



**HAL**  
open science

# Algorithmes de décomposition canonique polyadique en ligne de tenseurs de fluorescence : Application à la détection et au suivi des contaminations biologiques et organiques en milieu aquatique

Isaac Wilfried Sanou

## ► To cite this version:

Isaac Wilfried Sanou. Algorithmes de décomposition canonique polyadique en ligne de tenseurs de fluorescence : Application à la détection et au suivi des contaminations biologiques et organiques en milieu aquatique. Automatique / Robotique. Université de Toulon, 2022. Français. NNT : 2022TOUL0004 . tel-04783490

**HAL Id: tel-04783490**

**<https://theses.hal.science/tel-04783490v1>**

Submitted on 14 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Toulon

**ECOLE DOCTORALE 548**  
**MER ET SCIENCES**

Laboratoire d'Informatique et Systèmes (LIS), UMR CNRS 7020  
Équipe Signal et Image (SIIM)  
Mediterranean Institute of Oceanologie (MIO), CNRS/INSU, IRD, MIO UMR 110  
Chimie des Environnements Marins (CEM)

**THÈSE**

pour l'obtention du grade de

**Docteur en Sciences**

de l'Université de Toulon

Mention : Automatique, signal, productique, robotique

présentée par

Isaac Wilfried SANOU

**Algorithmes de décomposition canonique polyadique en ligne de tenseurs de fluorescence :  
Application à la détection et au suivi des contaminations biologiques et organiques en milieux  
aquatiques**

Thèse dirigée par Stéphane MOUNIER et encadrée par Xavier LUCIANI & Roland REDON

**Jury :**

MME. NADÈGE THIRION	Professeur à l'Université de Toulon, LIS	Présidente
MME. MARIANNE CLAUSEL	Professeur à l'Université de Lorraine, IECL	Rapporteuse
M. LAURENT ALBERA	Professeur à l'Université de Rennes 1, LTSI	Rapporteur
MME. FLORENCE VOUVE	Maître de conférence à l'Université de Perpignan, BAE-LBBM	Examinatrice
M. PIERRE COMON	Directeur de Recherche CNRS, GIPSA-lab	Examineur
M. STÉPHANE MOUNIER	Maître de conférence HDR à l'Université de Toulon, MIO	Directeur de thèse
M. XAVIER LUCIANI	Maître de conférence à l'Université de Toulon, LIS	Encadrant de thèse
M. ROLAND REDON	Maître de conférence à Aix-Marseille Université, MIO	Encadrant de thèse



# Remerciements

Je tiens, tout d'abord, à remercier l'ensemble des membres de mon jury. Je remercie Marianne Clausel et Laurent Albera pour avoir accepté de rapporter ma thèse ainsi que pour leurs remarques constructives. Je remercie aussi mes examinateurs, Pierre Comon, Nadège Thirion-Moreau et Florence Vouvé pour la pertinence de leurs questions et les différentes suggestions. Je remercie Pierre Comon et Florence Vouvé pour le suivi de cette thèse, cela fut vraiment très enrichissant.

J'aimerais montrer ma gratitude à mes trois superviseurs : Stéphane Mounier, Xavier Luciani et Roland Redon pour leur temps, pour leurs bons conseils, leur motivation et enfin leurs connaissances. Cette thèse n'aurait pas été possible sans leur soutien, leur encouragement et la patience dont ils ont fait preuve pendant ces années ainsi que les suggestions intéressantes qui m'ont aidé à établir une base de travail cohérente, solide et consistante. J'aimerais aussi montrer mon profond respect à leur famille respective avec qui nous avons passé des moments agréables en dehors du travail : je pense à Sophie Mounier, Nawal Redon, et Annamaria Monitillo. Je remercie l'ensemble des partenaires opérationnels et financiers du projet 3D zip et à Bioceanor.

J'adresse un grand merci aux membres du MIO et du LIS pour m'avoir chaleureusement accueilli au sein de leur équipe respective tout en faisant de telle sorte que je me sente le mieux possible tout au long de cette thèse. Je tiens à remercier tout spécialement Amonda, Christian, Marie-Christine, et Houssam pour leur disponibilité. Je remercie Eric Busvelle pour sa présence à ma soutenance et sa disponibilité. J'adresse aussi un grand merci aux membres du LABOMAR de l'université de Fortaleza au Brésil pour leur accueil lors de mon stage effectué au sein de leur laboratoire.

Je remercie toutes les personnes qui m'ont aidé de près ou de loin dans cette aventure, à mes amis qui m'ont aidé par leurs conseils et aussi pour les moments de détente. Enfin, je remercie toute ma famille pour m'avoir soutenu lors de ce voyage. Merci à mes parents pour avoir toujours cru en moi, à mes oncles en particulier tonton Serges pour tout son soutien. Merci à mes sœurs Christelle et Armelle et à mon petit frère Benjamin pour leur soutien.



# Résumé

Dans cette thèse nous nous intéressons au problème des décompositions polyadiques canoniques (CPD/PARAFAC) de tenseurs d'ordre 3 sous différentes contraintes comme le temps réel. La décomposition polyadique canonique est utilisée dans de nombreux domaines tels que la chimie, la biologie et la médecine. Les données issues de ces champs peuvent être dynamiques ce qui conduit à utiliser une décomposition en temps réel ou « en ligne ». Même s'il existe une variété d'algorithmes de décomposition de tenseur en ligne, l'hypothèse principale de tous ces algorithmes est que le rang ou le nombre de composants de la décomposition est connu et/ou ne varie pas dans le temps. Cependant, cela ne devrait pas être le cas dans des conditions expérimentales. Aussi, dans certains domaines d'application de la décomposition de tenseur en ligne comme la spectroscopie de fluorescence, l'imagerie spectrale, il est intéressant d'imposer une contrainte de non-négativité sur les matrices facteurs car les données issues de ces applications sont positives ou nulles. Nous proposons des algorithmes pour calculer cette décomposition non-négative de tenseur en ligne basé sur l'apprentissage par dictionnaire parcimonieux pour le suivi des composants chimiques dans l'eau en utilisant un ensemble de matrices d'émission et d'excitation de fluorescence. Dans ce contexte, tout d'abord, les algorithmes prennent en compte les facteurs qui ne sont pas connus mais aussi la variation du rang du tenseur. Deuxièmement, les informations extraites précédemment sont utilisées pour décomposer les nouveaux tenseurs à venir. Outre le développement de ces algorithmes, nous proposons l'acquisition en temps réel de données de fluorescence dans un environnement semi-contrôlé. Ces algorithmes ont été appliqués sur ces jeux de données réels de fluorescence afin de comparer nos algorithmes à des algorithmes de pointes. La sortie de ces algorithmes de décomposition peut être couplée aux données d'autres capteurs pour la détection des contaminants biologiques dans le cadre de la surveillance aquatique. Ces algorithmes sont présentés dans le cas particulier de la CPD non-négative des tenseurs de fluorescence de troisième ordre, mais ils ne sont pas limités à ce champ d'application et ils peuvent être facilement étendus aux tenseurs d'ordre supérieur.

**Mots-clefs :**

Traitement du signal ; Apprentissage de dictionnaire ; Décomposition canonique polyadique ; Décomposition canonique polyadique en ligne ; PARAFAC ; Tenseur ; Analyse de données multidimensionnelles ; Nonnégativité ; Sparsity ; Optimisation ; Nadam ; Séparation de sources ; Dissolved organic matter ; Spectroscopie de fluorescence.

# Abstract

In this thesis we focus on the Canonical Polyadic Decomposition problem (CPD/PARAFAC) of third-order tensors under different constraints such as real time. Canonical polyadic decomposition is used in many areas such as chemistry, biology and medicine. The data from these fields can be dynamic, which leads to the use of real-time or "online" decomposition. The main hypothesis of all existing algorithms is that the rank or the number of components of the decomposition is known and/or does not vary over time. However, this should not be the case in experimental conditions. Also, in some application domains of online tensor decomposition, such as fluorescence spectroscopy or spectral imaging, it is interesting to impose a nonnegative constraint on the factors because the data from these applications are known to be positive or zero. We propose algorithms to calculate this nonnegative decomposition of online tensors based on sparse dictionary learning methods for monitoring chemical components in water from a set of excitation and emission matrices of fluorescence (EEMs). In this context, first of all, the algorithms take into account that the factors are a priori unknown and also the rank of the decomposition can vary during the time. Second, the information previously extracted is used to decompose the new tensors. In addition to the development of these algorithms, we propose a real-time acquisition of fluorescence data in a semi-controlled environment. These algorithms were applied to these real fluorescence datasets in order to compare our algorithms to other state-of-the-art algorithms. The output of our algorithms can be coupled with data from other sensors for a best detection of biological contaminants in the context of aquatic monitoring. These algorithms are presented in the particular case of the nonnegative CPD of third-order fluorescence tensors, but they are not limited to this application field and they can be easily extended to tensors of higher order.

**Key words :**

Signal processing ; Dictionary learning ; Polyadic canonical decomposition ; Online polyadic canonical decomposition ; PARAFAC ; Tensor ; Multidimensional data analysis ; Nonnegativity ; parsimony ; Optimization ; Nadam ; Sources Separation ; Dissolved organic matter ; Fluorescence spectroscopy.

# Table des figures

1.1	Aquabox . . . . .	23
2.1	Représentation des différentes tranches d'un tenseur d'ordre 3. . . . .	34
2.2	Schéma visuel du dépliement matriciel d'un tenseur d'ordre 3. . . . .	34
2.3	Exemple d'un sous-tenseur. . . . .	35
2.4	Exemple de recouvrement de sous-tenseur. . . . .	36
2.5	Décomposition de Tucker d'un tenseur $\mathcal{T}_{i,j,k}$ . . . . .	37
2.6	Exemple de la CPD d'un tenseur d'ordre 3 vue comme une somme de tenseur de rang. . . . .	38
2.7	Tenseur de rang 1. . . . .	39
2.8	Illustration de la CPD avec surestimation du rang. $R$ correspond au rang de la CPD incluant des colonnes nulles au moins sur l'une des 3 matrices. . . . .	40
2.9	Illustration de la CPD d'un tenseur d'ordre 3 en ligne. . . . .	46
2.10	Diagramme de Jablonski (image tirée de [1]). . . . .	49
2.11	Schéma du spectrofluorimètre. . . . .	50
2.12	MEEF avec le phénomène de diffusion. . . . .	51
2.13	Exemple de tenseur de troisième ordre. Intensité de fluorescence fonction de 3 variables : longueurs d'onde d'excitation et d'émission et des échantillons. . . . .	52
2.14	MEEF illustrant l'emplacement des diffusions de Raman et Rayleigh dans un échantillon d'eau ultra pure (Image tirée de [2]). . . . .	53
2.15	Schéma de la vue de dessus de la cuve contenant l'échantillon à analyser . (Image tirée de [3]). . . . .	53
2.16	Exemple de découpage d'un ensemble d'observation des sous-tenseurs $\mathcal{T}_n$ . <b>a-</b> Cas avec chevauchement de MEEFs. <b>b-</b> Cas sans chevauchement de MEEFs. . . . .	55
2.17	Exemple imaginaire de données de fluorescence en ligne avec des profils de concentration de composants fluorescents (unités arbitraires) et la variation de rang physique $\tilde{R}_n$ . . . . .	55
2.18	Construction des MEEFs synthétiques. (a) : Simulation de MEEFs bruitées. (b) : MEEF filtrée. (c) : MEEF non bruitées. Excitation : Colonnes des MEEFs. Emission : lignes des MEEFs . . . . .	57
2.19	Corrélation cas 1. (a) : Spectres d'émission de 4 fluorophores. (b) : Spectres d'excitation de 4 fluorophores. . . . .	58
2.20	Corrélation cas 2. (a) : Spectres d'émission de 4 fluorophores. (b) : Spectres d'excitation de 4 fluorophores. . . . .	59
2.21	Corrélation cas 3. (a) : Spectres d'émission de 4 fluorophores. (b) : Spectres d'excitation de 4 fluorophores. . . . .	59
2.22	Erreur moyenne de reconstruction en fonction de $\alpha$ des différents algorithmes pour le cas 1. . . . .	61

2.23	Erreur moyenne de reconstruction en fonction de $\alpha$ des différents algorithmes pour le cas 2. . . . .	61
2.24	Erreur moyenne de reconstruction en fonction de $\alpha$ des différents algorithmes pour le cas 3. . . . .	62
4.1	Spectres excitation-émission . . . . .	84
4.2	Exemples de profils temporels. . . . .	85
4.3	Exemple d'un profil modèle. . . . .	86
4.4	Exemple d'apparition et de disparition. . . . .	87
4.5	Profil concentration série 2 (disparition d'un composant entre les MEEFs 315-415 ) . . . . .	88
4.6	Concentrations de référence et concentrations estimées de la série 1 . . . . .	89
4.7	Spectres d'excitation et d'émission de référence et spectres d'excitation et d'émission estimés de la série 1 . . . . .	90
4.8	Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 1 : variations des concentrations sans disparition de composant . . . . .	90
4.9	Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 2 : variations de concentrations avec une disparition d'un composant . . . . .	91
4.10	Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 3 obtenue avec l'algorithme OSNNCP1 : cas apparition . . . . .	92
4.11	Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 3 obtenue avec l'algorithme OSNNCP2 : cas apparition . . . . .	93
4.12	Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 3 obtenue avec l'algorithme ROSNNCPD : cas apparition . . . . .	94
4.13	Photo du Dispositif expérimental pour l'acquisition de données de fluorescence en ligne. . . . .	96
4.14	Raspberry model pi 4. . . . .	97
4.15	Dispositif expérimental pour l'acquisition de données de fluorescence en ligne. . . . .	98
4.16	Le spectrofluorimètre Hitachi F7000. . . . .	98
4.17	Spectres d'excitation et d'émission des fluorophores estimés par l'algorithme SNNCPD. . . . .	99
4.18	Ligne verte : Profil de contribution du bruit de fond. En pointillés : les 4 fluorophores décrits ci-dessus. Tous les spectres sont normalisés. . . . .	100
4.19	Image du bruit de fond obtenue après CPD. . . . .	100
4.20	Spectres d'émission et d'excitation des fluorophores en phase d'initialisation ( $\mathcal{T}_0$ ). En haut : spectres d'émission, milieu : spectres d'excitation. Lignes rouges : spectres estimés, lignes noires : spectres réels. Tous les spectres sont normalisés. En bas : concentration de fluorophores dans la phase d'initialisation ( $\mathcal{T}_0$ ). Ligne pleine : contributions « vraies » (la contribution du bruit de fond est inconnue), ligne pointillée : les contributions estimées. Toutes les concentrations sont normalisées. . . . .	105
4.21	Emission et excitation et spectres des fluorophores en phase d'initialisation ( $\mathcal{T}_3$ ). En haut : Spectres d'émission, en bas : Spectres d'excitation. Points rouges : spectres estimés, lignes noires : spectres réels. Tous les spectres sont normalisés. . . . .	107
4.22	Erreur moyenne ( $E_A, E_B, E_C$ ) sur les matrices facteurs dans le cas sans chevauchement. . . . .	109
4.23	Qualité de reconstruction relative pour le cas sans chevauchement. . . . .	110
4.24	Erreur moyenne ( $E_A, E_B, E_C$ ) pour le cas avec 50% de chevauchement. . . . .	111
4.25	Qualité de reconstruction relative pour le cas avec 50% de chevauchement. . . . .	112
4.26	Erreur moyenne ( $E_A, E_B, E_C$ ) pour le cas avec 90% de chevauchement. . . . .	114

4.27	Qualité de reconstruction relative pour le cas avec 90% de chevauchement. . . . .	115
5.1	Cartographie des principaux fluorophores en environnement. Image tirée de [4]. . .	120
5.2	Arrivé de l'eau de mer au sein du laboratoire SSL@MM. . . . .	121
5.3	Situation géographique du site (Endoume). . . . .	122
5.4	La pompe péristaltique. . . . .	123
5.5	Schéma du dispositif d'acquisition en ligne. . . . .	123
5.6	Schéma du dispositif d'acquisition en ligne. . . . .	124
5.7	Surveillance du dispositif en ligne. . . . .	124
5.8	Rappel du principe du traitement des MEEFs en ligne et in situ. . . . .	125
5.9	Image de la MEEF anormale trouvée lors du tri manuel (qui a été supprimée) avant la CPD et après suppression des phénomènes de diffusion. . . . .	126
5.10	Image des MEEFs du sous-tenseur $\mathcal{T}_0$ . . . . .	127
5.11	Image des fluorophores présents dans le sous-tenseur $\mathcal{T}_0$ . . . . .	128
5.12	Spectres d'excitation et d'émission des 3 fluorophores présents dans le sous-tenseur $\mathcal{T}_0$ . . . . .	129
5.13	Erreurs de reconstruction des algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD sur les sous-tenseurs. Cas sans chevauchement. . . . .	129
5.14	Sous-tenseur 57 qui est constitué de MEEFs anormales. . . . .	130
5.15	Erreur de reconstruction des algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD sur tous les sous-tenseurs. Cas avec chevauchement. . . . .	131
5.16	Profil de concentration de chaque composant au cours du temps avec R=7. . . . .	132
5.17	Planche de MEEFs prises entre 140 et 160 MEEFs. . . . .	133
A.1	microplaques. . . . .	140
A.2	Dilution en cascade. . . . .	140
A.3	Ferme aquacole . . . . .	141
A.4	Image du capteur de Biocenor <i>in situ</i> . . . . .	143
A.5	Tracé en gris : évolution sur 20 jours d'oxygène dissous dans l'eau (%). Tracé en bleu température de l'eau (° C). . . . .	143
A.6	Valeurs prédites en 24 heures (ligne bleue) pour le 20e jour, basées sur des données collectées (ligne noire), avec 80% (zone remplie de bleu profond) et 95% (zone remplie de bleu clair). . . . .	144
B.1	Les différentes étapes de surveillance d'un environnement aquatique. . . . .	148
B.2	MEEF avec le phénomène de diffusion. . . . .	149
B.3	Exemple de tenseur de troisième ordre. Intensité de fluorescence fonction de 3 variables : longueurs d'onde d'excitation et d'émission et des échantillons. . . . .	149
B.4	MEEF illustrant l'emplacement des diffusions de Raman et Rayleigh dans un échantillon d'eau ultra pure (Image tirée de [2]). . . . .	150
B.5	Illustration d'un autoencodeur avec X l'entrée et X' la sortie. . . . .	153
B.6	Illustration de notre architecture utilisé pour l'entraînement des données. . . . .	153
B.7	Les différentes étapes de l'utilisation de notre autoencodeur. . . . .	155
B.8	Suppression de la diffusion sur une MEEF. (a) MEEF bruitée (avec la diffusion). (b) MEEF Corrigée avec la méthode de Zepp. (b) MEEF Corrigée avec la méthode de l'autoencodeur. . . . .	155
B.9	Résultats de la CPD obtenus avec un traitement des MEEFs par autoencodeur. . . .	156

B.10 Images des composants obtenus avec un traitement des MEEF par autoencodeur après CPD. . . . .	157
B.11 Images des composants obtenus avec un traitement des MEEF par la méthode de Zepp. . . . .	157
B.12 Résultats de la décomposition obtenus avec un traitement des MEEF par la méthode de Zepp. . . . .	158

# Liste des tableaux

4.1	Les différents hyperparamètres utilisés par chaque algorithme. . . . .	95
4.2	Les différents sous-tenseurs $\mathcal{T}_n$ avec leur rang vrai correspondant de la première base de donnée. . . . .	102
4.3	Les différents sous-tenseurs $\mathcal{T}_n$ avec leur rang vrai correspondant. . . . .	103
4.4	Estimation du rang dans le cas sans chevauchement. . . . .	104
4.5	Erreur moyenne sur les matrices facteurs dans le cas sans chevauchement. . . . .	104
4.6	Estimation du rang dans le cas sans chevauchement. . . . .	106
4.7	Erreur moyenne sur les matrices facteurs dans le cas de chevauchement. . . . .	106
4.8	Les différents hyperparamètres utilisés par chaque algorithme pour ce dataset. . . .	106
4.9	Estimation du rang dans le cas sans chevauchement. . . . .	108
4.10	Estimation du rang avec 50% de chevauchement. . . . .	111
4.11	Estimation du rang avec 90% de chevauchement. . . . .	113
4.12	Moyenne des temps d'exécution (s) des 3 algorithmes dans le cas sans chevauchement.	115
4.13	Moyenne des temps d'exécution (s) des 3 algorithmes dans le cas de recouvrement à 50%. . . . .	116
4.14	Moyenne des temps d'exécution (s) des 3 algorithmes dans le cas de recouvrement à 90%. . . . .	116
4.15	Les différents hyperparamètres utilisés par chaque algorithme pour ce jeu de données.	117
5.1	Résumé des fluorophores décrits dans la littérature. . . . .	127
5.2	Les différents hyperparamètres utilisés par chaque algorithme. . . . .	130
5.3	Résumé de l'expérience avec les différents hyperparamètres utilisés par chaque algorithme. Cas avec chevauchement. . . . .	132



# Liste des algorithmes

- 1    Algorithme DIAG à l'ordre 3 . . . . . 41
- 2    ALS . . . . . 42
- 3    Descente de Gradient . . . . . 43
- 4    Forme générale de notre algorithme de comparaison . . . . . 60
- 5    Nadam optimizer . . . . . 70
- 6    Sparse NonNegative Canonical Polyadic Decomposition (SNNCPD). Phase d'initialisation . . . . . 73
- 7    Online Sparse and Nonnegative for Dictionary Tensor Learning . . . . . 75
- 8    Online Sparse and Nonnegative CPD 1 . . . . . 76
- 9    Online Sparse and Nonnegative CPD 2 . . . . . 77
- 10  Recursive Online Sparse Nonnegative CPD (ROSNNCPD) . . . . . 80



# Table des matières

<b>Remerciements</b>	<b>1</b>
<b>Résumé</b>	<b>3</b>
<b>Abstract</b>	<b>5</b>
<b>Table des figures</b>	<b>6</b>
<b>Liste des tableaux</b>	<b>10</b>
<b>Liste des algorithmes</b>	<b>11</b>
<b>1 Introduction générale</b>	<b>21</b>
1.1 Contexte de la thèse . . . . .	23
1.2 Objectif et plan de la thèse . . . . .	26
1.2.1 Objectif . . . . .	26
1.2.2 Structure du manuscrit . . . . .	26
<b>2 Décomposition Canonique Polyadique de tenseur et application à la spectroscopie de fluorescence</b>	<b>29</b>
2.1 Définitions, notations et algèbre tensorielle . . . . .	31
2.1.1 Rappels mathématiques . . . . .	31
2.1.1.1 Produit de Hadamard . . . . .	31
2.1.1.2 Produit de Kronecker . . . . .	31
2.1.1.3 Produit de Khatri-Rao . . . . .	32
2.1.1.4 Opérations et Normes matricielles . . . . .	32
2.1.2 Définitions . . . . .	33
2.1.2.1 Tenseur . . . . .	33
2.1.2.2 Sous-tenseur . . . . .	35
2.2 Décomposition de Tucker . . . . .	37
2.3 Décomposition Canonique Polyadique . . . . .	38
2.3.1 Rang du tenseur, rang de la CPD et rang physique du tenseur . . . . .	38
2.3.1.1 Estimation pratique du rang . . . . .	39
2.3.1.2 La surestimation du rang . . . . .	39
2.3.2 Unicité de la décomposition . . . . .	40
2.3.3 Algorithme de calcul de la CPD : état de l'art . . . . .	41
2.3.3.1 La CPD avec différentes contraintes . . . . .	44
2.4 Décomposition de tenseurs en ligne . . . . .	46

2.4.1	Limitations des algorithmes CPD . . . . .	47
2.5	Lien entre la CPD et la spectroscopie de fluorescence . . . . .	49
2.5.1	Phénomène de fluorescence . . . . .	49
2.5.2	Spectroscopie de fluorescence . . . . .	49
2.5.3	MEEF et tenseur de fluorescence . . . . .	51
2.5.4	Phénomène de diffusion . . . . .	52
2.5.5	Effet d'écran . . . . .	54
2.5.6	CPD et CPD en ligne du tenseur de fluorescence . . . . .	54
2.6	Étude comparative des algorithmes de la CPD sur des données synthétiques de fluorescence . . . . .	56
2.7	Conclusion du chapitre . . . . .	63
<b>3</b>	<b>La décomposition Canonique Polyadique nonnégative de tenseurs en ligne</b>	<b>65</b>
3.1	Algorithme d'optimisation utilisé pour minimiser les fonctions de coût . . . . .	66
3.1.1	Descente de gradient stochastique (SGD) . . . . .	66
3.1.2	Descente de gradient avec élan (Momentum) . . . . .	66
3.1.3	Nesterov Accelerated Gradient (NAG) . . . . .	66
3.1.4	Adaptive gradient (Adagrad) . . . . .	67
3.1.5	Adaptive learning rate (Adadelta) . . . . .	67
3.1.6	Root Mean Square prop (RMSprop) . . . . .	67
3.1.7	ADAPtive Moment estimation (ADAM) . . . . .	68
3.1.8	Nesterov accelerated ADAPtive Moment estimation (Nadam) . . . . .	68
3.1.9	Contrainte de nonnégativité . . . . .	70
3.2	Décomposition de tenseurs avec contrainte de parcimonie et de nonnégativité. <i>Offline Step (ou phase d'initialisation)</i> . . . . .	71
3.2.1	Décomposition parcimonieuse et nonnégative de matrices . . . . .	71
3.2.2	CPD nonnégative par apprentissage de dictionnaire avec surestimation du rang . . . . .	72
3.3	Décomposition de tenseurs en ligne avec contrainte de parcimonie et de nonnégativité. <i>Online Step (ou phase en ligne)</i> . . . . .	74
3.3.1	CPD utilisant une approche d'apprentissage par dictionnaire avec surestimation du rang. Approche sans variation de rang . . . . .	74
3.3.2	Approche 1 : CPD basée sur l'apprentissage par dictionnaire avec surestimation du rang. . . . .	75
3.3.3	Approche 2 : CPD basée sur le transfert de dictionnaire. . . . .	76
3.3.4	Extension de l'approche 2 : CPD basée sur le transfert de dictionnaire . . . . .	77
3.3.5	CPD utilisant une approche récursive d'apprentissage de dictionnaire. Approche 3 . . . . .	78
3.4	Conclusion du chapitre . . . . .	81
<b>4</b>	<b>Application sur des données de fluorescence simulées et de laboratoire</b>	<b>83</b>
4.1	Simulation des données synthétiques de fluorescence . . . . .	84
4.1.1	Présentation de l'acquisition des données synthétiques de fluorescence . . . . .	84
4.1.2	Résultats sur les données synthétiques de fluorescence . . . . .	88
4.2	Acquisition des données réelles contrôlées de fluorescences en ligne . . . . .	96
4.2.1	Matériel et dispositif expérimental . . . . .	96
4.2.2	Calibration . . . . .	97
4.2.3	Bruit de fond . . . . .	99

