation. Les résultats de simulation sont moyennés sur 100 réalisations de Monte Carlo de bruit/activité cérébrale pour chaque valeur de SNR. Dans ce contexte de localisation de sources, le facteur qui nous intéresse est le 1er facteur (la dimension spatiale). Les dipôles épileptiques sont identifiés en résolvant un problème de Fused LASSO pour chacune des 3 colonnes du 1er facteur en appliquant l'algorithme Sissy (Source Imaging based on Structured Sparsity) [14]. La figure 2 représente la reconstruction des patchs à la surface du cerveau par les différents algorithmes comparés à la vérité terrain, où l'on remarque que les différents algorithmes arrivent à localiser les patchs ainsi que leurs extensions. Afin d'évaluer les performances des algorithmes de façon précise, nous traçons les boîtes à moustaches qui représentent le DLE de chacune des méthodes (Fig. 3). Nous remarquons, pour toutes les valeurs du SNR, que l'erreur DLE des méthodes appliquées au tenseur  $\mathcal{X}$  d'ordre 4 est inférieure à l'erreur du NLS d'ordre 3 (NLS 3D) appliqué au tenseur  $\mathcal{Y}$ . Ceci atteste que l'utilisation de la 4ème dimension (répétition) donne une meilleure précision à la localisation. D'autre part, les performances, en terme de DLE, de R-P-JIRAFE, JIRAFE et le NLS 4D sont comparables, avec une légère différence en faveur du NLS 4D suivi par R-P-JIRAFE et JIRAFE. La légère différence entre R-P-JIRAFE et JIRAFE provient du fait que nous utilisons la partie du TT-coeur où la norme est maximale (la moins affectée par le bruit) pour estimer les facteurs. Le tableau 1 représente le temps de calcul écoulé par les différents algorithmes, où nous pouvons déduire que R-P-JIRAFE est le plus rapide parmi tous les algorithmes utilisés. En effet, R-P-JIRAFE est 10 fois plus rapide que la version classique de JIRAFE et 200 fois plus rapide que le NLS 4D. De plus, R-P-JIRAFE est 5 fois plus rapide que le NLS 3D appliqué à un tenseur d'ordre 3. Nous pou-

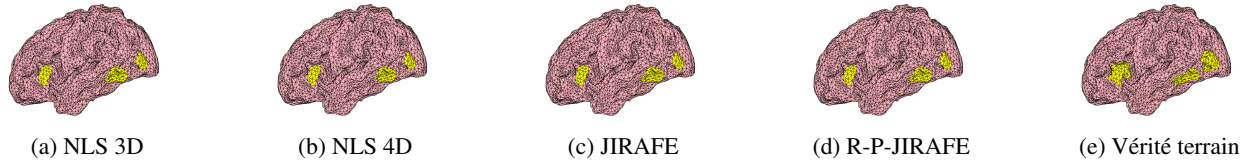


FIGURE 2 – Exemple de localisation des patches épileptiques à la surface du cerveau pour un SNR = 10 dB

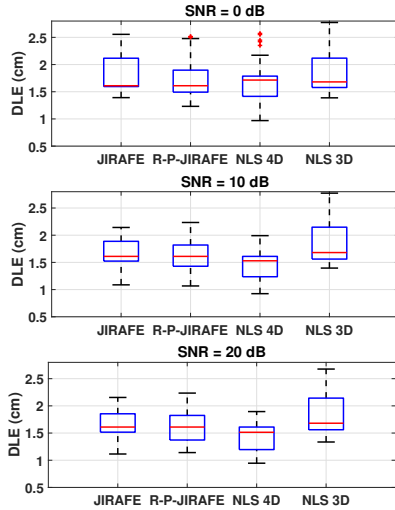


FIGURE 3 – Boîtes à moustaches représentant le DLE (cm)

TABLE 1 – Temps de calcul pour chaque algorithme (s)

SNR (dB)	Temps de calcul (s)			
	JIRAFE	R-P-JIRAFE	NLS 4D	NLS 3D
0	2.13	0.22	44.20	1.13
10	1.94	0.21	44.12	1.05
20	1.88	0.19	44.69	1.02

vons donc conclure que R-P-JIRAFE offre le meilleur compromis entre précision de localisation et rapidité de calcul parmi les algorithmes testés dans ce papier.