4.2.4	Modèle théorique des concentrations . . . . .	101
4.2.5	Métriques de comparaisons . . . . .	101
4.2.6	Mise en place des bases de données de test . . . . .	102
4.3	Résultats sur les données réelles contrôlées de fluorescence . . . . .	104
4.3.1	Premier jeu de données . . . . .	104
4.3.1.1	Cas sans chevauchement de données . . . . .	104
4.3.1.2	Cas 50% chevauchement de données . . . . .	105
4.3.2	Second jeu de données : 1000 MEEFs . . . . .	107
4.3.2.1	Cas sans chevauchement de données . . . . .	107
4.3.2.2	Cas avec 50% chevauchement de données . . . . .	110
4.3.2.3	Cas avec 90% chevauchement de données . . . . .	112
4.3.3	Temps d'exécution . . . . .	114
4.4	Conclusion du chapitre . . . . .	118
<b>5</b>	<b>Application sur des données de fluorescence <i>on field</i></b>	<b>119</b>
5.1	Dispositif expérimental . . . . .	121
5.1.1	Descriptif du site . . . . .	121
5.1.2	Dispositif de mesure des signaux de fluorescence 3D . . . . .	122
5.2	Décomposition en ligne des signaux . . . . .	125
5.2.1	CPD en ligne . . . . .	125
5.3	Conclusion du chapitre . . . . .	134
	<b>Conclusion générale</b>	<b>135</b>
5.4	Contributions . . . . .	135
5.5	Perspectives . . . . .	138
	<b>Annexes</b>	<b>138</b>
<b>A</b>	<b>Détection des contaminants biologiques et Chimiques</b>	<b>139</b>
A.1	Détection des contaminants biologiques : Cas du Escherichia coli . . . . .	139
A.1.1	Quantification d'E. coli dans les eaux aquacoles . . . . .	139
A.1.2	Prédiction des E. coli chez BIOCEANOR . . . . .	141
<b>B</b>	<b>Suppression des phénomènes de diffusion en ligne des matrices de spectroscopie de fluorescence en utilisant l'apprentissage profond</b>	<b>147</b>
B.1	Introduction . . . . .	148
B.2	Travaux connexes . . . . .	151
B.2.1	Méthode de Zepp . . . . .	151
B.2.2	Méthode de filtrage par morphologie mathématique . . . . .	151
B.3	Suppression des phénomènes de diffusion en utilisant les autoencodeurs . . . . .	152
B.3.1	Architecture ou réseau utilisé . . . . .	152
B.3.2	Préparation des Données d'entraînement . . . . .	154
B.3.3	Résultats et discussions . . . . .	154
B.4	Perspectives et conclusion . . . . .	159



# Liste des acronymes

**ALS** : Alternating Least Square  
**ADMM** : Alternating Direction Method of Multipliers  
**CPD** : Canonical Polyadique Decomposition  
**NNCPD** : NonNegative Canonical Polyadique Decomposition  
**OSNNCPD** : Online NonNegative Canonical Polyadique Decomposition  
**ROSNNCPD** : Recursive Online NonNegative Canonical Polyadique Decomposition  
**ELS** : Enhanced Line Search  
**PARAFAC** : PARAllel FACtor Analysis  
**MOD** : Matière Organique Dissoute  
**MEEF** : Matrice d'Excitation d'Emission de Fluorescence  
**pH** : potentiel Hydrogène  
**NMF** : Nonnegative Matrix Factorization  
**SGD** : Stochastique Gradient Descent  
**Nadam** : Nesterov accelerated ADaptive Moment estimation  
**F** : Fluorescéine  
**TRP** : Tryptophane  
**5S8HQ** : 5-Sulfate-8-HydroxyQuinine  
**RHO** : RHOdamine  
**SVD** : Singular Value Decomposition  
**U.A** : Unités Arbitraires



# Chapitre 1

## Introduction générale

L'évolution des technologies entraîne une possibilité d'acquisition de volumes de données conséquents avec une disponibilité de données en temps réel et tout ceci souvent en provenance de plusieurs sources ou capteurs. La mise en place d'outils rapides et efficaces pour traiter ces données sont une nécessité. Ainsi, les travaux présentés dans cette thèse portent sur le traitement des données environnementales issues de la spectroscopie de fluorescence en temps réel. En effet, cette technique rapide permet le suivi des contaminations biologiques et organiques en milieu aquatique. Lorsqu'on parle de contaminations biologiques ou organiques on parle entre autres de la présence ou non de bactéries, des composants chimiques issues de diverses origines mais aussi de la Matière Organique Dissoute (MOD). Il existe depuis des années des techniques pour la caractérisation ou le suivi de ces contaminants sur la mesure de paramètres tels que la Demande Chimique en Oxygène (DCO) et la Demande Biochimique en Oxygène (DBO5). Parmi les techniques optiques de caractérisation des eaux sans ajout de réactifs, c'est-à-dire compatibles avec des mesures en ligne, il y a également le SUVA<sub>254</sub> [5], les rapports  $E_2/E_3$  ou  $E_4/E_6$  [6], et la pente SR [7] pour l'absorbance [8], ou les indices de Kalbitz [9], Ra,c, Zsolnay [10], BIX, HIX [5] pour la spectroscopie de fluorescence. Ces techniques servent souvent de bases pour des techniques de laboratoire plus lourdes, plus complexes mais plus spécifiques comme la spectrométrie de masse à résonance cyclotronique ionique à transformée de Fourier (FT-ICR-MS), la spectroscopie de résonance magnétique nucléaire (RMN) homo et hétéro-corrélée multidimensionnelle [11], la chromatographie [12], couplée à un spectomètre de masse, ...

Les procédures de suivi actuelles sont chronophages et nécessitent souvent une analyse en laboratoire, ce qui implique que les résultats ne sont pas obtenus en temps réel et donc cela ne permet pas de réaliser des prises de décisions rapides lors d'une surveillance en milieu aquatique. Afin d'atteindre les objectifs de traitement rapide nous allons nous pencher sur un outil en émergence dans le domaine de la mesure in-situ, la spectroscopie de fluorescence [13]. La spectroscopie de fluorescence est un outil de surveillance et de mesure de plus en plus utilisé dans l'étude de l'environnement et des ressources naturelles aquatiques [2]. En chimie de l'environnement, cette technique permet non seulement de caractériser des composants inconnus présents dans le milieu aquatique des prélèvements, mais constitue également un traceur de la matière organique. Connaître finement la quantité et la qualité de la matière organique dans les milieux récepteurs présente un enjeu important pour la compréhension de nombreux processus dans l'eau en environnement [5] ou dans les processus industriels de production ou de traitement des eaux [14]. Dans les milieux récepteurs aquatiques, l'utilisation de la spectroscopie de fluorescence en ligne permettrait de suivre l'évolution en temps réel des qualités et des quantités relatives des contaminants [15]. Les données issues de la spectroscopie de fluorescence peuvent être de simples valeurs, des spectres d'émission

ou d'excitation, mais le plus souvent sont des Matrices d'Emission et d'Excitation de Fluorescence (MEEFs). Ces matrices permettent de déterminer les contributions de molécules organiques dans un mélange complexe. Le rassemblement des MEEFs forme un bloc de données multidimensionnelles de troisième ordre appelé tenseur. Nous y reviendrons en profondeur dans le chapitre suivant.

L'utilisation des techniques d'analyse multidimensionnelle telles que la Décomposition Canonique Polyadique (CPD) ou PARAFAC (de l'anglais PARalel FACtor analysis) ou encore décomposition de tenseurs permet de décomposer ces données en composants pouvant être interprétés par l'utilisateur. La décomposition de tenseur est utilisée depuis plusieurs années dans des applications couvrant de nombreux domaines, notamment la psychométrie [16], les neurosciences [17], la chimio-métrie [18], la vision par ordinateur [19], l'analyse des réseaux sociaux et extraction de données [19], traitement des images biomédicales [20], traitement du signal, imagerie hyperspectrale [21], les télécommunications [22, 23, 24], etc. La CPD a également prouvé son utilité pour des applications en lien avec le sujet de la thèse, on peut citer la qualité de l'eau de production [25] et les installations de traitement des eaux [14].



FIGURE 1.1 – Aquabox

## 1.1 Contexte de la thèse

D'un point de vue applicatif, les résultats de la CPD peuvent être une indication sur la présence ou non de contaminants biologiques c'est-à-dire la présence de bactéries dans le milieu [15]. Mais d'autres travaux montrent qu'une conclusion est difficile à tirer uniquement à partir de ces résultats [26]. C'est dans ce contexte qu'une collaboration est établie avec Bioceanor.

Bioceanor est une startup spécialisée dans la biologie marine. Bioceanor est notre partenaire économique. Fondée en fin 2017, Bioceanor a pour vocation de développer et déployer des services innovants pour la surveillance prédictive des océans [27]. Composée d'entrepreneurs, de biologistes, data scientists et d'expert des océans, l'équipe de Bioceanor s'engage dans l'atteinte des objectifs de développements durables fixés par l'Organisation des Nations Unies. Parmi toutes les solutions techniques proposées par Bioceanor, cette entreprise a développé un capteur nommé "Aquabox" (Figure 1.1) pour mesurer les paramètres physico-chimiques de l'eau en temps réel. Ces données issues de ce capteur (température, turbidité, pH,...) constituent l'entrée de l'algorithme (Figure A.4) développé au sein de Bioceanor afin de prédire la présence de bactéries comme indiqué dans l'annexe A.

A terme, les données fournies par nos algorithmes pourraient être combinées à celles qui sont mesurées par le capteur Aquabox afin d'améliorer la surveillance aquatique.

La surveillance des milieux aquatiques est l'un des défis majeurs dans l'étude de l'environnement, principalement celle de l'évolution des contaminants biologiques ou chimiques. Il y a des composés dit autochtones qui sont issus d'une activité sur place, ou provenant du milieu étudié comme par exemple l'activité microbienne [28]. Il y a également les composés d'origine allochtone c'est-à-dire qui proviennent de l'extérieur du système étudié, que l'on appelle parfois transfert horizontal ; ce sont les apports de matière d'origine externe, comme par exemple pour le milieu côtier les apports de rivière ou les apports éoliens. En particulier dans le cas des contaminations que l'on peut observer, ces dernières sont souvent de type allochtone, et s'introduisent dans le milieu et le dégrade ou le perturbe.

Pour les quantifier et les mesurer plusieurs techniques de caractérisation sont utilisées comme mentionné ci-dessus et présentent des avantages et inconvénients. Certaines de faible résolution

mais rapide comme la spectroscopie UV-Visible ou les sondes et capteur, d'autres sont nettement plus spécifiques et sensibles, souvent très riches en informations, mais très lourdes à mettre en place, nécessitant un traitement des échantillons, des instrumentations lourdes, de l'expertise et du temps comme les techniques de spectroscopie de masse (GC-MS, ICP-MS, HPLC-MS, ...). De plus, ces techniques ne peuvent pas être implémentées *in situ* ou alors au prix d'effort matériel et financier considérables. Aussi, les techniques actuelles ne permettent pas un suivi des paramètres de contaminations biologiques ou organiques en temps réel [29] et ce qui nécessite des moyens logistiques importants.

Cette thèse, entre dans le cadre d'un besoin particulier : la détection en temps quasi réel de la contamination bactériologique des eaux d'un parc conchilicole par *E. Coli* par spectroscopie de fluorescence. En effet, lors de la production de moules, et en particulier de leur récolte, si ces dernières sont contaminées par *E. Coli* elles ne sont pas commercialisables et cela fait une perte économique conséquente soit par la perte du lot, soit par la nécessité de mettre en bassin de décontamination le lot de moule incriminé. Un outil permettant de détecter en amont la contamination bactériologique permettrait au producteur de réagir à temps et d'éviter l'exposition des moules à la contamination. Des travaux antérieurs montrent qu'il est possible de détecter des signaux de fluorescence spécifique à la contamination bactériologique, et l'utilisation de la fluorescence [25], ou matrice d'excitation et d'émission de fluorescence, permet tenir compte de l'ensemble des composants fluorescents présents et de corriger des interférences ou d'autres contaminants quand ils sont présents. La spectroscopie de fluorescence attire donc notre attention car les données issues de cette modalité permettent d'appliquer la CPD pour avoir un suivi rapide et facilement transportable et transposable. Une autre décomposition tensorielle multilinéaire est la décomposition de Tucker [30, 31]. Cette dernière considère qu'un tenseur peut se décomposer à l'aide d'un tenseur cœur et de plusieurs matrices facteurs de manière multilinéaire (nous donnons plus de détails dans le chapitre suivant). Nous focalisons notre étude sur la décomposition tensorielle de type PARAFAC.

Du point de vue du traitement numérique des données, l'analyse en ligne et *in situ*, plus particulièrement le suivi par spectroscopie de fluorescence d'un milieu aquatique, pose trois difficultés :

**Le volume de données à traiter devient rapidement très important.** En effet, jusque-là, la manière classique consiste à faire des séries ponctuelles de prélèvements puis des mesures en laboratoire. Ce qui peut se relever fastidieux comme processus. En plus des moyens matériels énormes qui doivent être mis en place pour prélever les données, on se retrouve rapidement avec de gros volume d'échantillons. Cette méthode peut entraîner une altération des échantillons prélevés liée au transport et la conservation.

**Les résultats doivent être mis à jour en temps réel.** Avec de gros volume d'échantillons, il est pourtant impératif de mettre à jour les données en temps réel. La difficulté avec le traitement numérique des données actuelles est l'analyse en temps réel. En effet, l'analyse de la globalité du jeu de données augmentant avec le temps, la durée de décomposition augmente également, s'éloignant d'une résolution en temps réel. Le temps de calcul est en effet un élément important pour l'analyse *in situ*.

**Le nombre de composants fluorescents est inconnu et peut varier au cours du temps.** La composition d'un milieu naturel est changeante au cours du temps. En effet, de part ces apports autochtones, il y a la présence de composés attendus dont la présence sera fortement probable, avec des variations d'intensité : chlorophylle, matière organique, protéines. Mais, il y a l'apport chronique

ou ponctuel de matière autochtone qui ne sont pas attendu dans le milieu. Ces matières n'apparaissent que pendant le temps de contamination (contamination bactériologique, hydrocarbure), ou de manière continue en raison du milieu (aérosols, fleuve, ...). Il faut alors un algorithme capable de détecter l'émergence d'un nouveau composé tout en gardant la capacité de détecter les composés récurrents.

## 1.2 Objectif et plan de la thèse

### 1.2.1 Objectif

L'objectif de ce travail est la mise au point et l'évaluation d'un algorithme dérivé de PARAFAC capable de détecter des composants fluorescents déjà présents ("phase de détection") mais également d'extraire numériquement de nouveaux composants ("phase d'apprentissage") pour une surveillance en ligne et *in situ*. En résumé, à chaque intervalle de temps  $\Delta t$  nous aurons l'acquisition d'une nouvelle MEEF et cela entraînera une modification du tenseur de fluorescence en temps réel qui devra être traité.

Par conséquent, nous aurons besoin d'un traitement rapide, adaptatif (ou alors capable de traiter un gros volume de données en un temps inférieur à l'acquisition de la mesure) et peu sensible à la surestimation du nombre de composants fluorescents.

De nombreux algorithmes de décomposition de tenseur ont été développés ces dernières années. Il convient donc dans un premier temps de comparer ces algorithmes dans le cadre particulier de l'analyse de MEEF. Par ailleurs, il faut noter que la CPD/PARAFAC n'est qu'un modèle multilinéaire des données. Il y a d'autres modèles comme l'Analyse en Composantes Principales (ACP), l'Analyse en Composantes Indépendantes (ICA), ou encore la Nonnegative Matrix Factorization (NMF). Or, il est possible que ces modèles ne soient pas toujours adaptés aux données mesurées et que des effets non linéaires doivent être pris en compte. Plusieurs techniques de linéarisation des données ont été proposées [32] (cf Chapitre 2).

Ainsi, à travers ces nombreuses difficultés, nous essayerons de répondre à ces problèmes au cours de cette thèse.

### 1.2.2 Structure du manuscrit

Le manuscrit est présenté de la manière suivante :

**1. Chapitre 2 : Introduction aux décompositions tensorielles : Décomposition canonique polyadique et application à la spectroscopie de fluorescence.**

Ce chapitre rappelle des notions et des définitions propre à l'algèbre tensorielle qui nous seront utiles pour développer nos approches des décompositions en ligne. Nous proposons un bref état de l'art de la CPD et la CPD en ligne. Pour la suite, nous établissons un lien entre la décomposition de tenseur appliquée à la spectroscopie de fluorescence.

**2. Chapitre 3 : Algorithmes de décomposition Canonique Polyadique en ligne.**

La décomposition Canonique Polyadique de tenseurs en ligne est une approche qui a du succès ces dernières années. Certains algorithmes proposent une parallélisation des approches existantes et d'autres proposent des nouvelles approches de mise à jour pour traiter le problème de la décomposition en ligne. Dans ce chapitre, de nouvelles approches que l'on a développées seront présentées pour répondre à cette problématique décrite dans le chapitre 2.

**3. Chapitre 4 : Application sur des données de fluorescence : Simulation numérique et données réelles.**

Afin de valider les approches développées dans la section précédente, deux séries d'expérimentation ont été mises en place pour acquérir les données de spectroscopie de fluorescence.

La première correspond à une simulation numérique afin de valider les approches développées. La seconde a permis d'acquérir des données de fluorescence en temps réel avec diverses situations. Une comparaison des algorithmes existants et des algorithmes développés sur les données simulées est effectuée pour s'assurer de leur robustesse.

4. **Chapitre 5 : Application sur des données de fluorescence en milieu naturel *in situ*.**

Après une application sur des données synthétiques et des données réelles dans une expérience semi-contrôlée, ce chapitre décrit l'application et le comportement des algorithmes développés sur les données provenant de la mesure d'eau mer en continu *in situ*. Les différents processus d'acquisition et de mesures seront décrits, suivis de la présentation des résultats obtenus et des discussions.

5. **Chapitre 6 : Conclusion et perspectives.** Ce chapitre résume le travail effectué dans cette thèse et évoquera les pistes pour d'éventuelles améliorations.



## Chapitre 2

# Décomposition Canonique Polyadique de tenseur et application à la spectroscopie de fluorescence

CPD de tenseur et application à la spectroscopie de fluorescence Nous nous inspirons des travaux réalisés dans [2, 33, 34, 35, 36]. Nous rappelons ici les outils mathématiques qui nous serviront par la suite, la notion de CPD en traitement du signal et l'analyse de données, les définitions, ainsi qu'un état de l'art sur la CPD. Nous faisons par la suite un lien entre la CPD et son application en spectroscopie de fluorescence. Enfin, une comparaison sur des données synthétiques de fluorescence de quelques algorithmes de décomposition de tenseur est présentée.

Dans la suite de cette thèse, les notations suivantes seront utilisées :

- Ensemble des réels :  $\mathbb{R}$ .
- Ensemble des réels positifs :  $\mathbb{R}_+$ .
- Les tenseurs sont notés par des lettres majuscules calligraphiques :  $\mathcal{T}$ .
- Les matrices sont notées par des lettres majuscules en gras :  $\mathbf{A}$ .
- Les vecteurs sont notés par des lettres minuscules en gras :  $\mathbf{a}$ .
- Les composantes d'une matrice  $\mathbf{A}$  sont notées :  $A_{i,j}$ .
- La  $i^{\text{ème}}$  ligne de la matrice  $\mathbf{A}$  est notée :  $A_{i,:}$ .
- La  $j^{\text{ème}}$  colonne de la matrice  $\mathbf{A}$  est notée :  $A_{:,j}$ .
- La transposée d'une matrice :  $\mathbf{A}^T$ .
- Norme  $L_{0,0}$  de la matrice  $\mathbf{A}$  :  $\|\mathbf{A}\|_{0,0}$ .
- Norme  $L_{1,1}$  de la matrice  $\mathbf{A}$  :  $\|\mathbf{A}\|_{1,1}$ .
- Norme  $L_{1,2}$  de la matrice  $\mathbf{A}$  :  $\|\mathbf{A}\|_{1,2}$ .
- Norme  $L_{2,1}$  de la matrice  $\mathbf{A}$  :  $\|\mathbf{A}\|_{2,1}$ .
- Norme nucléaire de la matrice  $\mathbf{A}$  :  $\|\mathbf{A}\|_*$ .
- Norme de Frobenius de la matrice  $\mathbf{A}$  :  $\|\mathbf{A}\|_F$ , d'un tenseur  $\|\mathcal{T}\|_F$ .
- Une lettre minuscule désigne un scalaire :  $a$ .
- Rang d'une matrice ou d'un tenseur :  $R$ .
- Rang physique d'une matrice ou d'un tenseur :  $\tilde{R}$ .
- Le produit de Hadamard :  $\boxtimes$ .
- Le produit de Kronecker :  $\otimes$ .
- Le produit de Khatri-Rao :  $\odot$ .
- Une matrice  $(R, R)$  remplie de 1 :  $\mathbf{1}_R$ .

### 30 Décomposition Canonique Polyadique de tenseur et application à la spectroscopie de fluorescence

- Opérateur de dérivée partielle :  $\partial$  .
- Projection de la matrice  $\mathbf{A}$  sur  $\mathbb{R}_+$  :  $[\mathbf{A}]_+$  .
- Equivalent :  $\equiv$  .
- Opérateur qui renvoie les plus grandes valeurs passées en paramètre :  $\max\{.\}$  .
- Trace d'une matrice  $\mathbf{A}$  :  $\text{Tr}\{A\}$  .
- Matrice identité de taille  $N$  :  $\mathbf{I}_A$  .
- Inverse de la matrice  $\mathbf{A}$  :  $\mathbf{A}^{-1}$  .
- Pseudo-inverse de la matrice  $\mathbf{A}$  :  $\mathbf{A}^+$

## 2.1 Définitions, notations et algèbre tensorielle

### 2.1.1 Rappels mathématiques

Nous présentons ici les opérations algébriques utilisées dans le cadre de notre travail. Ces notions nous permettront, par la suite, de formuler le problème de décomposition tensorielle.

#### 2.1.1.1 Produit de Hadamard

Prenons deux matrices  $\mathbf{A} \in \mathbb{R}^{I \times J}$  et  $\mathbf{B} \in \mathbb{R}^{I \times J}$  de mêmes dimensions, le produit de Hadamard est le produit terme à terme entre ces deux matrices [37]. Nous le notons  $\mathbf{A} \square \mathbf{B} \in \mathbb{R}^{I \times J}$ . Les termes de la matrice résultante du produit de Hadamard s'expriment de la façon suivante :

$$(\mathbf{A} \square \mathbf{B})_{i,j} = A_{i,j} B_{i,j}. \quad (2.1)$$

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

1. associativité :  $\mathbf{A} \square \mathbf{B} \square \mathbf{C} = (\mathbf{A} \square \mathbf{B}) \square \mathbf{C} = \mathbf{A} \square (\mathbf{B} \square \mathbf{C})$
2. distributivité :  $\mathbf{A} \square (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \square \mathbf{B}) + (\mathbf{A} \square \mathbf{C})$
3. commutativité :  $\mathbf{A} \square \mathbf{B} = \mathbf{B} \square \mathbf{A}$
4.  $(\mathbf{A} \square \mathbf{B})^T = \mathbf{A}^T \square \mathbf{B}^T$
5.  $\alpha(\mathbf{A} \square \mathbf{B}) = (\alpha\mathbf{A}) \square \mathbf{B} = \mathbf{A} \square (\alpha\mathbf{B})$
6. Si  $\mathbf{A}$  et  $\mathbf{B}$  sont des matrices diagonales, alors  $\mathbf{A} \square \mathbf{B} = \mathbf{A}\mathbf{B}$ .

#### 2.1.1.2 Produit de Kronecker

Le produit de Kronecker noté  $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{IK \times JR}$  entre deux matrices  $\mathbf{A} \in \mathbb{R}^{I \times J}$  et  $\mathbf{B} \in \mathbb{R}^{K \times R}$  est défini de la manière suivante [38] :

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} A_{1,1}\mathbf{B} & A_{1,2}\mathbf{B} & \cdots & A_{1,J}\mathbf{B} \\ A_{2,1}\mathbf{B} & A_{2,2}\mathbf{B} & \cdots & A_{2,J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ A_{I,1}\mathbf{B} & A_{I,2}\mathbf{B} & \cdots & A_{I,J}\mathbf{B} \end{pmatrix}. \quad (2.2)$$

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

1. associativité :  $\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C} = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})$
2. distributivité :
  - $\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) + (\mathbf{A} \otimes \mathbf{C})$
  - $(\mathbf{B} + \mathbf{C}) \otimes \mathbf{A} = (\mathbf{B} \otimes \mathbf{A}) + (\mathbf{C} \otimes \mathbf{A})$
3. non-commutativité :  $\mathbf{A} \otimes \mathbf{B} \neq \mathbf{B} \otimes \mathbf{A}$
4.  $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$
5. si  $\mathbf{A}$  et  $\mathbf{B}$  sont carrées et inversibles alors  $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$
6.  $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{A}\mathbf{C}) \otimes (\mathbf{B}\mathbf{D})$
7.  $\alpha(\mathbf{A} \otimes \mathbf{B}) = (\alpha\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (\alpha\mathbf{B})$ .

### 2.1.1.3 Produit de Khatri-Rao

Soient deux matrices  $\mathbf{A} = [A_{:,1}, A_{:,2}, \dots, A_{:,J}] \in \mathbb{R}^{I \times J}$  et  $\mathbf{B} = [B_{:,1}, B_{:,2}, \dots, B_{:,J}] \in \mathbb{R}^{K \times J}$ , le produit de Khatri-Rao entre  $\mathbf{A}$  et  $\mathbf{B}$ , noté  $\mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{IK \times J}$ , est défini comme le produit de Kronecker entre les colonnes de  $\mathbf{A}$  avec les colonnes de  $\mathbf{B}$  correspondantes [39], ainsi nous avons :

$$\mathbf{A} \odot \mathbf{B} = [A_{:,1} \otimes B_{:,1}, \dots, A_{:,J} \otimes B_{:,J}]. \quad (2.3)$$

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

1. associativité :  $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = (\mathbf{B} \odot \mathbf{C}) \odot \mathbf{A}$
2. distributivité :  $(\mathbf{A} + \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot \mathbf{C} + \mathbf{B} \odot \mathbf{C}$
3. non-commutativité :  $\mathbf{A} \odot \mathbf{B} \neq \mathbf{B} \odot \mathbf{A}$
4.  $(\mathbf{A} \odot \mathbf{B})^T (\mathbf{A} \odot \mathbf{B}) = \mathbf{A}^T \mathbf{A} \square \mathbf{B}^T \mathbf{B}$
5.  $\alpha(\mathbf{A} \odot \mathbf{B}) = (\alpha\mathbf{A}) \odot \mathbf{B} = \mathbf{A} \odot (\alpha\mathbf{B})$

### 2.1.1.4 Opérations et Normes matricielles

**Norme  $L_{0,0}$  :** Soit  $\|\mathbf{A}\|_{0,0}$  la norme  $L_{0,0}$  de la matrice  $\mathbf{A} \in \mathbb{R}^{I \times J}$ . Elle est définie par :

$$\|\mathbf{A}\|_{0,0} = \sum_{i=1}^I \sum_{j=1}^J \mathbf{A}_{i,j} / \mathbf{A}_{i,j} \quad \forall \mathbf{A}_{i,j} \neq 0 \quad (2.4)$$

En pratique c'est le nombre d'élément non nuls de  $\mathbf{A}$ .

**Norme  $L_{1,1}$  :** Soit  $\|\mathbf{A}\|_{1,1}$  la norme  $L_{1,1}$  de la matrice  $\mathbf{A} \in \mathbb{R}^{I \times J}$ . Elle est définie par :

$$\|\mathbf{A}\|_{1,1} = \sum_{i=1}^I \sum_{j=1}^J |\mathbf{A}_{i,j}| \quad (2.5)$$

**Norme de Frobenius :** Soit  $\|\mathbf{A}\|_F$  la norme de Frobenius de la matrice  $\mathbf{A} \in \mathbb{R}^{I \times J}$ . Celle-ci est définie par

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^I \sum_{j=1}^J |\mathbf{A}_{i,j}|^2} \quad (2.6)$$

La norme de Frobenius peut aussi s'écrire en fonction de la trace :  $\|\mathbf{A}\|_F = \text{Tr}\{\mathbf{A}\mathbf{A}^T\}^{\frac{1}{2}}$ . Soient deux matrices  $\mathbf{A}$  et  $\mathbf{B}$  le produit scalaire est défini comme :  $\langle \mathbf{A}, \mathbf{B} \rangle = \text{Tr}\{\mathbf{A}^T \mathbf{B}\}$

**Norme  $L_{1,2}$  :** Soit  $\|\mathbf{A}\|_{1,2}$  la norme  $L_{1,2}$  de la matrice  $\mathbf{A} \in \mathbb{R}^{I \times J}$ . Elle est définie par :

$$\|\mathbf{A}\|_{1,2} = \sum_{j=1}^J \sqrt{\sum_{i=1}^I \mathbf{A}_{i,j}^2} \quad (2.7)$$

**Norme  $L_{2,1}$  :** Soit  $\|\mathbf{A}\|_{2,1}$  la norme  $L_{2,1}$  de la matrice  $\mathbf{A} \in \mathbb{R}^{I \times J}$ . Elle est définie par :

$$\|\mathbf{A}\|_{2,1} = \sum_{i=1}^I \sqrt{\sum_{j=1}^J A_{i,j}^2} \quad (2.8)$$

**Norme nucléaire :** Soit  $\|\mathbf{A}\|_*$  la norme nucléaire ou nuclear norm de la matrice  $\mathbf{A} \in \mathbb{R}^{I \times J}$  avec  $r = \text{rang}(\mathbf{A})$ . Elle est égale à la somme de ses valeurs singulières. Elle est donc définie par :

$$\|\mathbf{A}\|_* = \sum_{i=1}^r \sigma_i \quad (2.9)$$

avec  $\sigma_i$  les valeurs singulières de  $\mathbf{A}$ . Les applications de la norme nucléaire incluent l'approximation de rang et l'acquisition comprimée. Par exemple, le problème de minimisation de rang consiste à trouver une matrice  $\mathbf{X}$  de rang minimum qui satisfait  $\mathbf{A}\mathbf{X}=\mathbf{b}$ . Il a été montré qu'il s'agit d'un problème d'optimisation non convexe de complexité NP. Il s'avère que la norme nucléaire peut être utilisée comme une relaxation convexe de ce problème d'optimisation, ce qui simplifie grandement le problème et permet des applications intéressantes telles que des algorithmes accélérés pour la complétion de matrices (compressed sensing).

**Opérateur de projection sur  $\mathbb{R}^+$**  Soit  $\mathbf{A} \in \mathbb{R}^{I \times J}$  alors

$$\mathbf{A} = [\mathbf{A}]_+ \equiv \forall(i, j) A_{i,j} = \max\{0, A_{i,j}\}. \quad (2.10)$$

**Dérivation de la norme de Frobenius d'une expression matricielle :** Soient les matrices  $\mathbf{A}$ ,  $\mathbf{B}$  et  $\mathbf{X}$ . Soit la fonction  $\|\mathbf{C} + \mathbf{A}\mathbf{X}\mathbf{B}\|_F$ , sa dérivée partielle par rapport à  $\mathbf{X}$  est donnée par :

$$\frac{\partial \|\mathbf{C} + \mathbf{A}\mathbf{X}\mathbf{B}\|_F}{\partial \mathbf{X}} = \frac{\partial \text{Tr}\{(\mathbf{C} + \mathbf{A}\mathbf{X}\mathbf{B})(\mathbf{C} + \mathbf{A}\mathbf{X}\mathbf{B})^T\}}{\partial \mathbf{X}} = 2\mathbf{A}^T(\mathbf{C} + \mathbf{A}\mathbf{X}\mathbf{B})\mathbf{B}^T \quad (2.11)$$

## 2.1.2 Définitions

### 2.1.2.1 Tenseur

Un tenseur peut être défini de nombreuses manières, en particulier comme une application multilinéaire. Dans notre domaine comme un tableau de nombres à plus de deux indices [33, 40]. En d'autres termes, le nombre d'indices ici noté  $N$ , définit l'ordre du tenseur. Les dimensions d'un tenseur sont aussi appelées modes. Dans cette thèse, nous utilisons un tenseur d'ordre 3 nommé  $\mathcal{T}$  généralement de dimensions  $(I, J, K)$ .

Un tenseur d'ordre 3 peut être partitionné en trois séries de tranches matricielles possibles : les tranches frontales, les tranches horizontales et les tranches verticales comme l'indique la Figure 2.1.

Le tenseur  $\mathcal{T}$  peut être déplié sous forme de matrices [41] :

1. Dépliage suivant le mode 1 : le dépliage suivant le mode 1 est lorsqu'on isole la première dimension  $I$  sur les lignes de la matrice de dépliage et les autres modes sont alors regroupés sur ses colonnes. On obtient alors la matrice  $\mathbf{T}^{I,KJ}$  de taille  $(I, KJ)$  (voir Figure 2.2). (Rem. En permutant les rôles des modes 2 et 3, on obtiendrait la matrice  $\mathbf{T}^{I,JK}$ )

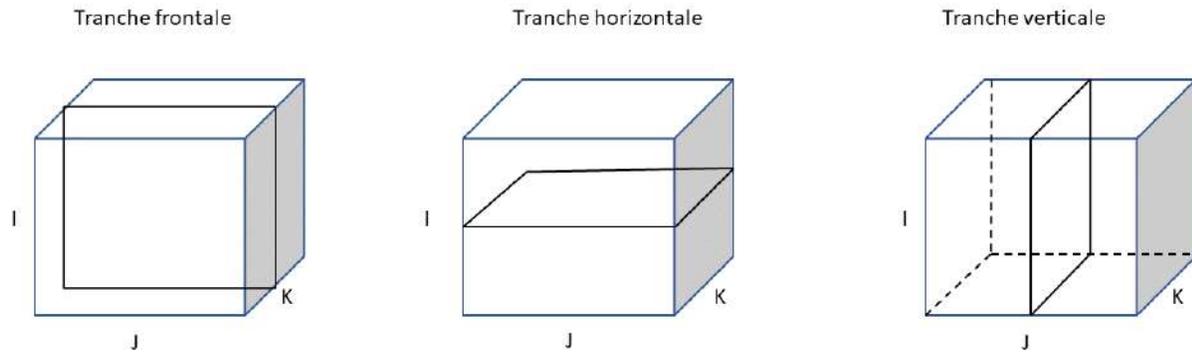


FIGURE 2.1 – Représentation des différentes tranches d'un tenseur d'ordre 3.

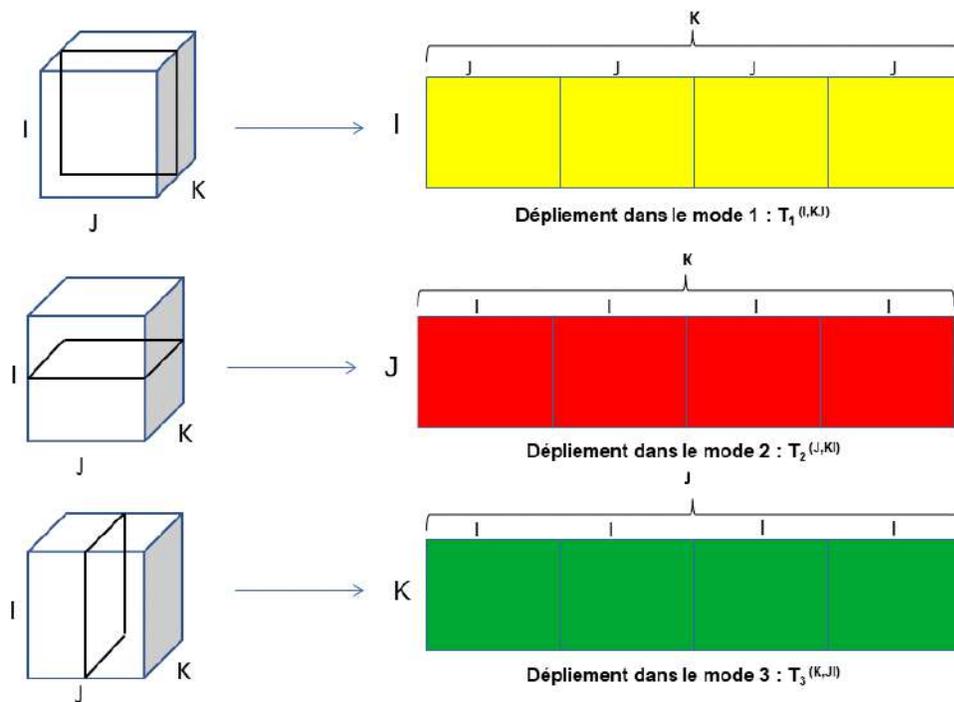


FIGURE 2.2 – Schéma visuel du dépliement matriciel d'un tenseur d'ordre 3.

2. Dépliement suivant le mode 2 : le dépliement suivant le mode 2 est lorsqu'on isole la seconde dimension  $J$  sur les lignes de la matrice de dépliement et les autres modes sont alors regroupés sur ses colonnes. On obtient alors la matrice  $\mathbf{T}^{J,KI}$  de taille  $(J, KI)$  (voir Figure 2.2).
3. Dépliement suivant le mode 3 : le dépliement suivant le mode 3 est lorsqu'on isole la troisième dimension  $K$  sur les lignes de la matrice de dépliement et les autres modes sont alors regroupés sur ses colonnes. On obtient alors la matrice  $\mathbf{T}^{K,JI}$  de taille  $(K, JI)$  (voir Figure 2.2).

### 2.1.2.2 Sous-tenseur

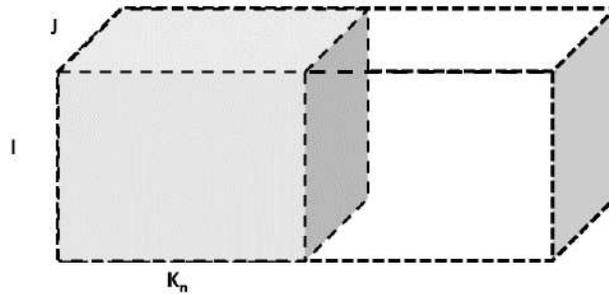


FIGURE 2.3 – Exemple d'un sous-tenseur.

Un sous-tenseur est une partie d'un tenseur originel suivant un mode et le sous-tenseur est du même ordre que le tenseur originel. On considère un tenseur originel de taille  $(I, J, K)$  et un sous-tenseur  $\mathcal{T}_n$  de taille  $(I, J, K_n)$  formé de  $K_n$  tranches frontales (Figure 2.1) consécutives notées  $\mathcal{T}_{\dots,k}$ .

Le tenseur originel  $\mathcal{T}$  peut être découpé en plusieurs sous-tenseurs  $\mathcal{T}_n$  avec un taux de recouvrement  $\eta$  (compris entre 0 et 1 inclus comme l'indique la Figure 2.4) qui définit le nombre  $\eta K_n$  de tranches frontales communes entre deux sous-tenseurs consécutifs (le taux de recouvrement  $\eta$  est choisi de telle façon que le produit  $\eta K_n$  soit entier). Dans ce cas, l'indice  $k$  des tranches frontales constituant le sous-tenseur  $\mathcal{T}_n$  est compris entre :

$$[(n-1)(1-\eta)K_n + 1; (n - (n-1)\eta)K_n] \quad (2.12)$$

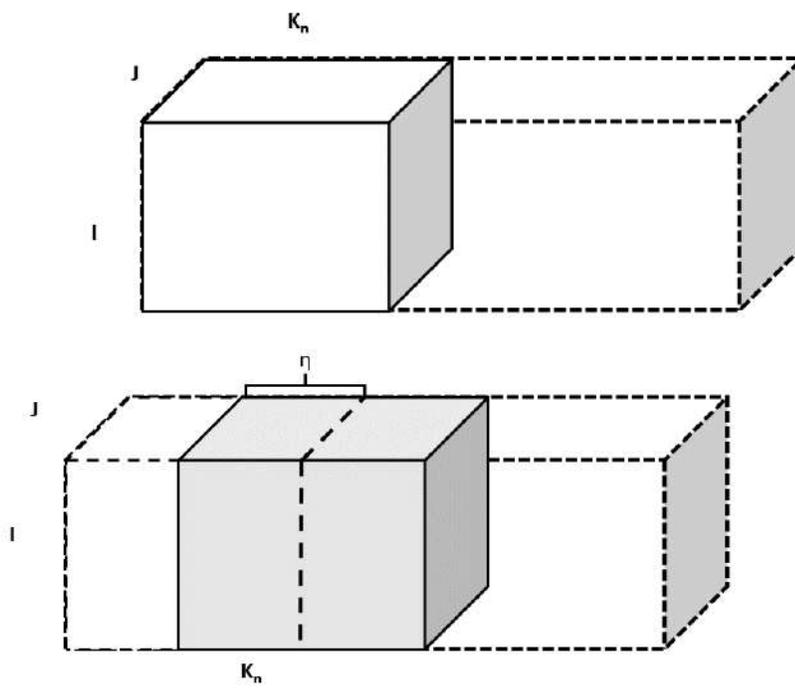


FIGURE 2.4 – Exemple de recouvrement de sous-tenseur.

## 2.2 Décomposition de Tucker

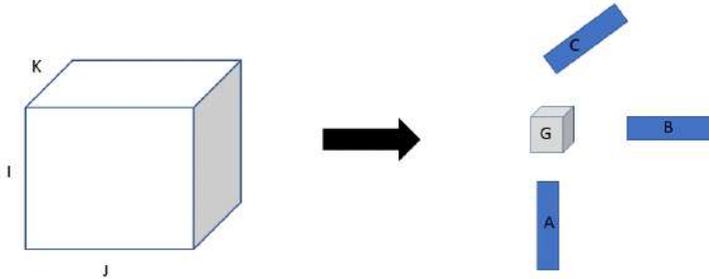


FIGURE 2.5 – Décomposition de Tucker d'un tenseur  $\mathcal{T}_{i,j,k}$ .

Il existe de nombreuses manières de décomposer des tenseurs. Ici, nous nous intéressons à la décomposition de Tucker [30, 42, 31]. Soit un tenseur  $\mathcal{T}$  d'ordre 3, l'objectif de Tucker est de réécrire ce tenseur en fonction d'un tenseur cœur  $\mathcal{G}$  et de matrices facteurs. Ce qui rend la décomposition de Tucker générale parmi les décompositions multilinéaires :

$$\forall i, j, k, \quad \mathcal{T}_{i,j,k} = \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R \mathcal{G}_{p,q,r} A_{ip} B_{jq} C_{kr} \quad (2.13)$$

$\mathcal{G}$  est du même ordre que  $\mathcal{T}$  et est appelé tenseur cœur. De manière générale, la décomposition de Tucker n'est pas unique mais plusieurs travaux sur la décomposition de Tucker imposent des contraintes sur les matrices facteurs pour assurer cette unicité [43, 44]. D'autres contraintes peuvent être imposées aussi comme la nonnégativité [45]. La décomposition de Tucker va nous servir à introduire la CPD qui en est un cas particulier pour lequel  $P=Q=R$  et le tenseur noyau  $\mathcal{G}$  est hyperdiagonal.

## 2.3 Décomposition Canonique Polyadique

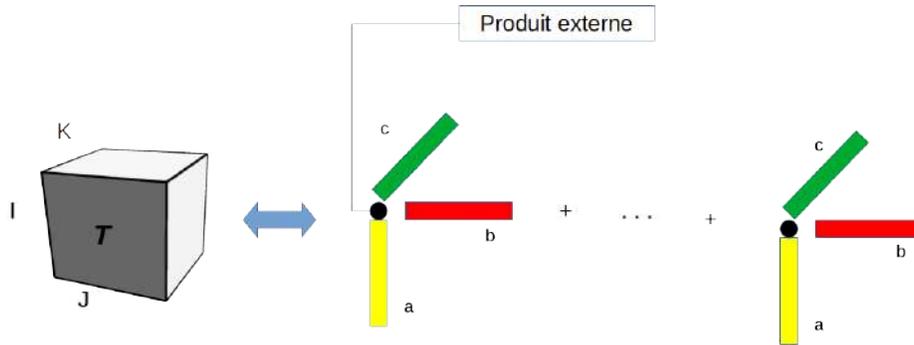


FIGURE 2.6 – Exemple de la CPD d'un tenseur d'ordre 3 vue comme une somme de tenseur de rang.

La Décomposition Polyadique de tenseur a été introduite par Hitchcock [46, 47] et Cattell [48, 49]. Au cours des dernières décennies, la CPD également connue en géoscience sous le nom de PARAFAC (analyse PARAllel FACTor) a été appliquée avec succès dans de nombreux domaines, notamment en psychométrie [16], en neurosciences [17], en chimiométrie [18], en vision par ordinateur [50], en analyse de réseaux sociaux [51], en data mining [19], en traitement d'images biomédicales [20], en imagerie hyperspectrale [21], en télécommunications [24], etc. Les données (matrices directes ou obtenues après transformation) issues de ces domaines peuvent être placées dans un tableau de données multidimensionnel (ou tenseur). Ensuite, la CPD permet de décomposer, généralement de manière approximative, ce tableau en facteurs interprétables par l'utilisateur (Figure 2.6). La CPD consiste à décomposer un tenseur comme une somme de tenseurs de rang 1 à l'aide de produit externe. Par exemple à l'ordre 3 :

$$\forall i, j, k, \mathcal{T}_{i,j,k} \simeq \widehat{\mathcal{T}}_{i,j,k}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{r=1}^R A_{ir} B_{jr} C_{kr} \quad (2.14)$$

Les matrices  $\mathbf{A} \in (I, R)$ ,  $\mathbf{B} \in (J, R)$ ,  $\mathbf{C} \in (K, R)$  sont appelées matrices facteurs de la décomposition. L'équation 2.14 peut se réécrire à l'aide des matrices de dépliement et du produit de Khatri-Rao :

$$\begin{aligned} \mathbf{T}_1 &= \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T, \\ \mathbf{T}_2 &= \mathbf{B}(\mathbf{A} \odot \mathbf{C})^T, \\ \mathbf{T}_3 &= \mathbf{C}(\mathbf{B} \odot \mathbf{A})^T. \end{aligned} \quad (2.15)$$

### 2.3.1 Rang du tenseur, rang de la CPD et rang physique du tenseur

$R$  défini dans 2.14 est le rang de la CPD. Lorsque les facteurs de la CPD ont une interprétation physique, il est important de distinguer deux valeurs particulières de  $R$  :

- La valeur maximale de  $R$  pour laquelle tous les facteurs ont une interprétation physique. Nous appelons cette valeur le rang physique  $\tilde{R}$  de  $\mathcal{T}$ .

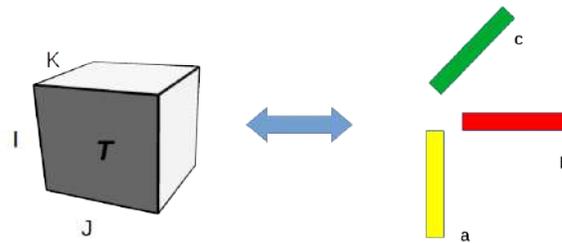


FIGURE 2.7 – Tenseur de rang 1.

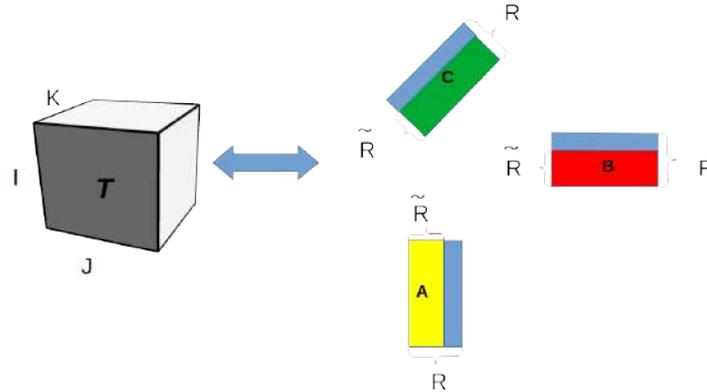
— La valeur minimale de  $R$  pour laquelle l'équation 2.14 est exacte (Figure 2.7). Cette valeur définit le rang du tenseur. En raison du bruit, il est généralement bien supérieur à  $\tilde{R}$ . En pratique, le rang physique est inconnu, ce qui pose un problème pour le calcul effectif de la CPD. Deux approches opposées sont traditionnellement utilisées pour traiter la question de l'estimation du rang du problème de la CPD, à savoir les méthodes d'estimation de rang a priori et la surfactorisation (overfactoring).

### 2.3.1.1 Estimation pratique du rang

L'estimation de rang consiste à estimer le rang de la CPD approprié avant la décomposition (rang physique du tenseur). Parmi ces méthodes, on peut citer le CORE CONSistency DIAgnostic (CORCONDIA) [52], split-half validation [53] ou récemment AutoTen [54]. Ces méthodes sont les plus utilisées en spectroscopie de fluorescence mais il existe d'autres méthodes. Dans [55], les auteurs proposent une méthode basée sur les plus grandes valeurs propres avec  $r = 1 \dots R$  via HOSVD appelés R-Dimensional Exponential Fitting Test (R-D EFT). Aussi, dans [56, 57] les auteurs utilisent un critère de probabilité dont le principe est également basé sur les valeurs propres pour estimer le rang. Dans [58], les auteurs combinent l'erreur de reconstruction entre le tenseur du départ et le tenseur à reconstruire avec le CORCONDIA. Automatic Relevance Determination (ARD) [59] résout le problème en considérant une distribution a priori qui peut être gaussienne ou laplacienne pour les matrices facteurs et en initialisant une borne supérieure du rang. En plus des méthodes d'estimation de rang citées ci-dessus, nous pouvons ajouter des méthodes d'estimation de rang pour les matrices. Ces méthodes sont basées sur des critères classiques de la théorie de l'information, telles que Akaike Information Criterion (AIC) [60], Bayesian Information Criterion (BIC) [61] et Minimum Description Length (MDL) [62]. Ces méthodes ont été généralisées et appliquées aux matrices dépliées d'un tenseur comme l'approche ND-MDL dans [55] ou dans [56] avec la méthode ND-MDL généralisée (GND-MDL). Même si les résultats obtenus des méthodes susmentionnées sont prometteurs, ces approches ne permettent pas toujours de trancher clairement entre plusieurs valeurs de rang possibles.

### 2.3.1.2 La surestimation du rang

A l'inverse, la surfactorisation peut être considérée comme une méthode d'estimation du rang *a posteriori* car la valeur de rang appropriée est déduite après calcul effectif de la CPD. Dans



**FIGURE 2.8 – Illustration de la CPD avec surestimation du rang.**  $R$  correspond au rang de la CPD incluant des colonnes nulles au moins sur l'une des 3 matrices.

cette approche, une valeur surestimée du rang est choisie au départ et les algorithmes CPD sont conçus pour produire des facteurs supplémentaires avec des contributions nulles (Figure 2.8). De tels algorithmes recourent généralement à une contrainte de parcimonie sur les facteurs [63, 64, 65, 66]. En plus de la contrainte de parcimonie, il a été récemment montré que des algorithmes “directs” basés sur la décomposition en valeurs singulières (SVD) et la diagonalisation conjointe permettaient de surfactoriser le rang [67].

### 2.3.2 Unicité de la décomposition

La combinaison de tenseurs de rang 1 dont la somme est égale au tenseur que l'on cherche à décomposer définit l'unicité de la CPD sans tenir compte de la permutation et de l'échelle. On parle alors d'unicité essentielle. Nous disposons d'une condition suffisante d'unicité essentielle simple basée sur la notion de rang de Kruskal.

**Le rang de Kruskal :** Le rang de Kruskal d'une matrice  $\mathbf{A}$ , noté  $R_k(\mathbf{A})$ , est le nombre maximal tel que chaque ensemble de  $R_k(\mathbf{A})$  colonnes de  $\mathbf{A}$  soit linéairement indépendant. Cet outil mathématique fut présenté dans [68] puis nommé rang de Kruskal dans [69].

**Condition suffisante d'unicité d'un tenseur d'ordre 3 et d'un tenseur d'ordre  $N$  :** Une condition d'unicité suffisante pour la CPD d'un tenseur d'ordre 3 est définie par :

$$2R + 2 \leq R_k(\mathbf{A}) + R_k(\mathbf{B}) + R_k(\mathbf{C}) \quad (2.16)$$

N. D. Sidiropoulos et R. Bro dans [70] ont généralisé les travaux de Kruskal à l'ordre quelconque  $N$  et la condition suffisante d'unicité essentielle s'écrit alors :

$$2R + (N - 1) \leq \sum_{n=1}^N R_k(\mathbf{A}_n) \quad (2.17)$$

D'autres conditions moins restrictives existent mais celle-ci est généralement suffisante pour notre application.

### 2.3.3 Algorithme de calcul de la CPD : état de l'art

Le problème de la CPD est un problème inverse dont le but est d'estimer les matrices facteurs à partir du tenseur  $\mathcal{T}$  initial. Dans la littérature, il y a beaucoup de façons de résoudre ce problème. Nous pouvons énumérer trois grandes familles d'algorithmes qui sont : i) les méthodes directes ou semi-algébriques, ii) les méthodes de block coordinate ou alternées, iii) les méthodes de descente. Chaque famille possède des avantages et des inconvénients ainsi qu'une complexité différente. Pour chaque famille, nous allons donner un exemple d'algorithme avec comme critère d'arrêt le nombre total d'itérations  $iterMax$  combiné à tout critère d'arrêt classique d'un algorithme d'optimisation. Les algorithmes présentés pour la troisième famille seront importants ici car nous l'utilisons pour optimiser nos différentes fonctions dans le chapitre suivant.

La CPD d'un tenseur  $\mathcal{T}$  d'ordre 3, en supposant un rang  $R$  donné, se calcule généralement en minimisant l'écart entre ce tenseur  $\mathcal{T}$  qu'on cherche à décomposer et son approximation  $\widehat{\mathcal{T}}(\mathbf{A}, \mathbf{B}, \mathbf{C})$  donnée par les facteurs que l'on cherche. Elle est généralement mesurée par la norme de Frobenius et s'écrit par :

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \left\{ E(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{A}, \mathbf{B}, \mathbf{C})\|_F^2 \right\} \quad (2.18)$$

---

#### Algorithme 1 Algorithme DIAG à l'ordre 3

---

Entrées :  $\mathcal{T}, R$

Une matrice de dépliement du le tenseur  $\mathcal{T}$  ;

Calculer la SVD  $USV^T$  de  $\mathcal{T}$ , tronquée au rang  $R$

Diviser la matrice  $SV^T$  en  $K$  blocs de taille  $(R \times J)$

**for**  $k_1 = 1$  to  $K$  **do**

**for**  $k_2 = 1$  to  $K$  avec  $i_2 \neq i_1$  **do**

    Calculer  $\Theta^{(k_1, k_2)} = (\mathbf{\Gamma}^{(i_1)})^{-1} \mathbf{\Gamma}^{(i_2)}$

**end for**

**end for**

Calculer la matrice de vecteur  $\mathbf{M}$  commune à l'ensemble des matrices  $\Theta^{(k_1, k_2)}$  ou d'un sous-ensemble en utilisant un algorithme de diagonalisation conjointe par similitude.

En déduire  $\mathbf{A} = \mathbf{U}\mathbf{M}$  et  $\mathbf{Y} = \mathbf{M}^{-1}\mathbf{S}\mathbf{V}^T$

**for**  $r = 1$  to  $R$  **do**

  Construire  $\mathcal{Y}_r$  à partir de la  $r$ -ème colonne de  $\mathbf{Y}$

  Déduire les colonnes  $r$  de  $\mathbf{B}, \mathbf{C}$  de la HOSVD de rang 1 de  $\mathcal{Y}_r$

**end for**

---

**Famille directe ou semi-algébrique :** Le principe de ce type de méthode est de réduire une ou plusieurs des dimensions du tenseur initial ce qui permet généralement de réécrire le problème comme un problème de diagonalisation conjointe de matrices [67]. L'intérêt principal est de réduire la complexité numérique du processus itératif et en général de faciliter la convergence. Voici par exemple quelques algorithmes disponibles :

- Direct Trilinear Decomposition (DTLD) [71]
- SEmi-algebraic framework for approximate CP decompositions via SImultaneous matrix diagonalization (SECSI) [72]
- DIrect AlGorithm (DIAG) [73, 74]

**Famille de block coordinate ou alternées :** Cette approche permet d’optimiser séparément des sous-ensembles de variables de manière séquentielle à chaque itération. En pratique, le principe est de fixer les matrices facteurs à optimiser sauf une, à une itération donnée, qui sera mise à jour. Comme exemples, nous pouvons citer :

- Alternating Least Square (ALS) [18, 69, 75]
- Alternating Direction Method of Multipliers (ADMoM) [76]
- Nonlinear Least Square (NLS) [77, 78]
- Alternating Nonnegative Least Squares (ANLS) [79]
- Projected and Compressed ANLS (ProCo) [80]

Dans cette approche, l’ALS présenté dans l’algorithme 2 est l’algorithme le plus utilisé et sert de base aux autres algorithmes qui permettent d’incorporer des contraintes de manière efficace (non-négativité, parcimonie, entre autres). ALS permet de réécrire le problème de l’équation 2.18 qui est un problème non-convexe en trois problèmes convexe avec une solution analytique (2.15). ALS est très sensible à l’initialisation des matrices facteurs au début de l’algorithme. Il est également peu efficace lorsqu’une ou plusieurs matrices facteurs ont des colonnes colinéaires ou quasi-colinéaires [74, 81, 82]. Dans le but d’accélérer l’ALS d’autres algorithmes ont vu le jour comme ELS [83, 24].