## 5 Conclusion

L'utilisation des modèles tensoriels présente de nombreux avantages dans le contexte de la localisation de sources de courant cérébral à partir des signaux EEG. Cependant, la complexité de calcul et le coût de stockage demeurent des éléments fortement limitatifs. Dans cet article, nous présentons un schéma de type JIRAFE parallélisé-randomisé, qui permet de réduire le temps de calcul et la mémoire requise tout en gardant un niveau acceptable d'erreur de localisation. Les performances de notre approche ont été évaluées sur des signaux EEG de manière réaliste.

## Références

- [1] R. A. Harshman, "Foundations of the PARAFAC procedure : Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.
- [2] 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.
- [3] H. Becker, L. Albera, P. Comon, M. Haardt, G. Birot, F. Wendling, M. Gavaret, C. Bénar, and I. Merlet, "EEG extended source localization : Tensor-based vs. conventional methods," *NeuroImage*, vol. 96, pp. 143 – 157, 2014.
- [4] H. Becker, L. Albera, P. Comon, R. Gribonval, F. Wendling, and I. Merlet, "Brain-source imaging : From sparse to tensor models," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 100–112, 2015.
- [5] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic, "Tensor networks for dimensionality reduction and large-scale optimization : Part 1 low-rank tensor decompositions," *Foundations and Trends® in Machine Learning*, vol. 9, no. 4-5, p. 249–429, 2016.
- [6] P. Comon, X. Luciani, and A. L. F. De Almeida, "Tensor decompositions, alternating least squares and other tales," *Journal of Chemometrics : A Journal of the Chemometrics Society*, vol. 23, no. 7-8, pp. 393–405, 2009.
- [7] Y. Znyied, R. Boyer, A. L. F. de Almeida, and G. Favier, "High-order CPD estimation with dimensionality reduction using a tensor train model," *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2613–2617, 2018.
- [8] Y. Znyied, R. Boyer, A. L. de Almeida, and G. Favier, "Multidimensional harmonic retrieval based on Vandermonde tensor train," *Signal Processing*, vol. 163, pp. 75 – 86, 2019.
- [9] C. I. Kanatsoulis and N. D. Sidiropoulos, "Large-scale Canonical Polyadic decomposition via regular tensor sampling," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.
- [10] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [11] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 1042–1049, 2000.
- [12] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness : Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, pp. 217–288, 2011.
- [13] A. H. Phan and A. Cichocki, "PARAFAC algorithms for large-scale problems," *Neurocomputing*, vol. 74, no. 11, pp. 1970 – 1984, 2011.
- [14] H. Becker, L. Albera, P. Comon, J.-C. Nunes, R. Gribonval, J. Fleureau, P. Guillotel, and I. Merlet, "SISSY : An efficient and automatic algorithm for the analysis of EEG sources based on structured sparsity," *NeuroImage*, vol. 157, pp. 157 – 172, 2017.
- [15] D. Cosandier-Riméle, J.-M. Badier, P. Chauvel, and F. Wendling, "A physiologically plausible spatio-temporal model for EEG signals recorded with intracerebral electrodes in human partial epilepsy," *IEEE transactions on bio-medical engineering*, vol. 54, pp. 380–8, 04 2007.
- [16] J. Cho, S. Hong, Y. Jung, H. Kang, H. Kim, M. Suh, K. Jung, and C. Im, "Evaluation of algorithms for intracranial EEG (iEEG) source imaging of extended sources : Feasibility of using iEEG source imaging for localizing epileptogenic zones in secondary generalized epilepsy," *Brain Topography*, vol. 24, no. 2, pp. 91–104, Jun. 2011.
- [17] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab 3.0," 2016. [Online]. Available : <https://www.tensorlab.net>