---

### Algorithme 2 ALS

---

*Entrées :*  $\mathcal{T}, R$

*initialisation de*  $\mathbf{A}, \mathbf{B}, \mathbf{C}$

$iter \leftarrow 0$

**while**  $iter < iterMax$  ou un critère d’arrêt non vérifié **do**

$$\mathbf{A} = \mathbf{T}_1 ((\mathbf{C} \odot \mathbf{B})^+)^T$$

$$\mathbf{B} = \mathbf{T}_2 ((\mathbf{A} \odot \mathbf{C})^+)^T$$

$$\mathbf{C} = \mathbf{T}_3 ((\mathbf{B} \odot \mathbf{A})^+)^T$$

$iter \leftarrow iter + 1$

**end while**

*Sortie :*  $\mathbf{A}, \mathbf{B}, \mathbf{C}$

---

**Famille de descente :** Le principe ici est d’utiliser des algorithmes de descente classiques pour optimiser conjointement les 3 matrices facteurs. Ces méthodes bien connues sont basées sur le gradient et/ou le Jacobien [33, 84, 85, 86, 87, 88, 89] de la fonction à minimiser. Ces méthodes permettent également d’appliquer différentes contraintes de manière simple. En voici quelques-unes :

- Descente de gradient
- Gradient conjugué
- Gauss-Newton
- Levenberg-Marquardt
- BFGS

Nous résumons dans l’algorithme 3, un exemple d’optimisation de cette famille : la descente de gradient.

Le problème de CPD peut se réécrire en intégrant différentes contraintes cela permet à l’algorithme choisi d’avoir la meilleure convergence et d’avoir si nécessaire une compatibilité avec la réalité physique.

---

**Algorithme 3** Descente de Gradient
 

---

Entrées :  $\mathcal{T}, R, \alpha$

initialisation de  $\mathbf{A}, \mathbf{B}, \mathbf{C}$

$iter \leftarrow 0$

**while**  $iter < iterMax$  ou un critère d'arrêt non vérifié **do**

$$\mathbf{X} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{bmatrix}$$

$$\mathbf{X} = \mathbf{X} - \alpha \begin{bmatrix} \frac{\partial E}{\partial \mathbf{A}} \\ \frac{\partial E}{\partial \mathbf{B}} \\ \frac{\partial E}{\partial \mathbf{C}} \end{bmatrix}$$

avec

$$\frac{\partial E}{\partial \mathbf{A}} = -(\mathbf{T}_1 - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\top)(\mathbf{C} \odot \mathbf{B})$$

$$\frac{\partial E}{\partial \mathbf{B}} = -(\mathbf{T}_2 - \mathbf{B}(\mathbf{A} \odot \mathbf{C})^\top)(\mathbf{A} \odot \mathbf{C})$$

$$\frac{\partial E}{\partial \mathbf{C}} = -(\mathbf{T}_3 - \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\top)(\mathbf{B} \odot \mathbf{A})$$

$iter \leftarrow iter + 1$

**end while**

Sortie :  $\mathbf{A}, \mathbf{B}, \mathbf{C}$

---

### 2.3.3.1 La CPD avec différentes contraintes

Il est utile de mentionner que la CPD peut incorporer différentes contraintes en fonction du champs d'où proviennent les données. Il peut s'agir par exemple des contraintes de non-négativité, d'orthogonalité ou de parcimonie pour ne citer que les plus couramment utilisées.

**Contrainte de non-négativité** La contrainte de non-négativité est imposée dans la CPD lorsque les données à traiter sont liées à des grandeurs physiques non-négatives. Dans ce sens, nous pouvons citer le domaine de l'imagerie [21, 90, 91] où la non-négativité est présente car on sait que les signaux optiques sont positifs ou nuls. C'est aussi le cas de la spectroscopie de fluorescence où les données issues du capteur sont non-négatives [18, 92, 93, 65]. Cette contrainte permet une meilleure convergence surtout lorsque les colonnes des matrices facteurs sont colinéaires. La contrainte de non-négativité peut être imposée sur les matrices facteurs lors de l'optimisation en projetant les valeurs de chaque matrice facteur sur  $\mathbb{R}^+$  [94]. Cette méthode d'imposition de la non-négativité montre parfois ses limites dans certains cas à cause du changement d'échelle. Dans [95, 96], les auteurs proposent de remédier à ce problème en faisant intervenir deux facteurs,  $\epsilon$  et  $\theta \in \{-1; 1\}$  qui multiplient chaque matrice facteur. La non-négativité peut aussi être imposée en choisissant une méthode d'optimisation [76, 63, 64] ou encore en utilisant un changement de variable exponentiel [35] ou carré [93].

**Contrainte de parcimonie** Concernant la contrainte de parcimonie, elle est généralement utilisée pour les tenseurs parcimonieux ou pour assurer la surestimation du rang, comme mentionné précédemment. Cette contrainte est imposée en fonction du but que l'on recherche et dans notre cas, c'est la surestimation du rang. La contrainte peut être appliquée sur les matrices facteurs sous la forme d'un terme de régularisation :

$$\min_x \underbrace{F(x)}_{\text{fonction}} + \underbrace{R(x)}_{\text{régularisation}} \quad (2.19)$$

$F(x)$  correspond ici à  $\|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{A}, \mathbf{B}, \mathbf{C})\|_F^2$  et est la partie d'attachement.  $R(x)$  est la partie régularisation sur la matrice  $\mathbf{A}$ ,  $\mathbf{B}$  et  $\mathbf{C}$ .

Dans la littérature, la norme  $L_{1,1}$  est la plus utilisée comme dans [63, 64, 65, 97, 98] et elle peut être définie en fonction de l'équation 2.19 comme suit :

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{A}, \mathbf{B}, \mathbf{C})\|_F^2 + \|\mathbf{A}\|_{1,1} + \|\mathbf{B}\|_{1,1} + \|\mathbf{C}\|_{1,1} \quad (2.20)$$

La norme  $L_{1,1}$  est classiquement utilisée par rapport à la norme  $L_{0,0}$  pour promouvoir la parcimonie car elle est plus simple à manipuler que la norme  $L_{0,0}$ . Dans l'optique de minimiser le rang d'une matrice, nous pouvons parler de la norme nucléaire qui permet d'imposer un faible rang [99, 100, 101] et peut être définie en fonction de l'équation 2.19 comme suit :

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{A}, \mathbf{B}, \mathbf{C})\|_F^2 + \|\mathbf{A}\|_* + \|\mathbf{B}\|_* + \|\mathbf{C}\|_* \quad (2.21)$$

Dans [102, 103], les auteurs montrent que la norme mixte fournit une meilleure enveloppe convexe de la fonction rang que la norme nucléaire. Elle permet donc de minimiser le rang d'une matrice. Dans la littérature, la norme mixte est de plus en plus utilisée comme dans [104, 105, 106, 107]. Elle peut être définie en fonction de l'équation 2.19 comme suit :

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{A}, \mathbf{B}, \mathbf{C})\|_F^2 + \|\mathbf{A}\|_{2,1} + \|\mathbf{B}\|_{2,1} + \|\mathbf{C}\|_{2,1} + \|\mathbf{A}\|_{1,2} + \|\mathbf{B}\|_{1,2} + \|\mathbf{C}\|_{1,2} \quad (2.22)$$

**Contrainte d'orthogonalité** La contrainte d'orthogonalité sur les facteurs est également utile dans certaines applications [108, 45]. Il est possible d'appliquer la contrainte d'orthogonalité sur les matrices facteurs en utilisant l'équation 2.19. Si, par exemple,  $\mathbf{A}$  doit être orthogonale alors :  $\mathbf{I}_A = \mathbf{A}^T \mathbf{A}$ . Nous cherchons donc à minimiser :

$$\min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{A}, \mathbf{B}, \mathbf{C})\|_F^2 + \|\mathbf{A}^T \mathbf{A} - \mathbf{I}_A\|_F + \|\mathbf{B}^T \mathbf{B} - \mathbf{I}_B\|_F + \|\mathbf{C}^T \mathbf{C} - \mathbf{I}_C\|_F \quad (2.23)$$

D'autre part, il est possible d'imposer cette contrainte en faisant une projection lors du processus d'optimisation [45]. Cette projection est une façon de réécrire la propriété de l'orthogonalité  $\mathbf{I}_A = \mathbf{A}^T \mathbf{A}$  par l'approximation de Stiefel. Cela consiste à projeter sur l'espace tangent de  $St(I, J)$  à une itération donnée la rétraction sur  $St(I, J)$ . La projection de  $\mathbf{A}$  sur l'espace tangent à  $St(p, q)$  au point  $\mathbf{X}$  est donnée par [108] :

$$(\mathbf{I}_p - \mathbf{X}\mathbf{X}^T)\mathbf{A} + \frac{1}{2}\mathbf{X}(\mathbf{X}^T \mathbf{A} - \mathbf{A}^T \mathbf{X}) \quad (2.24)$$

## 2.4 Décomposition de tenseurs en ligne

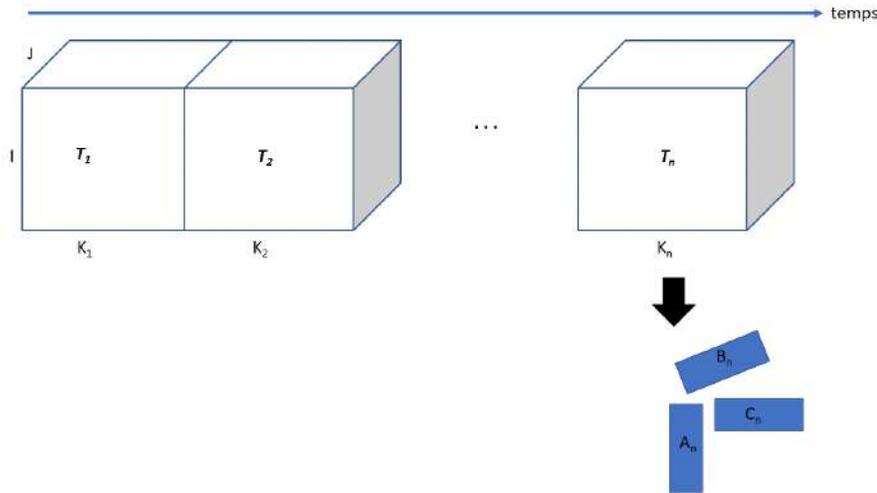


FIGURE 2.9 – Illustration de la CPD d’un tenseur d’ordre 3 en ligne.

Une décomposition tensorielle en ligne est calculée sur des données qui augmentent au fil du temps (Figure 2.9) [109, 110, 111]. De nos jours, les systèmes d’acquisition deviennent plus rapides ce qui peut créer rapidement une grande quantité de données au fil du temps. Dans de tels environnements dynamiques, un tenseur de données peut être modifié de plusieurs manières avant d’appliquer la CPD. Dans le cas de la CPD en ligne telle que nous la présentons ici, le tenseur  $\mathcal{T}$  peut être vu comme une collection de **sous-tenseurs**  $\mathcal{T}_n$  d’ordre 3, de taille  $(I, J, K_n)$  enregistrés à divers intervalles de temps consécutifs comme le montre la Figure 2.9. La décomposition tensorielle en ligne présente plusieurs difficultés. On a la variation de rang physique  $\tilde{R}_n$  des sous-tenseurs au cours du temps qui peut arriver avec certaines données comme les données issues de mesures (e-commerce) [51] ou des données environnementales [112].

Dans cette thèse, nous ne voulons pas traiter le tenseur en entier mais uniquement une portion obtenue à un instant précis et nous supposons que le rang  $\tilde{R}_n$  de la décomposition peut varier au cours du temps. Le problème de la CPD en ligne avec variation de rang a été bien posé dans [51] : il consiste à résoudre l’équation (2.18) pour chaque sous-tenseur  $\mathcal{T}_n$ . De plus, ici, on considère que  $\tilde{R}_n$  est inconnu. Une augmentation très rapide des données nécessite des méthodes de décomposition du tenseur capables de fournir des performances en temps réel ou quasi réel [113]. Le but de la CPD en ligne est alors de mettre à jour, à chaque nouvel intervalle de temps, les matrices facteurs précédemment estimées, sans effectuer la CPD de l’ensemble du tenseur et tout en tenant compte d’éventuelles variation du rang. En effet, des performances temps réel sont requises c’est-à-dire que le temps de traitement des données doit être inférieur au temps d’acquisition des données. D’autre part, on peut raisonnablement supposer que certaines matrices consécutives partagent plusieurs facteurs communs en particulier quand nous avons des chevauchements entre sous-tenseurs. Ainsi, intuitivement, nous ne voudrions peut-être pas décomposer chaque nouveau sous-tenseur indépendamment mais privilégier un chevauchement afin d’améliorer l’estimation de la matrice facteur.

**Etat de l'art sur les algorithmes de décomposition tensorielle en ligne :** A notre connaissance, peu d'approches ont été proposées pour calculer la CPD en ligne. Dans [114], les auteurs utilisent le filtre de Kalman [115] pour prédire la sortie de leur décomposition à l'instant  $t + 1$  à partir des données des temps antérieurs. Aussi dans [116], les auteurs utilisent une approximation de rang faible avec un gradient stochastique pour résoudre le problème de la CPD en ligne. Dans [117], les auteurs ont introduit deux algorithmes adaptatifs Simultaneous Diagonalization Tracking (SDT) et Recursive Least Squares Tracking (RLST) : SDT suit de manière incrémentielle la SVD du tenseur déplié. RLST met à jour récursivement les facteurs de décomposition en minimisant l'erreur quadratique moyenne. On peut également citer l'algorithme de factorisation de tenseur appelé (GridTF) [118] dont le principe est de partitionner le grand tenseur en plusieurs petites grilles et d'utiliser en parallèle l'algorithme ALS pour décomposer les grilles. Plus récemment, les auteurs dans [119] proposent OnlineCP qui est un calcul récursif des matrices factorielles basé sur ALS. Dans [78], les auteurs ont proposé une mise à jour non linéaire des moindres carrés de la CPD basée également sur un algorithme NLS pour mettre à jour la dernière matrice facteur ( $\mathbf{C}$ ) de manière ALS. Dans [120], les auteurs ont introduit SamBaTen et ils supposent qu'ils reçoivent des tranches sur le troisième mode, qui à leur tour doivent ajouter de nouvelles lignes à la matrice  $\mathbf{C}$  et mettent à jour cette matrice avec une matrice d'échelle diagonale et matrice de permutation. Toujours en [111], les auteurs ont développé une approche nommée OCTen basée sur la méthode de décomposition incrémentale parallèle.

Cependant, toutes ces approches supposent que le rang est connu et ne varie pas dans le temps. Récemment dans [51], les auteurs ont introduit une approche appelée SeekAndDestroy où les sous-tenseurs sont traités indépendamment en utilisant la méthode de détermination du rang par l'algorithme AutoTen [54] avant chaque CPD. Si le nouveau rang estimé est supérieur au précédent, les matrices facteurs sont mises à jour pour tenir compte de l'apparition d'un ou des nouveaux facteurs.

Plusieurs études ont été menées sur la décomposition de Tucker en ligne incorporant des techniques en ligne comme la mise à jour incrémentale des matrices de covariance [109, 110] ou SVD incrémentale [121, 122, 116]. Dans [45], les auteurs ont introduit une décomposition de Tucker en ligne avec un rang fixe. Plus récemment, on peut citer la Block Term Decomposition Least Square (BTD-IRLS) [123], une approche basée sur une autre forme de décomposition de Tucker. Enfin, dans [124], les auteurs proposent une approche d'apprentissage supervisé (deep learning) appliquée à la CPD en ligne avec rang fixe.

## 2.4.1 Limitations des algorithmes CPD

La plupart des algorithmes mentionnés ci-dessus supposent que le rang (physique) de la décomposition est connu. Cependant, dans de nombreuses situations pratiques, cette valeur est inconnue et nous ne pouvons supposer qu'un ordre de grandeur alors qu'un choix incorrect de cette valeur peut affecter gravement les estimations des facteurs [62, 125, 67]. En plus de ce problème de rang que l'on rencontre, la corrélation ou la quasi-colinéarité [96] des facteurs peut rendre la décomposition problématique. Bien que les appareils d'acquisition soient performants de nos jours, cela n'empêche pas le bruit de se mélanger aux données. Pour répondre à ces problèmes, il advient de choisir une famille d'algorithmes en fonction de certains critères. Parmi les critères importants pour notre problème, nous pouvons citer :

- \* Le coût du calcul de la méthode ;
- \* La précision de l'estimation des matrices facteurs à partir d'un tenseur bruité ;

#### 48 *Décomposition Canonique Polyadique de tenseur et application à la spectroscopie de fluorescence*

- \* La robustesse vis-à-vis de facteurs fortement corrélés ;
- \* La robustesse vis-à-vis de surfactorisation (ou l'overfactoring) ;
- \* La prise en compte de certaines contraintes comme la nonnégativité ou la parcimonie.

Certains signaux optiques comme les signaux de fluorescence suivent le modèle sur lequel est basée la CPD. Nous donnons les détails sur ce lien dans la section suivante.

## 2.5 Lien entre la CPD et la spectroscopie de fluorescence

Cette section présente la spectroscopie de fluorescence qui est une des techniques généralement utilisées pour la surveillance environnementale. Dans ce cas particulier, cette technique est employée pour identifier les espèces chimiques présentes dans le milieu ainsi que leur évolution de concentration relative pour la surveillance environnementale. Cette section est inspirée de [35, 2, 36].

### 2.5.1 Phénomène de fluorescence

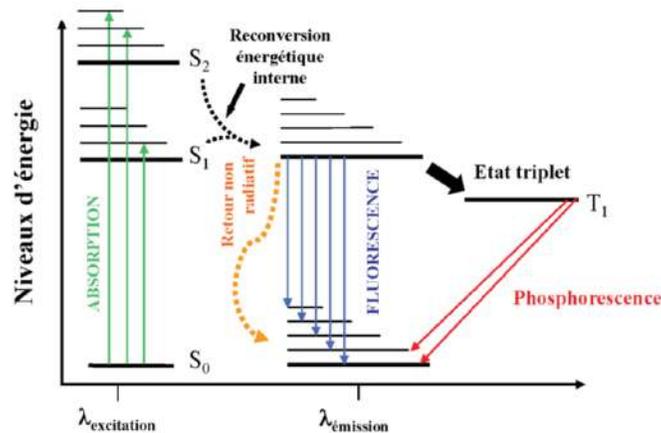


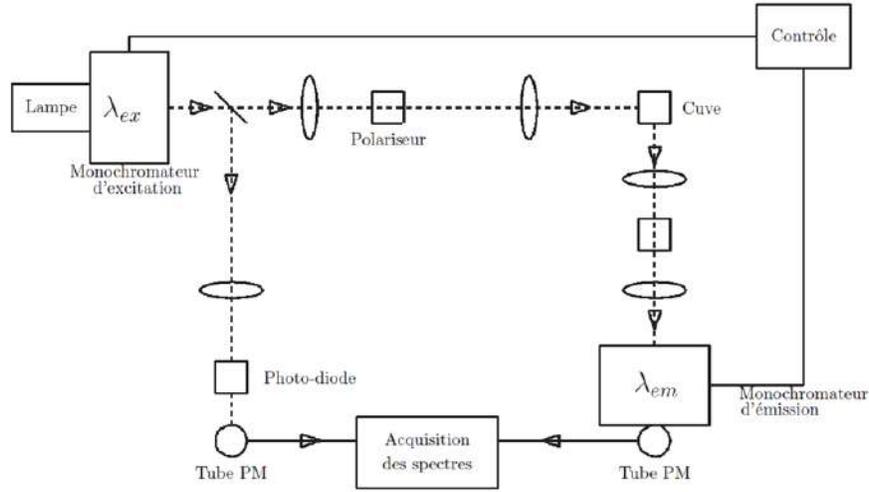
FIGURE 2.10 – Diagramme de Jablonski (image tirée de [1]).

Lors de l'absorption d'un photon, la molécule passe de l'état énergétique fondamental  $S_0$  à un état électronique excité de l'état singulet  $S_1^*$  ou  $S_2^*$  (Figure 2.10). Par conversion énergétique interne, la molécule atteint le niveau vibrationnel le plus bas de l'état  $S_1^*$  sans émission de photon (transition énergétique non-radiative). Dans les liquides, lors de la transition énergétique du niveau  $S_1^*$  vers l'état fondamental  $S_0$  par voie radiative, la molécule émet un photon d'énergie plus faible que le photon initialement absorbé : il s'agit du phénomène de fluorescence qui est donc une transition radiative. Les longueurs d'onde émises par fluorescence sont donc plus grandes que les longueurs d'onde ayant provoqué le phénomène d'absorption, cette différence est appelée décalage de Stokes. Il existe une autre possibilité de désexcitation qui fait intervenir l'état triplet  $T_1^*$  : il s'agit du phénomène de phosphorescence qui est caractérisé par un temps de relaxation plus long que celui de la fluorescence mais rarement présent dans les liquides en raison de désactivation par chocs moléculaires avec les molécules du solvant.

### 2.5.2 Spectroscopie de fluorescence

Dans le cas d'un échantillon liquide contenant un fluorophore dont l'absorbance est suffisamment faible, l'intensité de fluorescence pour un couple de longueurs d'onde d'excitation et d'émission ( $\lambda_{ex}$ ,  $\lambda_{em}$ ) est donnée par la relation suivante :

$$I(\lambda_{ex}, \lambda_{em}) = \kappa_1 I_0(\lambda_{ex}) \beta(\lambda_{ex}) \eta(\lambda_{em}) \phi_{FC} \quad (2.25)$$



**FIGURE 2.11 – Schéma du spectrofluorimètre.**

avec  $I_0$  l'intensité incidente,  $\beta(\lambda_{ex})$  le spectre d'excitation,  $\eta(\lambda_{em})$  le spectre d'émission,  $\phi_F$  le rendement quantique de fluorescence,  $\kappa_1$  la constante d'appareil mesurant la fluorescence et  $c$  la concentration du fluorophore.

Dans le cas plus général d'un échantillon liquide contenant  $\tilde{R}$  fluorophores de rendement quantique de fluorescence  $\phi_{F,r}$ , la relation précédente devient :

$$I(\lambda_{ex}, \lambda_{em}) = \kappa_1 I_0(\lambda_{ex}) \sum_{r=1}^{\tilde{R}} \beta_r(\lambda_{ex}) \eta_r(\lambda_{em}) \phi_{F,r} c_r \quad (2.26)$$

Pour un ensemble de  $K$  mélanges de  $\tilde{R}$  fluorophores, l'intensité de fluorescence est exprimée par :

$$I(\lambda_{ex}, \lambda_{em}, k) = \kappa_1 I_0(\lambda_{ex}) \sum_{r=1}^{\tilde{R}} \beta_r(\lambda_{ex}) \eta_r(\lambda_{em}) \phi_{F,r} c_{k,r} \quad (2.27)$$

Généralement, l'intensité de fluorescence émise par un échantillon liquide est mesurée à l'aide d'un spectrofluorimètre qui permet de sélectionner les longueurs d'onde d'excitation  $\lambda_{ex}$  et d'émission  $\lambda_{em}$ . Afin de tenir compte de la variation de  $I_0(\lambda_{ex})$  pendant la phase de mesure, la grandeur mesurée est la suivante :

$$I_F(\lambda_{ex}, \lambda_{em}, k) = \frac{I(\lambda_{ex}, \lambda_{em}, k)}{\kappa_2 I_0(\lambda_{ex})} = \sum_{r=1}^{\tilde{R}} \beta_r(\lambda_{ex}) \eta_r(\lambda_{em}) C_{k,r} \quad (2.28)$$

où  $\kappa_2$  est la constante d'appareil mesurant l'intensité lumineuse en excitation  $I_0(\lambda_{ex})$ .

L'équation 2.28 peut donc se réécrire de la façon suivante :

$$I_F(\lambda_{ex}, \lambda_{em}, k) = \kappa I_0 \sum_{r=1}^{\tilde{R}} \beta_r(\lambda_{ex}) \eta_r(\lambda_{em}) c_{k,r} \quad (2.29)$$

avec l'intensité  $I_0$  qui peut être considérée comme constante quelque soit la longueur d'onde d'excitation  $\lambda_{ex}$  et  $\kappa$  la constante d'appareil de l'ensemble du spectrofluorimètre.

Cependant, une partie importante des photons envoyés sur un échantillon liquide est inévitablement diffusée par le solvant (habituellement, l'eau). Pour une longueur d'onde d'excitation donnée, cela se traduit par deux pics de diffusion qui viennent s'ajouter au spectre d'émission de fluorescence. Le premier pic est dû à une diffusion élastique, le photon diffusé possède la même longueur d'onde que le photon incident : c'est la diffusion Rayleigh. Le second pic, d'intensité plus faible que le précédent, a pour origine la diffusion Raman, processus au cours duquel le photon exciteur perd une partie de son énergie ( $E_{vib}$ ) en interagissant avec une molécule de solvant. Le photon diffusé a donc une longueur d'onde  $\lambda_{em}$  plus grande que celle du photon incident (équation 2.30) :

$$\frac{1}{\lambda_{em}} = \frac{1}{\lambda_{ex}} - \frac{E_{vib}}{hc} \quad (2.30)$$

avec  $h$  la constante de Planck et  $c$  la vitesse de la lumière dans le vide.

### 2.5.3 MEEF et tenseur de fluorescence

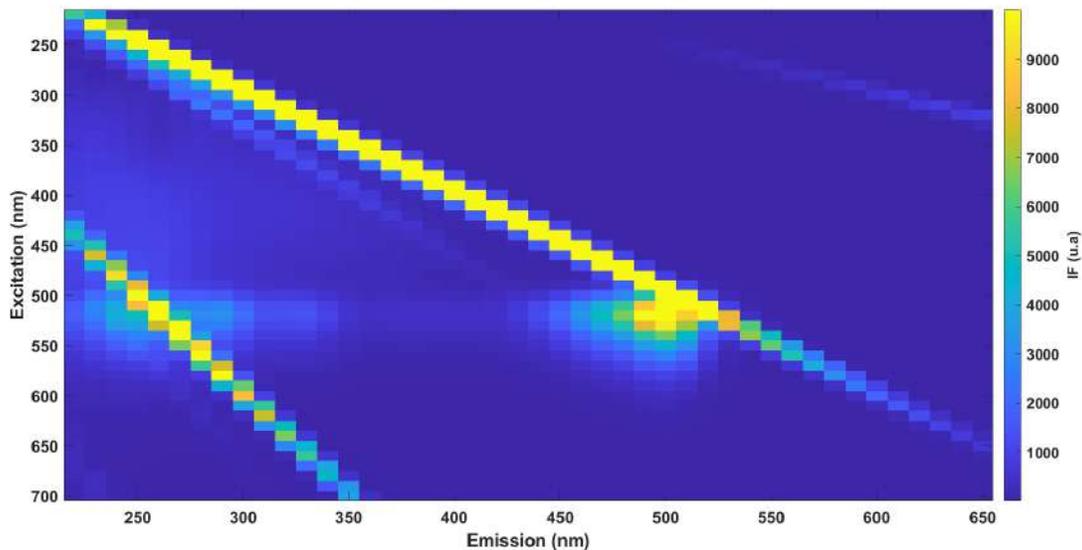


FIGURE 2.12 – MEEF avec le phénomène de diffusion.

Pour produire la Matrice d'Excitation et d'Emission de Fluorescence (MEEF) d'un échantillon, une longueur d'onde d'excitation  $\lambda_{ex}$  est sélectionnée par le premier monochromateur du spectrofluorimètre avant de balayer toutes les longueurs d'onde d'émission  $\lambda_{em}$  à l'aide du second monochromateur (Figure 2.11), ce qui permet d'obtenir un spectre d'émission : l'ensemble des spectres d'émission associés aux différentes longueurs d'onde d'excitation correspond à une MEEF dont un exemple est représenté par la Figure 2.12. Un tenseur de fluorescence (Figure 2.13) correspond au regroupement de plusieurs MEEF.

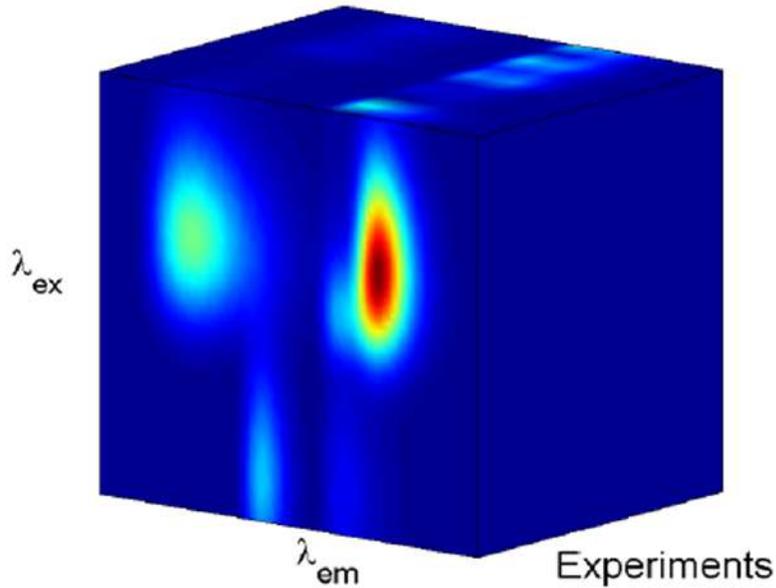


FIGURE 2.13 – Exemple de tenseur de troisième ordre. Intensité de fluorescence fonction de 3 variables : longueurs d’onde d’excitation et d’émission et des échantillons.

#### 2.5.4 Phénomène de diffusion

Comme cela a été évoqué précédemment, chaque MEEF contient une superposition de signaux de fluorescence et de pics de diffusion Rayleigh et Raman (Figure 2.14). Les réseaux de diffraction qui composent les monochromateurs des spectrofluorimètres produisent des spectres d’ordres plus ou moins élevés. Habituellement, on peut observer sur les MEEFs mesurées les spectres de fluorescence et les pics de diffusion pour l’ordre un mais aussi les spectres et les pics de diffusion pour les ordres plus élevés.

La présence inévitable des pics de diffusion engendre une double difficulté de traitement numérique ultérieur. Premièrement, ils peuvent masquer en partie un spectre de fluorescence d’intérêt et donc rendre plus délicate l’identification d’un fluorophore. Deuxièmement, les pics de diffusion Rayleigh et Raman ne suivent pas le modèle trilineaire de l’équation 2.29. Il est donc nécessaire de soigneusement les éliminer numériquement avant toute décomposition de tenseur de fluorescence. Il existe plusieurs méthodes possibles pour supprimer les pics de diffusion mais la plus utilisée est la méthode de Zepp [126] qui est basée sur l’identification des régions des différents pics de diffusions et de l’interpolation bilinéaire de chaque MEEF sur ces régions en utilisant la triangulation de Delaunay. Une autre méthode utilise une technique de filtrage des images basée sur la morphologie mathématique [93]. Une décomposition alternative à la CPD prenant en compte ces effets a été récemment proposée [127]. Cependant, l’algorithme associé est assez lent et n’a pas été testé sur ce type de données. De même, on ignore son comportement lorsque le nombre de composés fluorescents est mal estimé.

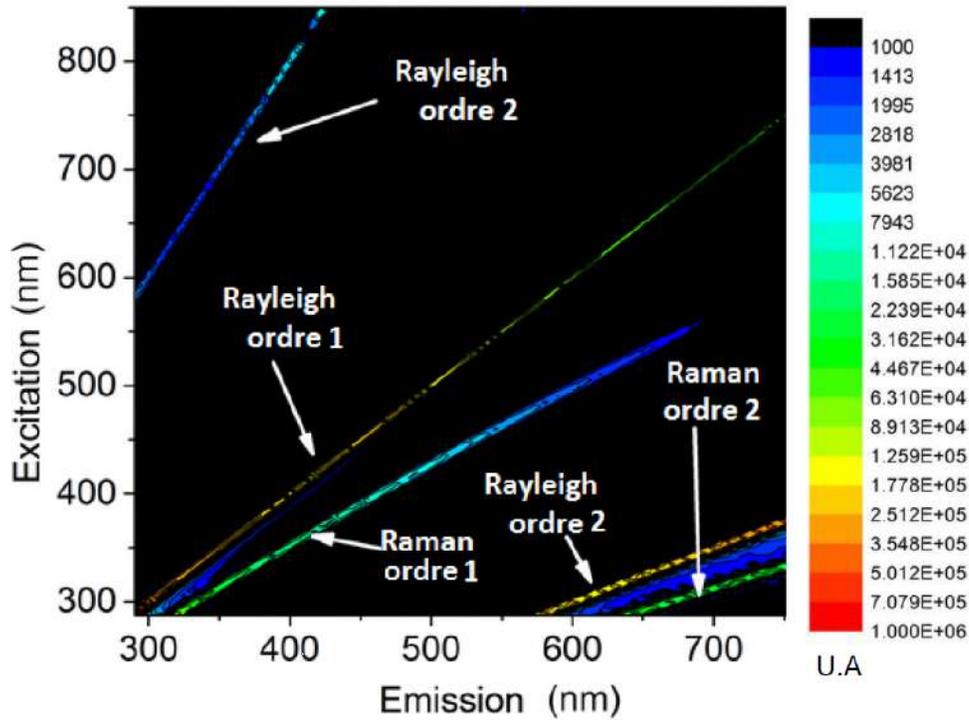


FIGURE 2.14 – MEEF illustrant l'emplacement des diffusions de Raman et Rayleigh dans un échantillon d'eau ultra pure (Image tirée de [2]).

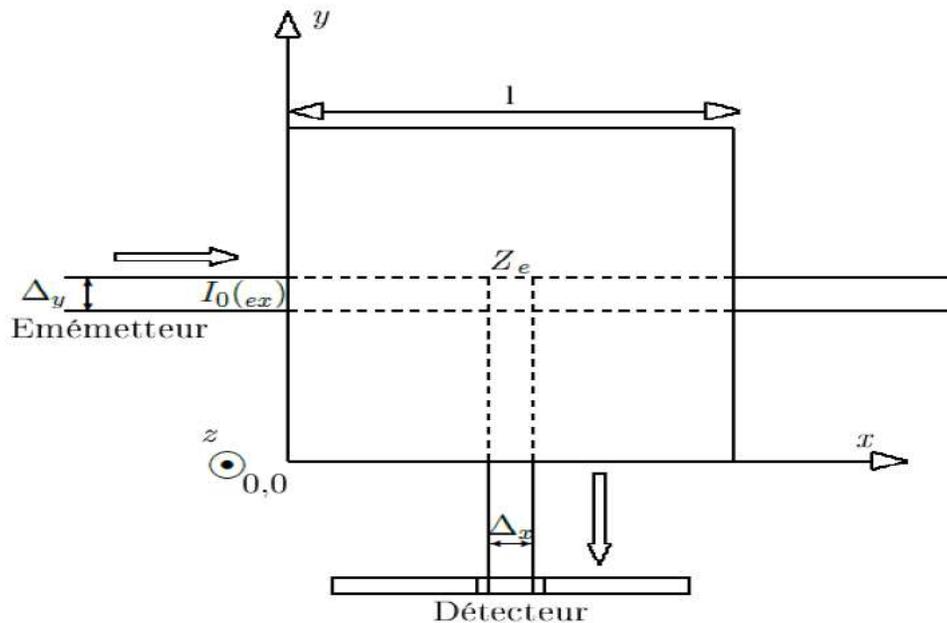


FIGURE 2.15 – Schéma de la vue de dessus de la cuve contenant l'échantillon à analyser . (Image tirée de [3]).

### 2.5.5 Effet d'écran

Toujours dans l'optique du prétraitement, en plus du phénomène de diffusion qui est présent, il peut y avoir également le phénomène de l'effet d'écran [128]. L'effet d'écran s'observe lorsque l'absorbance d'un échantillon liquide est trop élevée malgré les précautions d'usage. Cela se traduit par une forte atténuation de l'intensité (Figure 2.15) incidente en excitation (effet d'écran primaire) avant le volume de mesure éventuellement suivie par une absorption de la fluorescence par l'échantillon lui-même (effet d'écran secondaire).

Le modèle de fluorescence devient alors :

$$I(\lambda_{ex}, \lambda_{em}, k) = \underbrace{\left( \sum_{r=1}^{\tilde{R}} G_r C_{kr} \beta_r(\lambda_{ex}) \eta_r(\lambda_{em}) \right)}_{\text{partie linéaire}} \underbrace{\prod_{r=1}^R e^{-g(C_{kr} \beta_r(\lambda_{ex}) \eta_r(\lambda_{em}))}}_{\text{partie non-linéaire}} \quad (2.31)$$

Dans l'équation 2.31 nous avons une partie linéaire et une partie non-linéaire. Des méthodes ont été proposées pour corriger l'effet d'écran afin de linéariser le modèle. Nous pouvons citer la correction par absorbance [129] ou encore la correction par la dilution [32, 3]. Il faut souligner qu'en plus de la diffusion et de l'effet d'écran, le bruit de mesure et sa correction automatique introduisent d'autres non-linéarités. Dans cette thèse nous considérons que la diffusion et les phénomènes d'effet d'écran sont respectivement corrigées et négligeables.

### 2.5.6 CPD et CPD en ligne du tenseur de fluorescence

Si on regarde de plus près l'équation 2.29, nous pouvons faire l'analogie avec l'équation 2.14 :

$$\forall i, j, k, \quad \mathcal{T}_{i,j,k} \simeq \widehat{\mathcal{T}}_{i,j,k}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{r=1}^R \mathbf{A}_{ir} \mathbf{B}_{jr} \mathbf{C}_{kr} \quad (2.32)$$

où  $\mathbf{A} \in \mathbb{R}_+^{I \times R}$ ,  $\mathbf{B} \in \mathbb{R}_+^{J \times R}$ ,  $\mathbf{C} \in \mathbb{R}_+^{K \times R}$  sont respectivement les matrices des spectres d'excitation, des spectres d'émission, et du produit des concentrations par le rendement quantique que l'on peut nommer contribution.  $R$  est le rang de la CPD censé être égal au nombre de fluorophores  $\tilde{R}$  présents dans le milieu liquide. Les spectres d'émission, d'excitation et les contributions ayant des valeurs nonnégatives ce qui induit de résoudre un problème de CPD nonnégative.

Dans le cas de la décomposition en ligne, le tenseur de fluorescence  $\mathcal{T}$  peut être vu comme une collection de sous-tenseurs de fluorescence enregistrés à des intervalles de temps consécutifs. Cependant on peut vouloir construire des sous-tenseurs avec un chevauchement (overlapping) et dans ce cas  $\mathcal{T}_n$  contient aussi les dernières MEEFs de  $\mathcal{T}_{n-1}$  comme montré sur la Figure 2.16. Le chevauchement des sous-tenseurs devrait aider les estimations factorielles mais la décomposition en ligne devient alors plus lente car on se retrouve avec un plus grand nombre de sous-tenseurs à traiter. Comme expliqué plus haut, certains fluorophores peuvent apparaître et/ou disparaître entre deux intervalles de temps consécutifs. Par exemple, lors de la surveillance de composants chimiques dans un environnement marin naturel, l'apparition et la disparition de fluorophores peuvent être expliquées par des courants marins ou des événements de pollution, une dégradation naturelle... Par conséquent, le rang physique des sous-tenseurs peut varier avec le temps comme illustré sur la Figure 2.17. Dans la suite, le rang du sous-tenseur physique  $\mathcal{T}_n$  sera noté  $\tilde{R}_n$ .

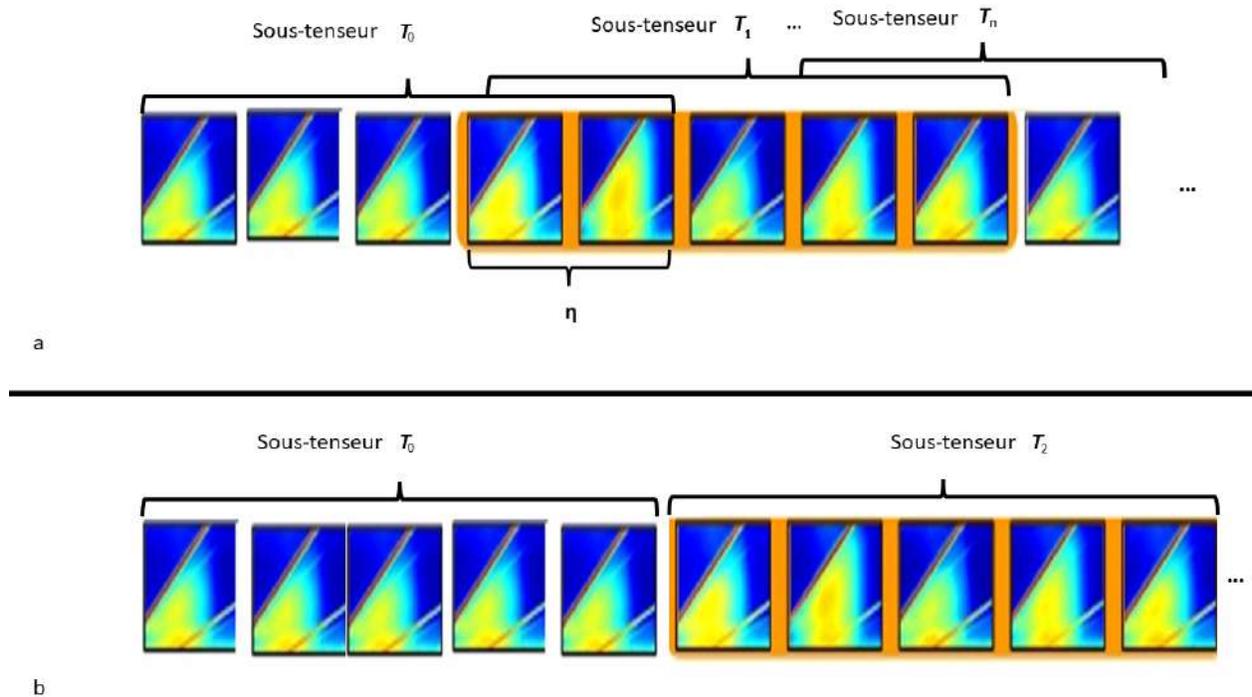


FIGURE 2.16 – Exemple de découpage d'un ensemble d'observation des sous-tenseurs  $\mathcal{T}_n$ . a- Cas avec chevauchement de MEEFs. b- Cas sans chevauchement de MEEFs.

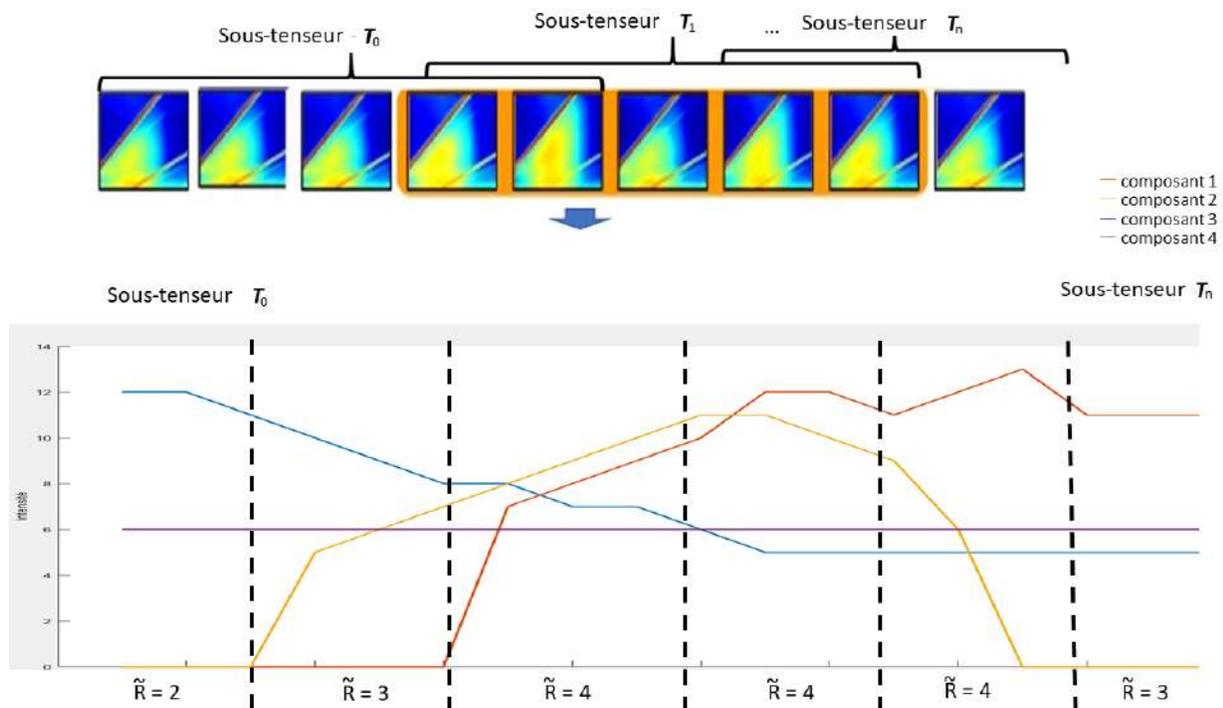


FIGURE 2.17 – Exemple imaginaire de données de fluorescence en ligne avec des profils de concentration de composants fluorescents (unités arbitraires) et la variation de rang physique  $\tilde{R}_n$ .

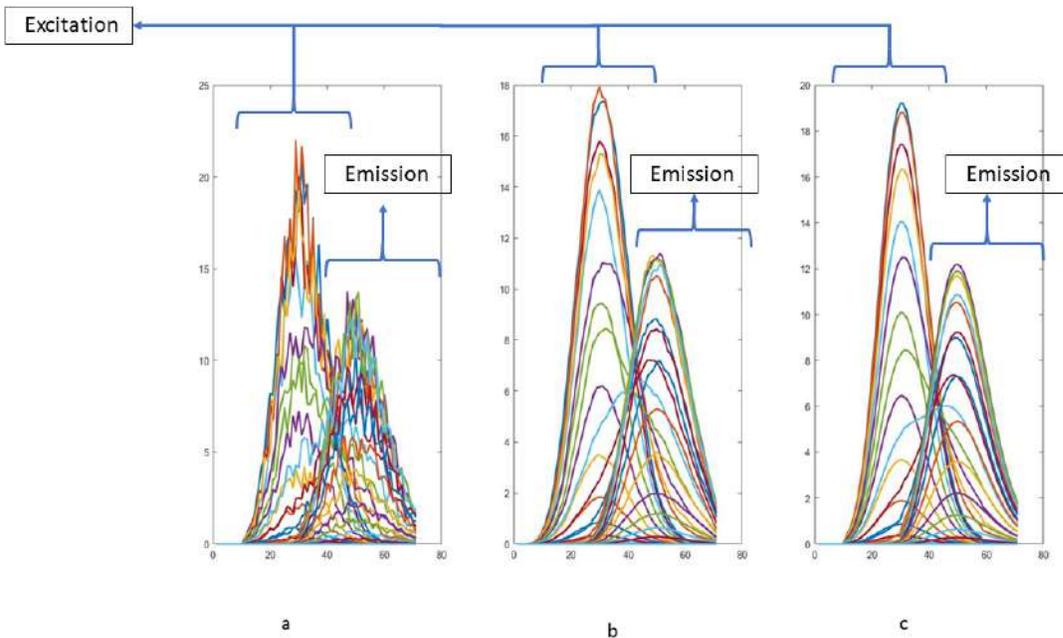
## 2.6 Étude comparative des algorithmes de la CPD sur des données synthétiques de fluorescence

Le but de cette étude est de voir le fonctionnement de certains algorithmes sur des données de fluorescences hors ligne. Pour cela, nous avons choisi quelques algorithmes des 4 toolbox de décompositions tensorielles les plus connus. Il s'agit du toolbox de R.Bro (N-way [130]), de T.Kolda [131], L.deLathauwer (Tensorlab [132]) et de Phan&Cichocki (TensorBox [133]). Les algorithmes suivants ont tous une contrainte de nonnégativité :

- Alternating Least Square (ALS) with nonnegativity constraint [18]. Cet algorithme est le plus utilisé en spectroscopie de fluorescence. L'auteur offre la possibilité d'appliquer une contrainte de nonnégativité sur les différentes matrices facteurs.
- Hierarchical Alternating Least Squares Algorithm (HALS) [134]. Les auteurs accélèrent l'algorithme ALS en minimisant un ensemble de fonctions de coût locales avec les mêmes minima globaux (par exemple, distances euclidiennes au carré et divergences alpha ou bêta avec un paramètre unique alpha ou bêta).
- Fast Recursive ALS algorithm (QALS) [135]. Cette méthode est une extension de l'ALS. Les auteurs utilisent le produit de Khatri-Rao de façon récursive afin d'accélérer la rapidité de convergence.
- Fast computation of gradients for Parafac (MLS) [135]. Une méthode rapide pour calculer le gradient est proposée. La vérification expérimentale montre que le gradient calculé par un produit permet d'accélérer l'algorithme.
- Nonnegative Multiplicative Updates (NMU) [136]. L'approche est basée sur la méthode des poids multiplicatifs. Cela consiste à minimiser la distance euclidienne entre les valeurs approximatives et vraies.
- Decompositions of a higher-order tensor in block terms : Alternating least squares algorithms [137]. Une méthode de décomposition de tenseur semblable à l'ALS, par contre pour assurer une meilleure convergence les auteurs utilisent une nouvelle approche. Le but est de se rapprocher d'un tenseur par une somme de termes de rang faible multilinéaire.
- Enhanced Line Search (ELS) [24]. ALS incorporant une étape de line-search avec pas optimal.
- Parafac In rank-( $L_r; L_r; 1$ ) terms (Minf) [77]. Une méthode basée sur l'utilisation des développements complexes de la série de Taylor pour généraliser les algorithmes d'optimisation existants, tant pour les problèmes généraux d'optimisation non linéaire que pour les problèmes de moindres carrés non linéaires/linéaires.

En plus des algorithmes des toolbox, nous avons ajouté :

- Alternating proximal gradient (APG) [63], la méthode est basée sur le calcul du gradient mais ici le gradient sera approximé avec l'inégalité Kurdyka–Lojasiewicz.



**FIGURE 2.18 – Construction des MEEFs synthétiques. (a) : Simulation de MEEFs bruitées. (b) : MEEF filtrée. (c) : MEEF non bruitées. Excitation : Colonnes des MEEFs. Emission : lignes des MEEFs**

— Nonnegative CPD (NN-CPD) [138], une méthode basée sur le gradient. Nous ajoutons Alternating Direction Method of Multipliers (ADMoM) [76].

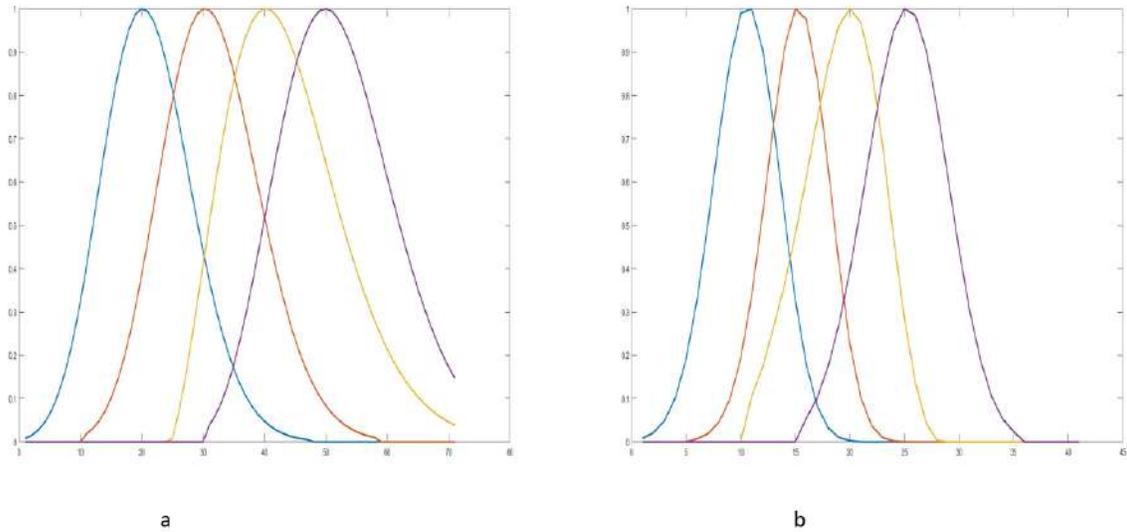
Dans cette simulation, nos MEEFs sont de taille  $41 \times 71$  (soit 41 longueurs d'onde d'excitation et 71 longueurs d'onde d'émission). Ces matrices sont simulées pour être le plus proche possible de la réalité terrain.

Chaque MEEF simulée est construite de la manière suivante :

$$\hat{\mathbf{T}}_k = \mathbf{A} \text{diag}\{\mathbf{c}_{k,\cdot}\} \mathbf{B}^T \quad (2.33)$$

avec  $\text{diag}\{\mathbf{c}_{k,\cdot}\}$ , une matrice diagonale dont les termes diagonaux sont tirés suivant une loi uniforme sur  $[0, 1]$ ,  $\mathbf{A}$  est une matrice  $(41, 4)$  contenant 4 spectres d'excitation synthétiques plus ou moins corrélés.  $\mathbf{B}$  de taille  $(71, 4)$  contenant 4 spectres d'émission synthétiques plus ou moins corrélés.

Se pose alors le problème du type de déviation ou interférence à ajouter à ces MEEFs synthétiques pour simuler l'acquisition de MEEFs réelles. Comme mentionné plus haut nous ne simulerons pas les phénomènes optiques tels que la diffusion ou les effets d'écran qui sont habituellement corrigés numériquement après acquisition ou atténués en jouant sur les paramètres d'acquisition. A notre connaissance peu d'études ont été menées sur le type de bruit issu du dispositif de mesure. D'autant que la plupart des spectrofluorimètres effectuent des corrections automatiques des mesures propres à chaque appareil et sont peu documentées. Il nous a semblé réaliste de considérer que la source de bruit la plus importante était le tube photomultiplicateur (PM). Celui-ci permet la détection du signal lumineux grâce à un dispositif de comptage de photons puis sa conversion et son amplification en signal électrique. Les erreurs peuvent donc venir du comptage et des fluctuations aléatoires du gain du tube PM lors de l'amplification. Par ailleurs, afin d'atténuer les effets de ces fluctuations, les spectres mesurés sont lissés après acquisition par un filtre passe bas. Afin de simuler ces différentes sources de bruit (Figure 2.18), nous proposons le modèle suivant pour construire chaque tranche



**FIGURE 2.19 – Corrélation cas 1. (a) : Spectres d'émission de 4 fluorophores. (b) : Spectres d'excitation de 4 fluorophores.**

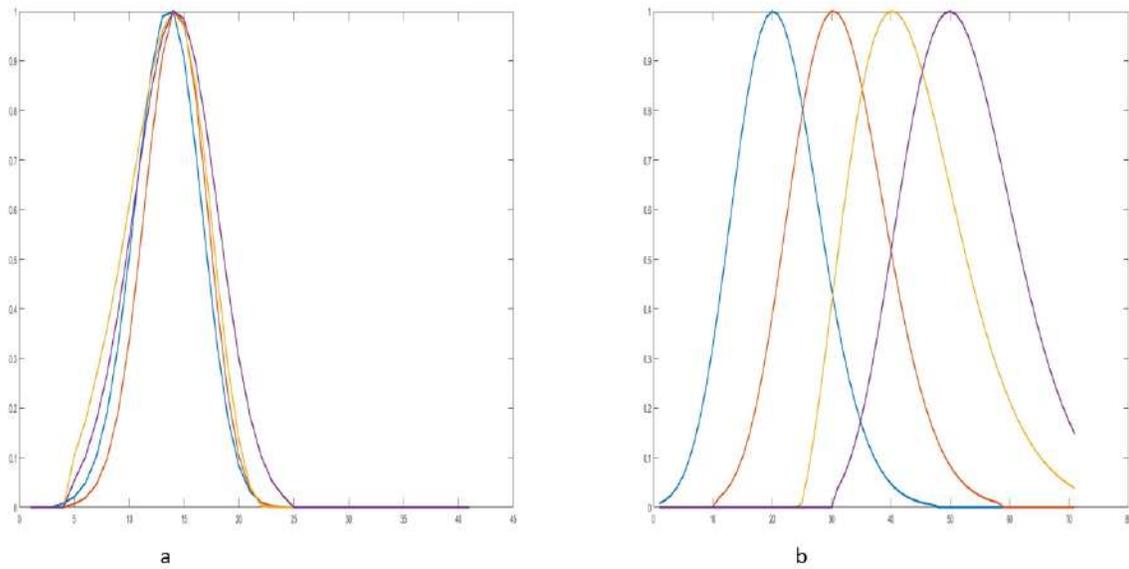
$\mathcal{T}_{\cdot,\cdot,k}$  de notre tenseur synthétique :

$$\mathcal{T}_{\cdot,\cdot,k} = \mathcal{A}\{\mathbf{G}_k \boxtimes \hat{\mathbf{T}}_k\} \quad (2.34)$$

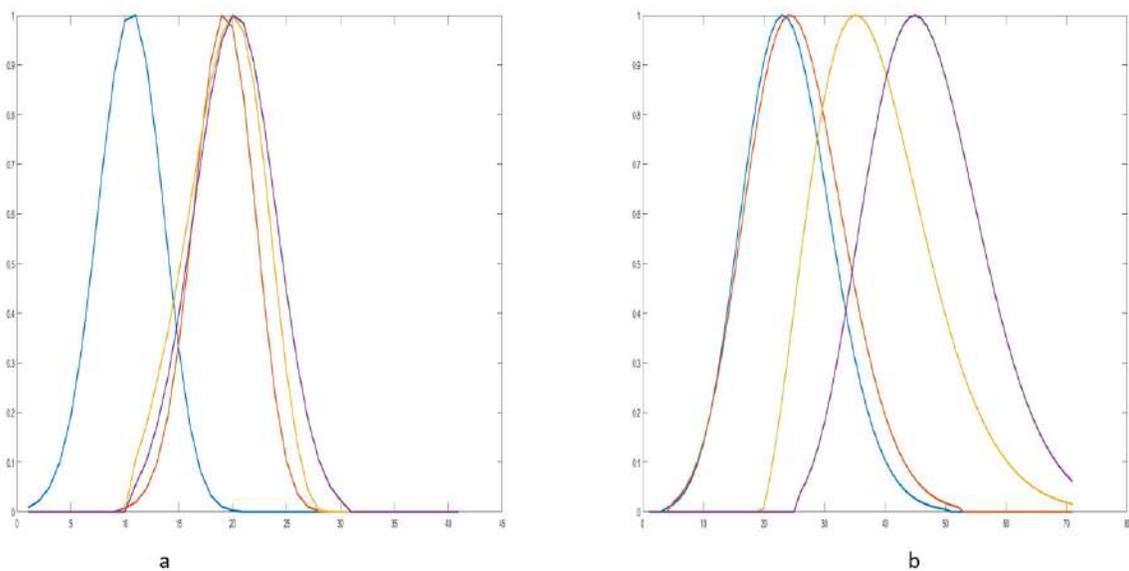
où  $\hat{\mathbf{T}}_k$  et  $\mathbf{G}_k$  sont deux matrices aléatoires de taille  $(I, J)$  simulant respectivement les effets du comptage de photon et les variations du gain du tube PM et  $\mathcal{A}$  est un opérateur de lissage. Plus précisément :

- L'élément  $(i, j)$  de  $\hat{\mathbf{T}}_k$  est tiré aléatoirement à l'aide d'une loi de Poisson de moyenne  $(T_k)_{p,q}$  :  $\forall (i, j), (\hat{T}_k)_{p,q} \sim \text{Pois}((T_k)_{i,j})$ . Rappelons que  $\hat{\mathbf{T}}_k = \mathbf{A} \text{diag}\{\mathbf{c}_{k,\cdot}\} \mathbf{B}^T, \forall k$ .  
Il convient de préciser qu'avant d'appliquer le bruit de Poisson, la matrice  $\hat{\mathbf{T}}_k$  a été normalisée afin que sa plus petite valeur soit égale à un.
- Les éléments de  $\mathbf{G}_k$  sont tirés aléatoirement à partir d'une distribution uniforme dans l'intervalle  $[1 - \alpha; 1 + \alpha]$  :  $\forall (i, j), (G_k)_{i,j} \sim \text{unif}(1 - \alpha, 1 + \alpha)$ .  
Ainsi, le paramètre  $\alpha$  nous permet de jouer sur l'amplitude du bruit multiplicatif.
- Nous avons choisi pour  $\mathcal{A}$  un filtre de type Savitzky-Golay. Ce type de filtre est couramment utilisé par les fabricants de spectrofluorimètres (en revanche les paramètres sont inconnus).  
Dans notre cas nous avons choisi 3 cas de figure avec 4 fluorophores et 10 MEEFs :
  - Cas 1 : Les spectres de tous les composants sont bien séparés en excitation et en émission (Figure 2.19).
  - Cas 2 : Les 4 spectres d'excitation sont très corrélés (Figure 2.20).
  - Cas 3 : Les trois derniers composants ont leurs spectres d'excitation très corrélés. Le premier et le second ont leurs spectres d'émission très corrélés (Figure 2.21).

Nous avons écrit ensuite un programme de comparaison des algorithmes mentionnés ci-dessus en fonction d'un bruit multiplicatif pour  $\alpha \in [0, 1]$  avec un pas de 0.2. Cette comparaison est faite en fonction des 3 cas de corrélations de spectres.



**FIGURE 2.20 –** Corrélation cas 2. (a) : Spectres d'émission de 4 fluorophores. (b) : Spectres d'excitation de 4 fluorophores.



**FIGURE 2.21 –** Corrélation cas 3. (a) : Spectres d'émission de 4 fluorophores. (b) : Spectres d'excitation de 4 fluorophores.

Ainsi, notre tenseur  $\mathcal{T}$  de taille  $41 \times 71 \times 10$  simulé, pour chaque algorithme et pour chaque niveau de bruit, nous avons fait 100 simulations de Monte-Carlo. Les détails sont donnés dans l'algorithme 4. Le but étant de faire une moyenne de l'erreur quadratique (2.14) entre le tenseur original simulé par notre algorithme et le tenseur reconstruit par les algorithmes issus des toolbox. Tous les algorithmes retenus ont besoin d'une initialisation et afin d'avoir la meilleure initialisation, on fera dix initialisations aléatoires de chaque algorithme, retenant l'initialisation qui donne la plus petite erreur. Les résultats donnés sur les Figures 2.22, 2.23 et 2.24, nous montrent que les algorithmes sont très sensibles à l'initialisation d'une part. Ensuite, on remarque que plus les spectres sont corrélés, plus l'erreur devient importante même si les contraintes sont appliquées. Pour finir, l'erreur augmente pour chaque algorithme avec le bruit multiplicatif.

Ces travaux montrent comment certains algorithmes peuvent se comporter en spectroscopie de fluorescence en fonction des difficultés de ce domaine. Cette simulation permet d'anticiper une grande erreur en imposant la contrainte de non-négativité sur nos algorithmes que nous développons dans cette thèse mais aussi la prise en compte du bruit multiplicatif causé par le spectrofluorimètre.

---

**Algorithme 4** Forme générale de notre algorithme de comparaison
 

---

*Entree* :  $T$

```

for  $i = 1$  to  $Nbniveauxdebruit$  do
  for  $j = 1$  to  $NbreAlgo$  do
    for  $k = 1$  to  $simulationdeMonte - Carlo$  do
      for  $d = 1$  to  $Nbreinitialisation$  do
         $[E, it] = calculErreur(T)$ 
      end for
    end for
  end for
end for

```

*Sortie* : sauvegarde des résultats des erreurs et nombre d'itérations

---

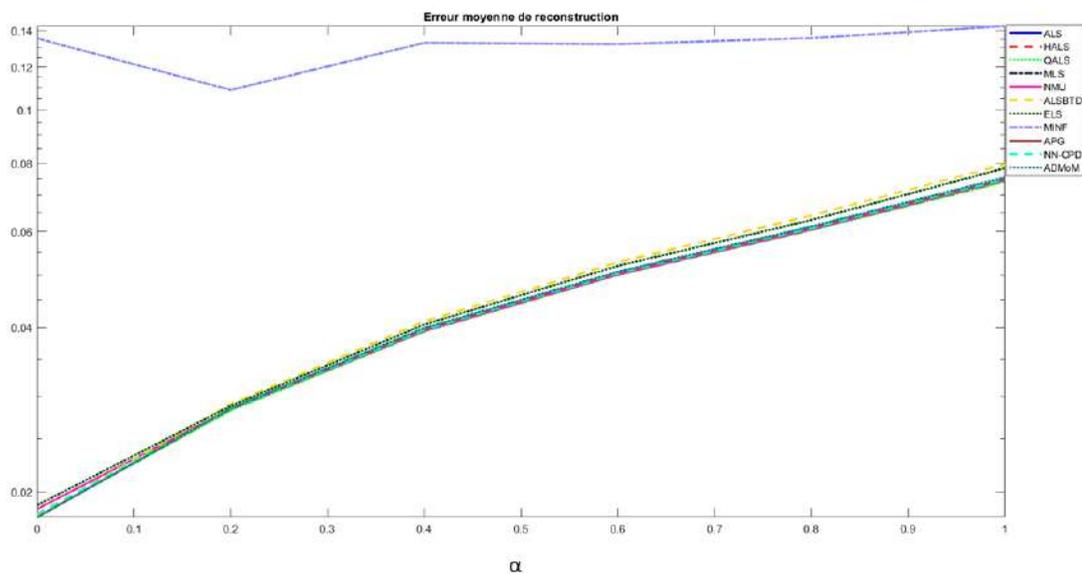


FIGURE 2.22 – Erreur moyenne de reconstruction en fonction de  $\alpha$  des différents algorithmes pour le cas 1.

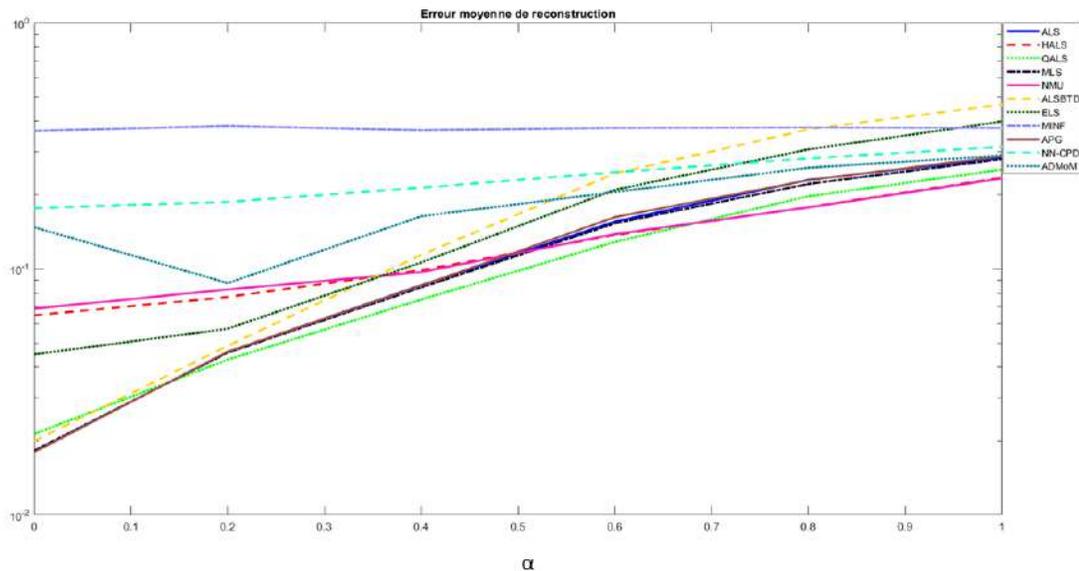


FIGURE 2.23 – Erreur moyenne de reconstruction en fonction de  $\alpha$  des différents algorithmes pour le cas 2.

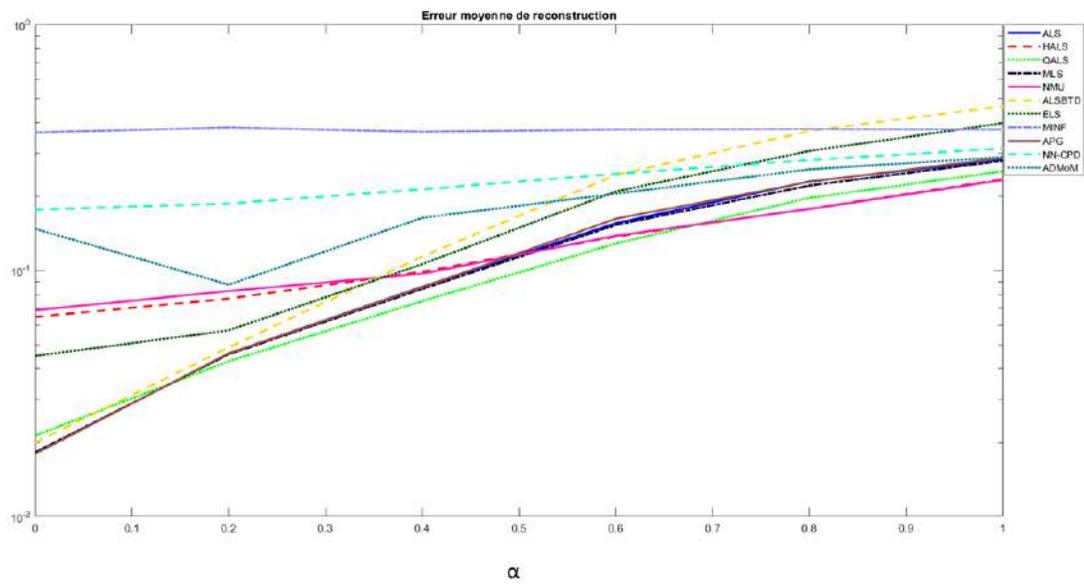


FIGURE 2.24 – Erreur moyenne de reconstruction en fonction de  $\alpha$  des différents algorithmes pour le cas 3.

## 2.7 Conclusion du chapitre

Dans ce chapitre, nous avons introduit les notions de décomposition tensorielle et de décomposition tensorielle en ligne. Les notions mathématiques nécessaires à la compréhension du problème ont été évoquées ainsi qu'un état de l'art des algorithmes a été détaillé. Nous avons fait le lien aussi entre la CPD et la spectroscopie de fluorescence qui est une technique de caractérisation des matières dissoutes en environnement (ou en géosciences). Nous avons effectué une étude comparative des différents algorithmes dans le but voir comment ces algorithmes de CPD se comportent sur des données synthétiques de fluorescence. On retient qu'une imposition de certaines contraintes lors de l'écriture de la fonction coût comme la non-négativité peut aider à une bonne convergence. L'initialisation joue aussi un rôle majeur dans la plupart des algorithmes. C'est pourquoi certains algorithmes utilisent une pré-initialisation des matrices facteurs avec des algorithmes issus de la première famille ([139]). Dans la suite, nous présentons les travaux effectués dans le cadre de cette thèse pour répondre à la problématique de la décomposition tensorielle en ligne avec contrainte de non-négativité et variation de rang au cours du temps. Le chapitre suivant sera donc consacré aux algorithmes de CPD en ligne que nous avons développés.



## Chapitre 3

# La décomposition Canonique Polyadique nonnégative de tenseurs en ligne

Nous allons maintenant proposer dans ce chapitre les algorithmes que nous avons développés pour répondre au problème de décomposition de tenseurs et décomposition de tenseurs en ligne. Nous rappelons le modèle de la CPD :

$$\forall i, j, k, \mathcal{T}_{i,j,k} \simeq \widehat{\mathcal{T}}_{i,j,k}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{r=1}^R \mathbf{A}_{ir} \mathbf{B}_{jr} \mathbf{C}_{kr} \quad (3.1)$$

avec  $\mathbf{A} \in \mathbb{R}_+^{I \times R}$ ,  $\mathbf{B} \in \mathbb{R}_+^{J \times R}$ ,  $\mathbf{C} \in \mathbb{R}_+^{K \times R}$  où  $R$  est inconnu. Les problèmes de CPD en ligne auront deux phases :

- Une phase d'initialisation : Cette phase est nécessaire parce que nous supposons n'avoir aucune information a priori à l'instant  $t = 0$ .
- Une phase en ligne : à l'instant  $t > 0$ , nous rentrons dans la phase en ligne.

Le principal problème est le nombre de facteurs (rang de la CPD) inconnu à l'instant  $t = 0$  et ses variations possibles  $t > 0$  : apparition et/ou disparition possible de nouveaux facteurs.

Rappelons l'expérimentation évoquée dans le chapitre précédent, la méthode de descente de gradient a retenu notre attention alors nous la choisissons pour optimiser nos différents problèmes. Cette méthode d'optimisation a eu beaucoup de succès dans la décomposition tensorielle du fait de sa flexibilité et de son implémentation [138, 45, 87, 140, 141] incluant les tenseurs de fluorescence [93]. De plus, elle est facile à étendre à la CPD en ligne [142, 143]. Nous la présentons dans la section suivante avant d'introduire nos algorithmes de CPD en ligne.

## 3.1 Algorithme d'optimisation utilisé pour minimiser les fonctions de coût

### 3.1.1 Descente de gradient stochastique (SGD)

Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ , la mise à jour de  $\mathbf{x}$  en descente de gradient est donnée par :

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \gamma \mathbf{f}'(\mathbf{x}(k)) \quad (3.2)$$

avec  $\gamma > 0$  le pas d'apprentissage (le step-size ou learning rate). L'algorithme de descente de gradient nécessite un point d'initialisation  $\mathbf{x}(0)$ , par exemple, un point sélectionné au hasard dans l'espace de  $\mathbf{x}$ . Le gradient de  $f$  est ensuite calculé et le pas d'apprentissage est choisi de manière optimale où à l'aide d'une heuristique. La descente de gradient stochastique est similaire à la descente de gradient. La descente de gradient stochastique (stochastic gradient descent, SGD) est un principe plus général, dans lequel la direction de la mise à jour est une variable aléatoire, dont l'espérance est le véritable gradient qui nous intéresse. Les conditions de convergence de la SGD sont similaires à celles de la descente de gradient, malgré la plus grande quantité de hasard.

Une limitation de la descente de gradient est que la progression de la recherche peut ralentir si le gradient devient plat ou à grande courbure. Par exemple, un élan peut être ajouté à la descente de gradient qui intègre une certaine inertie lors de la mises à jour. Généralement  $\gamma$  est fixé pour chaque itération  $k$ , mais peut être adaptatif. Afin d'améliorer la convergence, plusieurs variantes de la descente de gradient ont vu le jour.

### 3.1.2 Descente de gradient avec élan (Momentum)

Descente de gradient avec élan ou Momentum [144] est une méthode qui permet d'accélérer la descente dans une direction  $\mathbf{v}_k$  appropriée et d'amortir les oscillations. Pour ce faire, on choisit la direction comme une combinaison linéaire du gradient et de la direction précédente (variante du gradient conjugué). Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ , la mise à jour de  $\mathbf{x}$  est donnée par :

$$\begin{aligned} \mathbf{v}_k &= \beta \mathbf{v}_{k-1} + \gamma \mathbf{f}'(\mathbf{x}(k)) \\ \mathbf{x}(k+1) &= \mathbf{x}(k) - \mathbf{v}_k \end{aligned} \quad (3.3)$$

avec  $\gamma > 0$  le pas d'apprentissage,  $\beta \in [0, 1]$  et  $\mathbf{v}$  initialisé à 0. Certaines implémentations échangent les signes dans les équations. Le terme de "quantité de mouvement"  $\beta$  est généralement fixé à 0,9 ou à une valeur proche. Une bonne stratégie peut être de modifier  $\beta$  en fonction de  $\gamma$ .

### 3.1.3 Nesterov Accelerated Gradient (NAG)

Le gradient accéléré de Nesterov ou Nesterov Accelerated Gradient (NAG) [145] est un moyen de donner un terme d'impulsion lors de la mise à jour. On présente souvent l'optimisation de l'accélération de Nesterov comme une balle roulant sur la colline mais sait exactement quand ralentir avant que la pente de la colline n'augmente à nouveau. On calcule le gradient non pas par rapport au pas d'apprentissage courant mais par rapport au pas d'apprentissage futur. NAG est donc une extension de SGD et du momentum. Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ ,

la mise à jour de  $\mathbf{x}$  est donnée par :

$$\begin{aligned} \mathbf{v}_k &= \beta \mathbf{v}_{k-1} + \gamma \mathbf{f}'(\mathbf{x}(k) - \mathbf{v}_{k-1}) \\ \mathbf{x}(k+1) &= \mathbf{x}(k) - \mathbf{v}_k \end{aligned} \quad (3.4)$$

avec  $\gamma > 0$  le pas d'apprentissage,  $\beta \in [0, 1]$  et  $\mathbf{v}$  initialisé à 0. NAG fonctionne légèrement mieux que le momentum standard.

### 3.1.4 Adaptive gradient (Adagrad)

Le gradient adaptatif ou Adaptive gradient (AdaGrad) [146] agit sur le pas d'apprentissage en le divisant par la racine carrée de  $\mathbf{v}$ , qui est la somme cumulée des gradients au carré actuels et passés (c'est-à-dire jusqu'à l'itération  $k$ ). Notons que la composante de gradient reste inchangée comme dans SGD. Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ , la mise à jour de  $\mathbf{x}$  est donnée par :

$$\begin{aligned} \mathbf{g}_k &= \mathbf{f}'(\mathbf{x}(k)) \\ \mathbf{v}_k &= \mathbf{v}_{k-1} + \mathbf{g}_k^2 \\ \mathbf{x}(k+1) &= \mathbf{x}(k) - \frac{\gamma}{\sqrt{\mathbf{v}_k + \epsilon}} \mathbf{g}_k \end{aligned} \quad (3.5)$$

avec  $\gamma > 0$  le pas d'apprentissage et  $\mathbf{v}$  initialisé à 0.

### 3.1.5 Adaptive learning rate (Adadelta)

Adaptive learning rate ou Adadelta [147] est une amélioration de AdaGrad concentrant sur le pas d'apprentissage. Adadelta est l'abréviation de "delta adaptatif", où delta fait ici référence à la différence entre le poids actuel et le poids nouvellement mis à jour. Autrement dit, Adadelta est une extension d'Adagrad qui cherche à réduire son pas d'apprentissage (step-size ou learning rate) agressif et décroissant de manière monotone. Au lieu d'accumuler tous les gradients passés au carré, Adadelta restreint la fenêtre des gradients passés accumulés à une taille fixe (poids). Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ , la mise à jour de  $\mathbf{x}$  est donnée par :

$$\begin{aligned} \mathbf{g}_k &= \mathbf{f}'(\mathbf{x}(k)) \\ \delta_k &= \mathbf{x}(k) - \mathbf{x}(k-1) \\ \mathbf{v}_k &= \beta \mathbf{v}_{k-1} + (1 - \beta) \mathbf{g}_k^2 \\ \mathbf{d}_k &= \beta \delta_k^2 + (1 - \beta) \mathbf{g}_k^2 \\ \mathbf{x}(k+1) &= \mathbf{x}(k) - \frac{\gamma \sqrt{\mathbf{d}_{k-1} + \epsilon}}{\sqrt{\mathbf{v}_k + \epsilon}} \mathbf{g}_k \end{aligned} \quad (3.6)$$

avec  $\gamma > 0$  le pas d'apprentissage,  $\beta \in [0, 1]$ ,  $\mathbf{d}$ ,  $\mathbf{v}$  initialisés à 0.

### 3.1.6 Root Mean Square prop (RMSprop)

Root Mean Square Prop (RMSprop), une méthode non publiée et proposée par Geoff Hinton ([http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_1ec6.pdf](http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_1ec6.pdf)). RMSprop est une autre méthode du pas d'apprentissage adaptatif qui tente d'améliorer

AdaGrad. Au lieu de prendre la somme cumulée des gradients au carré comme dans AdaGrad, nous prenons la moyenne quadratique glissante de ces gradients. Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ , la mise à jour de  $\mathbf{x}$  est donnée par :

$$\begin{aligned} \mathbf{g}_k &= \mathbf{f}'(\mathbf{x}(k)) \\ \mathbf{v}_k &= \beta \mathbf{v}_{k-1} + (1 - \beta) \mathbf{g}_k^2 \\ \mathbf{x}(k+1) &= \mathbf{x}(k) - \frac{\gamma}{\sqrt{\mathbf{v}_k + \epsilon}} \mathbf{g}_k \end{aligned} \quad (3.7)$$

avec  $\gamma > 0$  le pas d'apprentissage,  $\beta \in [0, 1]$  et  $\mathbf{v}$  initialisé à 0.

### 3.1.7 ADaptive Moment estimation (ADAM)

ADaptive Moment estimation (Adam) [148], est simplement une combinaison du momentum et de RMSprop. Il agit sur la composante de gradient en utilisant la moyenne mobile exponentielle des gradients (comme dans le momentum), et la composante du learning rate en divisant le learning rate par la racine carrée de  $\mathbf{v}$ , la moyenne mobile exponentielle des gradients au carré (comme dans RMSprop). Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ , la mise à jour de  $\mathbf{x}$  est donnée par :

$$\begin{aligned} \mathbf{g}_k &= \mathbf{f}'(\mathbf{x}(k)) \\ \mathbf{m}_k &= \beta_1 \mathbf{m}_{k-1} + (1 - \beta_1) \mathbf{g}_k \\ \mathbf{v}_k &= \beta_2 \mathbf{v}_{k-1} + (1 - \beta_2) \mathbf{g}_k^2 \\ \hat{\mathbf{m}}_k &= \frac{\mathbf{m}_k}{1 - (\beta_1)^k} \\ \hat{\mathbf{v}}_k &= \frac{\mathbf{v}_k}{1 - (\beta_2)^k} \\ \mathbf{x}(k+1) &= \mathbf{x}(k) - \frac{\gamma}{\sqrt{\hat{\mathbf{v}}_k + \epsilon}} \hat{\mathbf{m}}_k \end{aligned} \quad (3.8)$$

avec  $\gamma > 0$  le pas d'apprentissage,  $\beta_1 \in [0, 1]$ ,  $\beta_2 \in [0, 1]$  et  $\mathbf{m}, \mathbf{v}$  initialisés à 0.

### 3.1.8 Nesterov accelerated ADaptive Moment estimation (Nadam)

L'algorithme Nesterov accelerated ADaptive Moment estimation ou Nadam, est une extension de l'algorithme d'optimisation Adam pour y ajouter NAG. L'algorithme a été décrit dans l'article de 2016 de Timothy Dozat [149]. Prenons une fonction dérivable  $f(\mathbf{x})$ . Pour chaque itération  $k$ , la mise à jour de  $\mathbf{x}$  est donnée par :

$$\begin{aligned} \mathbf{g}_k &= \mathbf{f}'(\mathbf{x}(k-1)) \\ \mathbf{m}_k &= \beta_1 \mathbf{m}_{k-1} + (1 - \beta_1) \mathbf{g}_k \\ \mathbf{n}_k &= \beta_2 \mathbf{n}_{k-1} + (1 - \beta_2) \mathbf{g}_k^2 \\ \hat{\mathbf{m}}_k &= \frac{\beta_1 \mathbf{m}_k}{1 - (\beta_1)^k} + \frac{(1 - \beta_1) \mathbf{g}_k}{1 - (\beta_1)^k} \\ \hat{\mathbf{n}}_k &= \frac{\beta_2 \mathbf{n}_k}{1 - \beta_2} \\ \mathbf{x}(k+1) &= \mathbf{x}(k) - \frac{\gamma \hat{\mathbf{m}}_k}{\sqrt{\hat{\mathbf{n}}_k + \epsilon}} \end{aligned} \quad (3.9)$$

Nadam est l'algorithme d'optimisation que nous avons choisi pour la suite car il intègre la convergence accélérée de NAG et la précision de Adam [150]. Nous pouvons étendre la mise à jour Nadam en fonction du cas de la CPD. Prenons l'équation 2.18, pour chaque itération  $k$ , la mise à jour de  $\mathbf{A}$ ,  $\mathbf{B}$  et  $\mathbf{C}$  est donnée par :

$$\begin{aligned}
 \mathbf{G}_k^A &= \frac{\partial E}{\partial \mathbf{A}}; \quad \mathbf{G}_k^B = \frac{\partial E}{\partial \mathbf{B}}; \quad \mathbf{G}_k^C = \frac{\partial E}{\partial \mathbf{C}} \\
 \mathbf{M}_k^A &= \beta_1 \mathbf{M}_{k-1}^A + (1 - \beta_1) \mathbf{G}_k^A \\
 \mathbf{M}_k^B &= \beta_1 \mathbf{M}_{k-1}^B + (1 - \beta_1) \mathbf{G}_k^B \\
 \mathbf{M}_k^C &= \beta_1 \mathbf{M}_{k-1}^C + (1 - \beta_1) \mathbf{G}_k^C \\
 \mathbf{N}_k^A &= \beta_2 \mathbf{N}_{k-1}^A + (1 - \beta_2) (\mathbf{G}_k^A)^2 \\
 \mathbf{N}_k^B &= \beta_2 \mathbf{N}_{k-1}^B + (1 - \beta_2) (\mathbf{G}_k^B)^2 \\
 \mathbf{N}_k^C &= \beta_2 \mathbf{N}_{k-1}^C + (1 - \beta_2) (\mathbf{G}_k^C)^2 \\
 \hat{\mathbf{M}}_k^A &= \frac{\beta_1 \mathbf{M}_k^A}{1 - (\beta_1)^k} + \frac{(1 - \beta_1) \mathbf{G}_k^A}{1 - (\beta_1)^k} \\
 \hat{\mathbf{M}}_k^B &= \frac{\beta_1 \mathbf{M}_k^B}{1 - (\beta_1)^k} + \frac{(1 - \beta_1) \mathbf{G}_k^B}{1 - (\beta_1)^k} \\
 \hat{\mathbf{M}}_k^C &= \frac{\beta_1 \mathbf{M}_k^C}{1 - (\beta_1)^k} + \frac{(1 - \beta_1) \mathbf{G}_k^C}{1 - (\beta_1)^k} \\
 \hat{\mathbf{N}}_k^A &= \frac{\beta_2 \mathbf{N}_k^A}{1 - \beta_2}; \quad \hat{\mathbf{N}}_k^B = \frac{\beta_2 \mathbf{N}_k^B}{1 - \beta_2}; \quad \hat{\mathbf{N}}_k^C = \frac{\beta_2 \mathbf{N}_k^C}{1 - \beta_2} \\
 \mathbf{A}(k+1) &= \mathbf{A}(k) - \frac{\gamma \hat{\mathbf{M}}_k^A}{\sqrt{\hat{\mathbf{N}}_k^A + \epsilon}} \\
 \mathbf{B}(k+1) &= \mathbf{B}(k) - \frac{\gamma \hat{\mathbf{M}}_k^B}{\sqrt{\hat{\mathbf{N}}_k^B + \epsilon}} \\
 \mathbf{C}(k+1) &= \mathbf{C}(k) - \frac{\gamma \hat{\mathbf{M}}_k^C}{\sqrt{\hat{\mathbf{N}}_k^C + \epsilon}}
 \end{aligned} \tag{3.10}$$

Pour encore mieux accélérer cette convergence, nous adaptons le pas d'apprentissage à chaque itération suivant le processus décrit dans [151] :

$$\begin{aligned}
 \text{Si } \|f(\mathbf{x}) - f(\mathbf{x} - 1)\| &< \epsilon \\
 \gamma &= \gamma * 2 \\
 \text{Sinon } \gamma &= \gamma/3
 \end{aligned} \tag{3.11}$$

Les détails de Nadam sont donnés dans l'algorithme 5.

avec  $\gamma > 0$  le pas d'apprentissage,  $\beta_1 \in [0, 1]$ ,  $\beta_2 \in [0, 1]$  sont des hyperparamètres qui peuvent être adaptés en fonction des données. Les valeurs  $\mathbf{m}$ ,  $\mathbf{v}$  sont initialisées à 0. Une comparaison des différents algorithmes d'optimisation est présenté dans [150] ainsi que d'autres algorithmes que nous ne présentons pas ici. Les algorithmes présentés dans cette thèse utiliseront l'algorithme Nadam car il assure une meilleure convergence que SGD, il est moins sensible au pas d'apprentissage ([150]) et est aussi facile à mettre en place.

**Algorithme 5** Nadam optimizer

---

• **Entrée :**  $\gamma, f(x), \beta_1, \beta_2, eps = 1e^{-6}$   $K = \text{nombre d'iteration max}$   
 $\mathbf{m}(1) = 0, \mathbf{n}(1) = 0, k = 1$   
**while**  $k < K$  **do**  
 $\mathbf{g}_k = \mathbf{f}'(\mathbf{x}(k - 1))$   
 $\mathbf{m}_k = \beta_1 \mathbf{m}_{k-1} + (1 - \beta_1) \mathbf{g}_k$   
 $\mathbf{n}_k = \beta_2 \mathbf{n}_{k-1} + (1 - \beta_2) \mathbf{g}_k^2$   
 $\hat{\mathbf{m}}_k = \frac{\beta_1 \mathbf{m}_k}{1 - \beta_1^k} + \frac{(1 - \beta_1) \mathbf{g}_k}{1 - \beta_1^k}$   
 $\hat{\mathbf{n}}_k = \frac{\beta_2 \mathbf{n}_k}{1 - \beta_2^k}$   
 $\mathbf{x}(k + 1) = \mathbf{x}(k) - \frac{\gamma \hat{\mathbf{m}}}{\sqrt{\hat{\mathbf{n}}_k + \epsilon}}$   
calcul de  $\gamma$  à l'aide de l'équation 3.11  
 $k = k + 1$   
**end while**  
• **Sortie :** optimized parameters  $f(x)$

---

**3.1.9** Contrainte de nonnégativité

Rappelons ici que la nonnégativité est assurée par la projection des éléments de chaque matrice facteur sur  $\mathbb{R}^+$  [94, 152] à chaque itération, autrement dit, en mettant à zéro chaque élément négatif. Ce choix est classique comme en témoigne [45, 63, 153]. En utilisant l'algorithme 5, la mise à jour lors du processus d'optimisation peut s'écrire :

$$x(k + 1) \leftarrow \max\left(\mathbf{x}(k) - \frac{\gamma \hat{\mathbf{m}}}{\sqrt{(\hat{\mathbf{n}}) + eps}}, 0\right) \quad (3.12)$$

## 3.2 Décomposition de tenseurs avec contrainte de parcimonie et de nonnégativité. *Offline Step (ou phase d'initialisation)*

En ayant aucune information a priori sur les matrices facteur, le défi est de retrouver les bons facteurs avec un rang inconnu au départ. Pour cela nous nous inspirons des techniques d'apprentissage de dictionnaires parcimonieux.

### 3.2.1 Décomposition parcimonieuse et nonnégative de matrices

La décomposition nonnégative de matrices ou Nonnegative Matrix Factorization (NMF) est une résultante de l'apprentissage de dictionnaire avec des contraintes comme la nonnégativité. L'apprentissage de dictionnaires est une technique standard populaire pour apprendre des caractéristiques interprétables, utilisée dans plusieurs domaines comme le traitement de signal [154], le traitement d'image [155], l'analyse spectrale [156, 157] ou la classification [158]. La décomposition nonnégative de matrices est le fait d'approximer la matrice  $\mathbf{X}$  par le produit de deux matrices  $\mathbf{D}$  par  $\mathbf{Y}$  :

$$\mathbf{X} \simeq \mathbf{D}\mathbf{Y}, \text{ S.t } \mathbf{D} \geq 0, \mathbf{Y} \geq 0 \quad (3.13)$$

Cette approximation peut se faire en minimisant une fonction de coût basée sur la relation suivante :

**la norme euclidienne [153] :**

$$E(\mathbf{D}, \mathbf{Y}) = \|\mathbf{X} - \mathbf{D}\mathbf{Y}\|^2 \quad (3.14)$$

**la divergence de Kullback-Leibler [159] :**

$$D_{KL}(\mathbf{X} \parallel \mathbf{D}\mathbf{Y}) = \sum_{ij} \left( \mathbf{X}_{ij} \log \frac{\mathbf{X}_{ij}}{(\mathbf{D}\mathbf{Y})_{ij}} - \mathbf{X}_{ij} + (\mathbf{D}\mathbf{Y})_{ij} \right) \quad (3.15)$$

**la divergence de Itakura-Saito [160] :**

$$D_{IS}(\mathbf{X} \parallel \mathbf{D}\mathbf{Y}) = \sum_{ij} \left( \frac{\mathbf{X}_{ij}}{(\mathbf{D}\mathbf{Y})_{ij}} - \log \frac{\mathbf{X}_{ij}}{(\mathbf{D}\mathbf{Y})_{ij}} - 1 \right) \quad (3.16)$$

Dans beaucoup d'applications sous contrainte de nonnégativité du traitement du signal, il est pertinent de représenter le signal avec une partie dense appelée dictionnaire et une partie "creuse" ou parcimonieuse appelée atome. Autrement dit, la notion de parcimonie consiste à reconstruire un signal en utilisant un nombre minimal d'éléments tout en restant fidèle au signal de départ. Cette représentation a été largement étudiée ces dernières années notamment en débruitage d'images [161], en séparation de sources [79] ou en analyse spectrale [157]. La NMF parcimonieuse ne garantit pas l'unicité de la décomposition, ce qui est problématique pour analyser des signaux qui ont un sens physique. C'est là l'une des principales motivations des approches tensorielles.

### 3.2.2 CPD nonnégative par apprentissage de dictionnaire avec surestimation du rang

Avant de calculer la CPD en ligne, nous devons travailler sur un premier sous-tenseur sans a priori sur les facteurs. Il s'agit d'une étape *Offline* CPD. Afin de traiter le problème de rang inconnu, nous décidons de surestimer la valeur du rang et de forcer les facteurs supplémentaires à zéro. Pour ce faire, nous représentons nos matrices  $\mathbf{A}$  et  $\mathbf{B}$  comme le produit de dictionnaires parcimonieux  $\mathbf{D}^A$  et  $\mathbf{D}^B$  par des atomes  $\mathbf{V}^A$  et  $\mathbf{V}^B$  respectivement :

$$\mathbf{A} = \mathbf{D}^A \mathbf{V}^A \text{ et } \mathbf{B} = \mathbf{D}^B \mathbf{V}^B, \quad (3.17)$$

où  $\mathbf{D}^A$  et  $\mathbf{D}^B$  sont respectivement de taille  $(I, R)$  et  $(J, R)$  et  $\mathbf{V}^A$  et  $\mathbf{V}^B$  sont  $(R, R)$ . Nos dictionnaires  $\mathbf{D}^A$  et  $\mathbf{D}^B$  contiennent les différents facteurs ayant un sens physique (les signaux par exemple) sur chaque colonne et nos atomes  $\mathbf{V}^A$  et  $\mathbf{V}^B$  sont les poids correspondant à chaque facteur. L'objectif est d'avoir les matrices atomes ( $\mathbf{V}^A$  et  $\mathbf{V}^B$ ) creuses. Nous utilisons la norme  $L_{1,1}$  pour promouvoir la parcimonie (cf section 2.3.3.1). Cette approche est largement utilisée dans la NMF mais aussi en CPD avec le problème de rang inconnu [162, 163, 63, 138, 102, 164]. La norme  $L_{1,1}$  est classiquement utilisée pour promouvoir la parcimonie [98, 164, 45, 162] car elle est plus facile à traiter que la norme  $L_{0,0}$ .

Cette contrainte permet lorsque le rang est surestimé de déduire le rang réel de la décomposition car, après convergence de l'algorithme, on doit obtenir des colonnes remplies de zéros sur les matrices contraintes. Ici, nous choisissons le rang de la CPD  $R$  supérieur à  $\tilde{R}$  où  $\tilde{R}$  est le rang physique. Les matrices facteurs  $\mathbf{A}$  de taille  $(I, R)$  et  $\mathbf{B}$  de taille  $(J, R)$  sont de dimensions fixes. Nous nous attendons donc à ce que  $\mathbf{D}^A$  et  $\mathbf{D}^B$  contiennent  $\tilde{R}$  vrais facteurs parmi leurs colonnes avec  $R - \tilde{R}$  facteurs sans signification physique. On souhaite que  $\mathbf{V}^A$  et  $\mathbf{V}^B$  aient  $R - \tilde{R}$  colonnes nulles et que les autres colonnes forment une matrice de permutation généralisée, c'est-à-dire une matrice carrée dont tous les coefficients sont égaux à 0, sauf un coefficient sur chaque ligne et sur chaque colonne. Pour cela, nous résolvons le problème de minimisation suivant :

$$\begin{aligned} \min \{ E_1(\mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C}) \} \text{ s.t. } \mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C} \geq 0 \\ \text{où } E_1 = \frac{1}{2} \|\mathcal{T} - \hat{\mathcal{T}}(\mathbf{D}^A \mathbf{V}^A, \mathbf{D}^B \mathbf{V}^B, \mathbf{C})\|_F^2 + \alpha \|\mathbf{V}^A\|_{1,1} + \alpha \|\mathbf{V}^B\|_{1,1} \end{aligned} \quad (3.18)$$

avec  $\alpha > 0$  le coefficient de pénalité. Il est à noter que contrairement à la plupart des algorithmes CPD avec contrainte de parcimonie, nous n'appliquons pas la contrainte directement sur la matrice facteur mais sur l'atome. C'est pourquoi on ne cherche pas à minimiser le rang des colonnes mais à favoriser l'apparition de zéros dans toute la matrice. Nous privilégions donc ici la norme  $L_{1,1}$  plutôt que les normes mixtes comme dans [102]. Par exemple, la matrice  $\mathbf{A}$  peut être écrite comme un produit de son dictionnaire  $\mathbf{D}^A$  avec son atome creux  $\mathbf{V}^A$  :

$$\mathbf{A} = \mathbf{D}^A \mathbf{V}^A = \begin{pmatrix} 2 & 9 & 1 & 11 \\ 5 & 7 & 4 & 13 \\ 10 & 21 & 3 & 4 \\ 2 & 1 & 6 & 3 \\ 14 & 0 & 2 & 5 \\ 2 & 7 & 5 & 9 \end{pmatrix} \cdot \begin{pmatrix} 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 1.8 & 0 & 22 & 0 \\ 4.5 & 0 & 26 & 0 \\ 9 & 0 & 8 & 0 \\ 1.8 & 0 & 6 & 0 \\ 3.6 & 0 & 10 & 0 \\ 1.8 & 0 & 18 & 0 \end{pmatrix}$$

Ainsi le nombre de colonnes non nulles de  $\mathbf{A}$  ou  $\mathbf{B}$  nous donne une estimation *a posteriori* de  $\tilde{R}$  avec  $R = 4$  et  $\tilde{R} = 2$ . On considère qu'une colonne est nulle si sa norme  $L_{1,1}$  est inférieure à une valeur seuil. Ici, cette valeur est de  $10^{-5}$ .

Nous résolvons l'équation (3.18) pour le sous-tenseur  $\mathcal{T} = \mathcal{T}_0$  à l'instant  $t_0$  en utilisant l'algorithme d'optimisation Nadam. Il faut donc calculer le gradient de  $E_1$  par rapport aux différentes variables :

$$\begin{aligned}
 \frac{\partial E_1}{\partial \mathbf{V}^A} &= -(\mathbf{D}^A)^\top (\mathbf{T}^A - \mathbf{D}^A \mathbf{V}^A (\mathbf{L}^A)^\top) \mathbf{L}^A + \alpha \mathbf{1}_{R,R} \\
 \frac{\partial E_1}{\partial \mathbf{V}^B} &= -(\mathbf{D}^B)^\top (\mathbf{T}^B - \mathbf{D}^B \mathbf{V}^B (\mathbf{L}^B)^\top) \mathbf{L}^B + \alpha \mathbf{1}_{R,R} \\
 \frac{\partial E_1}{\partial \mathbf{D}^A} &= -(\mathbf{T}^A - \mathbf{D}^A \mathbf{V}^A (\mathbf{L}^A)^\top) \mathbf{L}^A (\mathbf{V}^A)^\top \\
 \frac{\partial E_1}{\partial \mathbf{D}^B} &= -(\mathbf{T}^B - \mathbf{D}^B \mathbf{V}^B (\mathbf{L}^B)^\top) \mathbf{L}^B (\mathbf{V}^B)^\top \\
 \frac{\partial E_1}{\partial \mathbf{C}} &= -(\mathbf{T}^C - \mathbf{C} (\mathbf{L}^C)^\top) (\mathbf{L}^C)
 \end{aligned} \tag{3.19}$$

avec  $\mathbf{L}^A = \mathbf{C} \odot (\mathbf{D}^B \mathbf{V}^B)$ ,  $\mathbf{L}^B = \mathbf{C} \odot (\mathbf{D}^A \mathbf{V}^A)$  et  $\mathbf{L}^C = (\mathbf{D}^B \mathbf{V}^B) \odot (\mathbf{D}^A \mathbf{V}^A)$ . Les matrices  $\mathbf{T}^A$ ,  $\mathbf{T}^B$ ,  $\mathbf{T}^C$  sont obtenues en dépliant le tenseur  $\mathcal{T}$  par rapport au premier, deuxième et troisième mode respectivement.  $\mathbf{1}_R$  est une matrice de taille  $(R, R)$  remplie de 1 qui correspond à la dérivée de la norme  $L_{1,1}$ .

Nous appelons cette approche Sparse NonNegative Canonical Polyadic Decomposition (SNNCPD). Les différentes étapes sont résumées dans l'algorithme 6. A noter qu'il est possible d'utiliser des approches plus sophistiquées pour résoudre notre optimisation contrainte comme ADMM [162] utilisé pour la factorisation matricielle en ligne ou APG [63].

---

**Algorithme 6** Sparse NonNegative Canonical Polyadic Decomposition (SNNCPD). Phase d'initialisation

---

*Entrée* :  $\mathcal{T}$ ,  $R$  surestimé

**while** un critère d'arrêt non vérifié **do**

Mise à jour des dictionnaires  $\mathbf{D}^A$  et  $\mathbf{D}^B$  avec Nadam optimizer utilisant les dérivées données dans l'équation (3.19)

Mise à jour des atomes  $\mathbf{V}^A$  et  $\mathbf{V}^B$  avec Nadam optimizer utilisant les dérivées données dans l'équation (3.19)

Mise à jour de la matrice  $\mathbf{C}$  avec Nadam optimizer utilisant les dérivées données dans l'équation (3.19)

**end while**

*Sortie* :  $\mathbf{D}^A$ ,  $\mathbf{D}^B$ ,  $\mathbf{C}$ ,  $\mathbf{V}^A$ ,  $\mathbf{V}^B$

---

Ainsi le nombre de colonnes non nulles de  $\mathbf{A}$  ou  $\mathbf{B}$  obtenues avec l'équation (3.17), donne une estimation *a posteriori* de  $\widetilde{R}_0$ . Au début du processus de minimisation, les atomes  $\mathbf{V}^A$  et  $\mathbf{V}^B$  sont initialisés comme matrices identités tandis que  $\mathbf{D}^A$  et  $\mathbf{D}^B$  et  $\mathbf{C}$  sont initialisés avec des valeurs aléatoires nonnégatives.

Afin d'assurer la nonnégativité des entrées de la matrice, tous les éléments des matrices  $\mathbf{D}^A$ ,  $\mathbf{D}^B$ ,  $\mathbf{C}$ ,  $\mathbf{V}^A$  et  $\mathbf{V}^B$  sont projetés sur  $\mathbb{R}^+$ . Cette phase est importante car nous démarrons sans information. Cela signifie que nous ne disposons pas des informations sur les matrices facteurs ni sur le rang. Cela nous permet également de travailler sur un domaine sur lequel la norme  $L_{1,1}$  est différentiable.

### 3.3 Décomposition de tenseurs en ligne avec contrainte de parcimonie et de nonnégativité. *Online Step (ou phase en ligne)*

Après l'initialisation des matrices à l'instant  $t = 0$ , nous exposons dans cette partie les algorithmes que l'on a développés pour la phase en ligne. En tout, nous avons développés 4 algorithmes pour la CPD en ligne :

- CPD utilisant une approche d'apprentissage par dictionnaire avec surestimation du rang. Approche sans variation de rang.
- Approche 1 : CPD utilisant une approche d'apprentissage par dictionnaire avec surestimation du rang.
- Approche 2 : CPD utilisant une approche d'apprentissage par transfert de dictionnaire.
- Approche 3 : CPD utilisant une approche récursive d'apprentissage de dictionnaire.

Le premier algorithme a une particularité par rapport au trois autres parce que, dans ce cas de figure, nous supposons qu'il n'y a pas de variation de rang au cours du temps c'est-à-dire que les signaux des matrices facteurs obtenues à l'instant  $t = 0$  ne varient pas au cours du temps (matrices  $\mathbf{A}, \mathbf{B}$ ). Dans ce cas de figure, nous considérons que la variation est appliquée sur la matrice  $\mathbf{C}$ .

#### 3.3.1 CPD utilisant une approche d'apprentissage par dictionnaire avec surestimation du rang. Approche sans variation de rang

Dans un premier temps, nous supposons que nous n'avons pas de variation du rang au cours du temps. Dans notre cas, si une telle hypothèse advient, cela signifie que la matrice  $\mathbf{C}$  qui correspond aux différentes contributions des fluorophores aura une de ses colonnes remplie de zéros ou alors, les contributions auront simplement changé de valeur. Cette approche est comparable avec la plupart des approches CPD en ligne proposées dans la littérature (cf section 2.4) qui ne tiennent pas en compte d'une quelconque variation de rang. Durant la phase en ligne, en supposant que les matrices facteurs sont bien estimées, nous pouvons reprendre l'équation 3.18 et écrire le nouveau problème à minimiser pour  $\mathcal{T} = \mathcal{T}_n$  :

$$\begin{aligned} \min \{ & E_{1,1}(\mathbf{V}^A, \mathbf{V}^B, \mathbf{C}_n) \} \text{ s.t. } \mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C} \geq 0 \\ \text{où } E_{1,1} = & \frac{1}{2} \|\mathcal{T} - \widehat{\mathcal{T}}(\mathbf{D}^A \mathbf{V}^A, \mathbf{D}^B \mathbf{V}^B, \mathbf{C}_n)\|_F^2 + \alpha \|\mathbf{V}_n^A\|_{1,1} + \alpha \|\mathbf{V}_n^B\|_{1,1} \end{aligned} \quad (3.20)$$

Le but est de fixer  $\mathbf{D}^A, \mathbf{D}^B$  déjà calculés à l'instant  $t = 0$  par l'algorithme 6 et de mettre à jour la nouvelle matrice  $\mathbf{C}$  ainsi que les poids  $\mathbf{V}_n^A, \mathbf{V}_n^B$  car la matrice  $\mathbf{C}$  change au cours du temps (nous la notons  $\mathbf{C}_n$ ) et  $\mathbf{V}_n^A, \mathbf{V}_n^B$  nos atomes, jouent le rôle de l'indétermination des permutations. L'avantage de cette approche est la convergence rapide car nous avons moins de matrices à estimer et aussi la bonne estimation des matrices facteurs ainsi que le rang. Nous appelons cet algorithme Online Sparse and Nonnegative for Dictionary Tensor Learning (OSNDTL). Les différents étapes sont résumées dans l'algorithme 7.

Bien que cet algorithme soit rapide en termes de convergence, il ne prend pas en compte la variation de rang. Pour pallier cela, nous proposons trois algorithmes qui prennent en compte la variation du rang tout en se basant les matrices facteurs obtenues à l'instant  $t_{n-1}$ .

---

**Algorithme 7** Online Sparse and Nonnegative for Dictionary Tensor Learning
 

---

**• STEP 1 : Phase d'initialisation**

 Entrée :  $\mathcal{T}_0$ ,  $R$  surestimé

 Résoudre Eq. (3.18) avec  $\mathcal{T} = \mathcal{T}_0$  à l'aide de l'algorithme 6 (SNNCPD)

 Calcul de  $\mathbf{A}_0$  et  $\mathbf{B}_0$  utilisant l'Eq. (3.17)

 Sortie :  $\mathbf{A}_0$ ,  $\mathbf{B}_0$  et  $\mathbf{C}_0$ 
**• STEP 2 : Phase en ligne**
**while** un critère d'arrêt non vérifié **do**

Fixer les dictionnaires

 Mise à jour des atomes  $\mathbf{V}^A$  et  $\mathbf{V}^B$  avec Nadam utilisant  
 les dérivées données dans l'équation (3.19)

 Mise à jour de la matrice  $\mathbf{C}_n$  avec Nadam utilisant  
 les dérivées données dans l'équation (3.19)

**end while**

 Sortie :  $\mathbf{C}_n$ ,  $\mathbf{V}_n^A$ ,  $\mathbf{V}_n^B$ 


---

### 3.3.2 Approche 1 : CPD basée sur l'apprentissage par dictionnaire avec surestimation du rang.

Cette approche a pour finalité de prendre en compte la variation du rang au cours du temps. Nous montrons maintenant comment calculer les matrices facteurs à l'instant  $t_n$  à partir des matrices facteurs à l'instant  $t_{n-1}$ . Nous ne recherchons donc ici que l'apparition de nouveaux facteurs ou la disparition d'anciens. Dans ce but, nous définissons d'abord  $\mathbf{D}^A = \mathbf{A}_{n-1}$  et  $\mathbf{D}^B = \mathbf{B}_{n-1}$  comme initialisation avant le processus d'optimisation. Ensuite, nous résolvons l'équation (3.18) pour  $\widehat{\mathcal{T}} = \widehat{\mathcal{T}}_n$ . Cependant, au lieu de mettre à jour les matrices entières, nous gardons cette fois les colonnes non nulles de  $\mathbf{D}^A$  et  $\mathbf{D}^B$  inchangées tout au long du processus d'optimisation. On a donc  $R - \widetilde{R}_n$  colonnes nulles à optimiser comme par exemple pour  $\widetilde{R}_{n-1} = 3$

$$\mathbf{D}^A = \mathbf{A}_{n-1} = \begin{array}{c} \overbrace{\hspace{10em}}^R \\ \overbrace{\hspace{3em}}^{\text{fixées}} \quad \overbrace{\hspace{3em}}^{\text{à optimiser}} \\ \left( \begin{array}{cccccc} A_{11} & A_{12} & A_{13} & ? & \dots & ? \\ A_{21} & A_{22} & A_{23} & ? & \dots & ? \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{J1} & A_{J2} & A_{J3} & ? & \dots & ? \end{array} \right) \end{array}$$

En effet, premièrement, les colonnes fixes correspondent aux facteurs estimés précédemment. Deuxièmement, nous gardons les indices de ces dits facteurs. Troisièmement, d'un point de vue analytique, nous annulons le gradient des matrices facteurs obtenu à l'instant  $t_{n-1}$  grâce aux indices collectés. Le gradient étant annulé, cela induit que la matrice facteur en question est minimisée, ce qui nous permet de minimiser les autres matrices facteurs.

Après convergence, si un de ces facteurs a disparu à l'instant  $t_n$  on devrait alors obtenir un zéro dans  $\mathbf{V}^A$  et  $\mathbf{V}^B$ . Afin de tenir compte de l'indétermination des permutations, après chaque décomposition en sous-tenseurs, les colonnes de la matrice  $\mathbf{B}$  (associées aux spectres d'émission) sont permutées de manière à ce que les positions des maxima de chaque colonne soient triées par ordre croissant. Cette permutation est ensuite appliquée aux modes  $\mathbf{A}$  et  $\mathbf{C}$ . Nous appelons cet

algorithme Online Sparse and NonNegative CPD 1 (OSNNCPD 1). Les différentes étapes sont résumées dans l'algorithme 8.

---

**Algorithme 8** Online Sparse and Nonnegative CPD 1
 

---

**• STEP 1 : Phase d'initialisation**

Entrée :  $\mathcal{T}_0$ ,  $R$  surestimé

Résoudre Eq. (3.18) avec  $\mathcal{T} = \mathcal{T}_0$  à l'aide de l'algorithme 6 (SNNCPD)

Calcul de  $\mathbf{A}_0$  et  $\mathbf{B}_0$  utilisant l'Eq. (3.17)

Sortie :  $\mathbf{A}_0$ ,  $\mathbf{B}_0$  et  $\mathbf{C}_0$

**• STEP 2 : Phase en ligne**

Entrée :  $\mathcal{T}_n$  et  $\mathbf{A}_{n-1}$ ,  $\mathbf{B}_{n-1}$

Initialisation :  $\mathbf{D}^A = \mathbf{A}_{n-1}$ ,  $\mathbf{D}^B = \mathbf{B}_{n-1}$

Remplir les colonnes nulles de  $\mathbf{D}^A$  et  $\mathbf{D}^B$  par des nombres aléatoires, en excluant les autres colonnes du problème d'optimisation de l'Eq. (3.18),

Résoudre :  $\mathcal{T} = \mathcal{T}_n$

Mise à jour des matrices facteurs actuelles :  $\mathbf{A}_n = \mathbf{D}^A \mathbf{V}^A$  et  $\mathbf{B}_n = \mathbf{D}^B \mathbf{V}^B$

Sortie :  $\mathbf{A}_n$ ,  $\mathbf{B}_n$ ,  $\mathbf{C}_n$  et  $\tilde{R}_n$

**• STEP 3 : Retour à l'étape 2 avec  $n=n+1$** 


---

### 3.3.3 Approche 2 : CPD basée sur le transfert de dictionnaire.

Pour le deuxième algorithme on considère que les facteurs estimés à l'instant  $t_n$  (depuis  $\mathcal{T}_{n-1}$ ) peuvent être améliorés et on choisit donc de mettre à jour l'ensemble des matrices  $\mathbf{A}$  et  $\mathbf{B}$ . Nous remplaçons d'abord les colonnes nulles de  $\mathbf{A}_{n-1}$  et  $\mathbf{B}_{n-1}$  par des colonnes de nombres aléatoires. Ensuite on calcule  $\mathbf{A}_n$  et  $\mathbf{B}_n$  par combinaisons linéaires des entrées de  $\mathbf{A}_{n-1}$  et  $\mathbf{B}_{n-1}$  et nous sélectionnons des facteurs mis à jour :

$$\mathbf{A}_n = \mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^A \text{ et } \mathbf{B}_n = \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B. \quad (3.21)$$

où  $\mathbf{U}^A$  et  $\mathbf{U}^B$  sont deux matrices carrées de tailles respectives  $I$  et  $J$ . Les produits  $\mathbf{U}^A \mathbf{A}_{n-1}$  et  $\mathbf{U}^B \mathbf{B}_{n-1}$  jouent le rôle des dictionnaires alors que les matrices  $\mathbf{V}^A$  et  $\mathbf{V}^B$  sont nos atomes. Le problème d'optimisation devient donc pour  $\mathcal{T} = \mathcal{T}_n$  :

$$\begin{aligned} & \min \left\{ E_2(\mathbf{U}_n^A, \mathbf{V}_n^A, \mathbf{U}_n^B, \mathbf{V}_n^B, \mathbf{C}) \right\} \text{ s.t } \mathbf{U}_n^A, \mathbf{U}_n^B, \mathbf{C}, \mathbf{V}_n^A, \mathbf{V}_n^B \geq 0 \\ & \text{où } E_2 = \frac{1}{2} \left\| \mathcal{T} - \widehat{\mathcal{T}}(\mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^A, \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B, \mathbf{C}) \right\|_F^2 + \alpha \|\mathbf{V}_n^A\|_{1,1} + \alpha \|\mathbf{V}_n^B\|_{1,1} \end{aligned} \quad (3.22)$$

Le gradient de la fonction de coût (3.22) est donné par :

$$\begin{aligned}
 \frac{\partial E_2}{\partial \mathbf{V}_n^A} &= -(\mathbf{U}_n^A \mathbf{A}_{n-1})^\top (\mathbf{T}^A - \mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^A (\mathbf{Z}^A)^\top) \mathbf{Z}^A + \alpha \mathbf{1}_R \\
 \frac{\partial E_2}{\partial \mathbf{V}_n^B} &= -(\mathbf{U}_n^B \mathbf{B}_{n-1})^\top (\mathbf{T}^B - \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B (\mathbf{Z}^B)^\top) \mathbf{Z}^B + \alpha \mathbf{1}_R \\
 \frac{\partial E_2}{\partial \mathbf{U}_n^A} &= -(\mathbf{T}^A - \mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^B (\mathbf{Z}^A)^\top) \mathbf{Z}^A (\mathbf{V}_n^A)^\top (\mathbf{A}_{n-1})^\top \\
 \frac{\partial E_2}{\partial \mathbf{U}_n^B} &= -(\mathbf{T}^B - \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B (\mathbf{Z}^B)^\top) \mathbf{Z}^B (\mathbf{V}_n^B)^\top (\mathbf{B}_{n-1})^\top \\
 \frac{\partial E_2}{\partial \mathbf{C}} &= -(\mathbf{T}^C - \mathbf{C} (\mathbf{Z}^C)^\top) \mathbf{Z}^C
 \end{aligned} \tag{3.23}$$

avec  $\mathbf{Z}^A = \mathbf{C} \odot (\mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B)$ ;  $\mathbf{Z}^B = \mathbf{C} \odot (\mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^A)$  et  $\mathbf{Z}^C = (\mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B) \odot (\mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^A)$ . Au début du processus de minimisation,  $\mathbf{U}^A$ ,  $\mathbf{U}^B$ ,  $\mathbf{V}^A$  et  $\mathbf{V}^B$  sont initialisées comme matrices identités.

L'avantage de cette approche est donc sa flexibilité. L'inconvénient est que les matrices à optimiser sont plus grandes ce qui rend donc l'algorithme plus coûteux en temps. Nous appelons cet algorithme Online Sparse and NonNegative CPD 2 (OSNNCPD 2). Les différentes étapes sont résumées dans l'algorithme 9. L'indétermination de la permutation entre les sous-tenseurs est résolue de la même manière que dans OSNNCPD1.

---

**Algorithme 9** Online Sparse and Nonnegative CPD 2
 

---

**• STEP 1 : Phase d'initialisation**

Identique à l'initialisation de l'algorithme 8

**• STEP 2 : Phase en ligne**

 Entrée :  $\mathcal{T}_n$ , et  $\mathbf{A}_{n-1}$ ,  $\mathbf{B}_{n-1}$ 

 Remplir les colonnes nulles de  $\mathbf{A}_{n-1}$ ,  $\mathbf{B}_{n-1}$  par des nombres aléatoires et résoudre l'Eq. (3.22) pour  $\mathcal{T} = \mathcal{T}_n$ 

Mise à jour des matrices facteurs actuelles avec l'Eq. (3.21)

 Sortie :  $\mathbf{A}_n$ ,  $\mathbf{B}_n$ ,  $\mathbf{C}_n$  and  $\tilde{R}_n$ 
**• STEP 3 : Retour à l'étape 2 avec  $n=n+1$** 


---

### 3.3.4 Extension de l'approche 2 : CPD basée sur le transfert de dictionnaire

En prenant l'équation 3.21 ainsi que la fonction de coût 3.22, l'optimisation de l'algorithme devient coûteuse à cause des matrices dictionnaires  $\mathbf{U}^A$  et  $\mathbf{U}^B$ . Cela est causé par le fait qu'aucune contrainte n'est appliquée sur ces matrices pour restreindre l'espace de recherche. Dans cette approche, nous allons appliquer une contrainte d'orthogonalité sur ces dictionnaires. Pour  $\mathcal{T} = \mathcal{T}_n$ , la fonction de coût devient :

$$\begin{aligned}
 &\min \left\{ E_2(\mathbf{U}_n^A, \mathbf{V}_n^A, \mathbf{U}_n^B, \mathbf{V}_n^B, \mathbf{C}) \right\} \\
 &\text{où } E_2 = \frac{1}{2} \left\| \mathcal{T} - \widehat{\mathcal{T}}(\mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^B, \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^A, \mathbf{C}) \right\|_F^2 + \alpha \|\mathbf{V}_n^A\|_{1,1} + \alpha \|\mathbf{V}_n^B\|_{1,1} + \\
 &\|\mathbf{I}_A - (\mathbf{U}_n^A)^T \mathbf{U}_n^A\|_F^2 + \|\mathbf{I}_B - (\mathbf{U}_n^B)^T \mathbf{U}_n^B\|_F^2
 \end{aligned} \tag{3.24}$$

Le gradient de la fonction de coût (3.22) est donné par :

$$\begin{aligned}
\frac{\partial E_2}{\partial \mathbf{V}_n^A} &= -(\mathbf{U}_n^A \mathbf{A}_{n-1})^\top (\mathbf{T}^A - \mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^A (\mathbf{Z}^A)^\top) \mathbf{Z}^A + \alpha \mathbf{1}_R \\
\frac{\partial E_2}{\partial \mathbf{V}_n^B} &= -(\mathbf{U}_n^B \mathbf{B}_{n-1})^\top (\mathbf{T}^B - \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B (\mathbf{Z}^B)^\top) \mathbf{Z}^B + \alpha \mathbf{1}_R \\
\frac{\partial E_2}{\partial \mathbf{U}_n^A} &= -(\mathbf{T}^A - \mathbf{U}_n^A \mathbf{A}_{n-1} \mathbf{V}_n^B (\mathbf{Z}^A)^\top) \mathbf{Z}^A (\mathbf{V}_n^A)^\top (\mathbf{A}_{n-1})^\top - (2\mathbf{U}_n^A (\mathbf{I}_A^\top - (\mathbf{U}_n^A)^\top \mathbf{U}_n^A) + \\
&\quad (2\mathbf{U}_n^A (\mathbf{I}_A \mathbf{U}_n^A - (\mathbf{U}_n^A)^\top \mathbf{U}_n^A))) \\
\frac{\partial E_2}{\partial \mathbf{U}_n^B} &= -(\mathbf{T}^B - \mathbf{U}_n^B \mathbf{B}_{n-1} \mathbf{V}_n^B (\mathbf{Z}^B)^\top) \mathbf{Z}^B (\mathbf{V}_n^B)^\top (\mathbf{B}_{n-1})^\top - (2\mathbf{U}_n^B (\mathbf{I}_B^\top - (\mathbf{U}_n^B)^\top \mathbf{U}_n^B) + \\
&\quad (2\mathbf{U}_n^B (\mathbf{I}_B \mathbf{U}_n^B - (\mathbf{U}_n^B)^\top \mathbf{U}_n^B))) \\
\frac{\partial E_2}{\partial \mathbf{C}} &= -(\mathbf{T}^C - \mathbf{C} (\mathbf{Z}^C)^\top) \mathbf{Z}^C
\end{aligned} \tag{3.25}$$

L'avantage de cette approche est sa rapidité de convergence car nous réduisons l'espace de recherche de nos matrices dictionnaires. Par exemple, si  $\mathbf{A}_n$  est similaire à  $\mathbf{A}_{n-1}$ , la matrice  $\mathbf{U}_n^A$  est une matrice diagonale. Nous appelons cet algorithme Online Orthogonality Sparse CPD (OOSCPD). Les différentes étapes sont similaires avec l'algorithme 9. L'indétermination de la permutation entre les sous-tenseurs est résolue de la même manière que dans OSNNCPD1.

### 3.3.5 CPD utilisant une approche récursive d'apprentissage de dictionnaire. Approche 3

Cette troisième approche est toujours basée sur l'apprentissage de dictionnaires. Cependant, elle doit permettre un gain en rapidité important lors de l'optimisation car elle est basée sur un calcul récursif du gradient à l'aide des matrices facteurs obtenues à l'instant  $t_{n-1}$ . Cela devient facile à mettre en place car nous utilisons la famille de descente de gradient. En plus du temps de calcul réduit, cette approche permet de garder le lien entre deux sous-tenseurs consécutifs ce qui nous permet de ne pas sauvegarder les données à chaque fois.

La différence avec l'approche que nous proposons ici est que dans [165, 162], les auteurs traitent le problème d'une factorisation matricielle en ligne alors que dans [119] les auteurs traitent des tenseurs en ligne mais les méthodes proposées ne tiennent pas compte de la variation de rang qui peut se produire dans notre problème. Enfin, dans [45], les auteurs traitent de la décomposition Tucker. Cette approche nous permet donc d'inférer une base génératrice de facteurs latents en traitant les données au fur et à mesure de leur acquisition et de prendre en compte la variation de rang.

L'idée principale est de calculer chaque matrice facteur à l'instant  $t_n$  en utilisant le gradient de chaque matrice facteur calculé au temps  $t_{n-1}$  obtenu à la dernière itération  $k$  du processus d'optimisation. Prenons l'équation (3.18) pour  $\widehat{\mathcal{T}} = \widehat{\mathcal{T}}_0$ , nous la remplaçons ici par :

$$\begin{aligned}
&\min \left\{ E_3(\mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C}) \right\} \text{ s.t. } \mathbf{D}^A, \mathbf{V}^A, \mathbf{D}^B, \mathbf{V}^B, \mathbf{C} \geq 0 \\
&\text{où } E_3 = \frac{1}{n+1} \sum_{i=0}^n \frac{1}{2} \|\mathcal{T}_i - \widehat{\mathcal{T}}_i(\mathbf{D}^A \mathbf{V}^A, \mathbf{D}^B \mathbf{V}^B, \mathbf{C})\|_F^2 + \alpha \|\mathbf{V}^A\|_{1,1} + \alpha \|\mathbf{V}^B\|_{1,1}
\end{aligned} \tag{3.26}$$

où  $n$  est le numéro du sous-tenseur courant cela nous permet de plus de tenir compte de toutes les données passées. A l'instant  $t_0$ ,  $n = 0$ , nous sommes dans la phase *Initialisation* et cela conduit à

résoudre l'équation (3.18). On sait que dans notre cas, les informations intéressantes sont contenues dans les dictionnaires  $\mathbf{D}^A$  et  $\mathbf{D}^B$ . Ainsi, pour  $n = 0$  les gradients par rapport aux dictionnaires  $\mathbf{D}^A$ ,  $\mathbf{D}^B$  sont données par :

$$\begin{aligned}\frac{\partial E_3}{\partial \mathbf{D}^A} &= -\mathbf{T}^A \mathbf{L}^A (\mathbf{V}^A)^\top + \mathbf{D}^A \mathbf{V}^A (\mathbf{L}^A)^\top \mathbf{L}^A (\mathbf{V}^A)^\top \text{ et} \\ \frac{\partial E_3}{\partial \mathbf{D}^B} &= -\mathbf{T}^B \mathbf{L}^B (\mathbf{V}^B)^\top + \mathbf{D}^B \mathbf{V}^B (\mathbf{L}^B)^\top \mathbf{L}_2 (\mathbf{V}^B)^\top\end{aligned}\quad (3.27)$$

avec  $\mathbf{L}^A = \mathbf{C} \odot \mathbf{D}^B \mathbf{V}^B$  et  $\mathbf{L}^B = \mathbf{D}^A \mathbf{V}^A \odot \mathbf{C}$ .

Dans la phase en ligne c'est à dire  $\forall n > 0$  :

$$\begin{aligned}\frac{\partial E_3}{\partial \mathbf{D}^A} &= -\frac{1}{n+1} \sum_{i=0}^n \mathbf{T}_i^A \mathbf{L}_i^A (\mathbf{V}_i^A)^\top + \frac{\mathbf{D}_i^A}{n+1} \sum_{i=0}^n \mathbf{V}_i^A (\mathbf{L}_i^A)^\top \mathbf{L}_i^A (\mathbf{V}_i^A)^\top \text{ et} \\ \frac{\partial E_3}{\partial \mathbf{D}^B} &= -\frac{1}{n+1} \sum_{i=0}^n \mathbf{T}_i^B \mathbf{L}_i^B (\mathbf{V}_i^B)^\top + \frac{\mathbf{D}_i^B}{n+1} \sum_{i=0}^n \mathbf{V}_i^B (\mathbf{L}_i^B)^\top \mathbf{L}_i^B (\mathbf{V}_i^B)^\top.\end{aligned}\quad (3.28)$$

Posons :

$$\begin{aligned}\mathbf{P}_n^A &= \sum_{i=0}^n \mathbf{T}_i^A \mathbf{L}_i^A (\mathbf{V}_i^A)^\top \\ \mathbf{P}_n^B &= \sum_{i=0}^n \mathbf{T}_i^B \mathbf{L}_i^B (\mathbf{V}_i^B)^\top \\ \mathbf{Q}_n^A &= \sum_{i=0}^n \mathbf{V}_i^A (\mathbf{L}_i^A)^\top \mathbf{L}_i^A (\mathbf{V}_i^A)^\top \\ \mathbf{Q}_n^B &= \sum_{i=0}^n \mathbf{V}_i^B (\mathbf{L}_i^B)^\top \mathbf{L}_i^B (\mathbf{V}_i^B)^\top.\end{aligned}\quad (3.29)$$

On peut réécrire ces dérivées sous la forme :

$$\frac{\partial E_3}{\partial \mathbf{D}^A} = -\frac{\mathbf{P}_n^A}{n+1} + \frac{\mathbf{D}^A \mathbf{Q}_n^A}{n+1} \text{ et } \frac{\partial E_3}{\partial \mathbf{D}^B} = -\frac{\mathbf{P}_n^B}{n+1} + \frac{\mathbf{D}^B \mathbf{Q}_n^B}{n+1}\quad (3.30)$$

On veut utiliser les matrices  $\mathbf{P}_{n-1}$  et  $\mathbf{Q}_{n-1}$  pour calculer  $\mathbf{P}_n$  et  $\mathbf{Q}_n$ . Pour cela, nous pouvons réécrire  $\mathbf{P}_n^A$ ,  $\mathbf{P}_n^B$ ,  $\mathbf{Q}_n^A$  et  $\mathbf{Q}_n^B$  comme :

$$\begin{aligned}\mathbf{P}_n^A &= \sum_{i=0}^n \mathbf{T}_i^A \mathbf{L}_i^A (\mathbf{V}_i^A)^\top = \mathbf{T}_1^A \mathbf{L}_1^A (\mathbf{V}_1^A)^\top + \mathbf{T}_2^A \mathbf{L}_2^A (\mathbf{V}_2^A)^\top + \dots + \mathbf{T}_{n-1}^A \mathbf{L}_{n-1}^A (\mathbf{V}_{n-1}^A)^\top + \mathbf{T}_n^A \mathbf{L}_n^A (\mathbf{V}_n^A)^\top, \\ \mathbf{P}_n^B &= \sum_{i=0}^n \mathbf{T}_i^B \mathbf{L}_i^B (\mathbf{V}_i^B)^\top = \mathbf{T}_1^B \mathbf{L}_1^B (\mathbf{V}_1^B)^\top + \mathbf{T}_2^B \mathbf{L}_2^B (\mathbf{V}_2^B)^\top + \dots + \mathbf{T}_{n-1}^B \mathbf{L}_{n-1}^B (\mathbf{V}_{n-1}^B)^\top + \mathbf{T}_n^B \mathbf{L}_n^B (\mathbf{V}_n^B)^\top,\end{aligned}$$

$$\begin{aligned}\mathbf{Q}_n^A &= \sum_{i=0}^n \mathbf{V}_i^A (\mathbf{L}_i^A)^\top (\mathbf{L}_i^A) (\mathbf{V}_i^A)^\top = \mathbf{V}_1^A (\mathbf{L}_1^A)^\top (\mathbf{L}_1^A) (\mathbf{V}_1^A)^\top + \mathbf{V}_2^A (\mathbf{L}_2^A)^\top (\mathbf{L}_2^A) (\mathbf{V}_2^A)^\top + \dots + \\ &\mathbf{V}_{n-1}^A (\mathbf{L}_{n-1}^A)^\top (\mathbf{L}_{n-1}^A) (\mathbf{V}_{n-1}^A)^\top + \mathbf{V}_n^A (\mathbf{L}_n^A)^\top (\mathbf{L}_n^A) (\mathbf{V}_n^A)^\top,\end{aligned}$$

$$\begin{aligned}\mathbf{Q}_n^B &= \sum_{i=0}^n \mathbf{V}_i^B (\mathbf{L}_i^B)^\top (\mathbf{L}_i^B) (\mathbf{V}_i^B)^\top = \mathbf{V}_1^B (\mathbf{L}_1^B)^\top (\mathbf{L}_1^B) (\mathbf{V}_1^B)^\top + \mathbf{V}_2^B (\mathbf{L}_2^B)^\top (\mathbf{L}_2^B) (\mathbf{V}_2^B)^\top + \dots + \\ &\mathbf{V}_{n-1}^B (\mathbf{L}_{n-1}^B)^\top (\mathbf{L}_{n-1}^B) (\mathbf{V}_{n-1}^B)^\top + \mathbf{V}_n^B (\mathbf{L}_n^B)^\top (\mathbf{L}_n^B) (\mathbf{V}_n^B)^\top.\end{aligned}$$

Ainsi les matrices  $\mathbf{P}_n$  et  $\mathbf{Q}_n$  peuvent être écrites en fonction de  $\mathbf{P}_{n-1}$  et  $\mathbf{Q}_{n-1}$  comme :

$$\begin{aligned}\mathbf{P}_n^A &= \mathbf{P}_{n-1}^A + \mathbf{T}_n^A \mathbf{L}_n^A (\mathbf{V}_n^A)^\top; \quad \mathbf{Q}_n^A = \mathbf{Q}_{n-1}^A + \mathbf{V}_n^A (\mathbf{L}_n^A)^\top (\mathbf{L}_n^A) (\mathbf{V}_n^A)^\top \\ \mathbf{P}_n^B &= \mathbf{P}_{n-1}^B + \mathbf{T}_n^B \mathbf{L}_n^B (\mathbf{V}_n^B)^\top; \quad \mathbf{Q}_n^B = \mathbf{Q}_{n-1}^B + \mathbf{V}_n^B (\mathbf{L}_n^B)^\top (\mathbf{L}_n^B) (\mathbf{V}_n^B)^\top\end{aligned}\quad (3.31)$$

Les éléments des suites  $\mathbf{P}_n$  et  $\mathbf{Q}_n$  peuvent être mis à jour de manière récursive, lors du processus d'optimisation par les matrices obtenues précédemment. Ensuite, les gradients sont calculés à l'aide des matrices dépliées du sous-tenseur  $\mathcal{T}_n$  ( $\mathbf{T}_n^A$  et  $\mathbf{T}_n^B$ ) et des matrices  $\mathbf{P}_n$  et  $\mathbf{Q}_n$  en utilisant l'équation (3.31). Les atomes  $\mathbf{V}^A$ ,  $\mathbf{V}^B$  et la matrice  $\mathbf{C}$  sont mis à jour à l'aide de l'algorithme *Nadam* à chaque itération afin de gérer la variation de rang. Nous ne calculons pas récursivement la matrice  $\mathbf{C}$  dans notre cas car elle ne contient aucune information sur la forme des spectres. Nous appelons cet algorithme Recursive Online Sparse NonNegative CPD (ROSNNCPD). Les différentes étapes sont résumées dans l'algorithme 10. Encore une fois, l'indétermination de permutation est résolue de la même manière que dans OSNNCPD1.

---

**Algorithme 10** Recursive Online Sparse Nonnegative CPD (ROSNNCPD)
 

---

**• STEP 1 : Phase d'initialisation**

Identique à l'initialisation de l'algorithme 8

 Calcul de  $\mathbf{P}_0^A, \mathbf{Q}_0^A, \mathbf{P}_0^B, \mathbf{Q}_0^B$  avec l'Eq. (3.31)

**• STEP 2 : Phase en ligne**
*Entrée* :  $n, \mathcal{T}_n, \mathbf{A}_{n-1}, \mathbf{B}_{n-1}, \mathbf{P}_{n-1}^A, \mathbf{Q}_{n-1}^A, \mathbf{P}_{n-1}^B, \mathbf{Q}_{n-1}^B$ 
*Initialize*  $\mathbf{D}^A = \mathbf{A}_{n-1}, \mathbf{D}^B = \mathbf{B}_{n-1},$ 
**while** un critère d'arrêt non vérifié **do**

 Calcul de  $\mathbf{P}_n^A, \mathbf{Q}_n^A, \mathbf{P}_n^B, \mathbf{Q}_n^B$  utilisant l'Eq. (3.31)

 Calcul de  $\frac{\partial E_3}{\partial \mathbf{D}^A}$  et  $\frac{\partial E_3}{\partial \mathbf{D}^B}$  défini dans l'Eq. (3.30)

 Mise à jour des dictionnaires  $\mathbf{D}^A$  et  $\mathbf{D}^B$  avec  $\frac{\partial E_3}{\partial \mathbf{D}^A}$  et  $\frac{\partial E_3}{\partial \mathbf{D}^B}$  utilisant *Nadam*

 Mise à jour des atomes  $\mathbf{V}^A$  et  $\mathbf{V}^B$  utilisant *Nadam*

 Mise à jour de  $\mathbf{C}$  utilisant *Nadam*
**end while**

 Mise à jour des matrices facteurs  $\mathbf{A}_n = \mathbf{D}^A \mathbf{V}^A$  et  $\mathbf{B}_n = \mathbf{D}^B \mathbf{V}^B$ 
*Sortie* :  $\mathbf{A}_n, \mathbf{B}_n, \mathbf{C}_n, \mathbf{P}_n^A, \mathbf{Q}_n^A, \mathbf{P}_n^B, \mathbf{Q}_n^B$  et  $\tilde{\mathbf{R}}_n$ 
**• STEP 3 : Retour à l'étape 2 avec  $n=n+1$** 


---

### 3.4 Conclusion du chapitre

Dans ce chapitre, nous avons proposé cinq algorithmes (SNNCPD, OSNDTL, OSNNCPD1, OSNNCPD2, ROSNNCPD) au total de décomposition de tenseurs dont les quatre derniers sont consacrés à la décomposition en ligne. OSNNCPD1, OSNNCPD2 et ROSNNCPD peuvent prendre en considération une éventuelle variation du rang tandis que OSNDTL ne prend pas en considération ce problème mais peut prendre néanmoins en considération une disparition d'un élément déjà présent. Ces méthodes sont inspirées de méthodes de décomposition en ligne pour les matrices et utilisent l'approche de l'apprentissage de dictionnaires. Ces approches sont flexibles pour incorporer certaines contraintes classiques telles que la nonnégativité et la parcimonie.

Dans le chapitre suivant, nous présenterons différentes simulations effectuées en laboratoire sur des données synthétiques et réelles de fluorescence afin de comparer les algorithmes développés. Nous détaillerons les outils et les méthodes utilisés pour acquérir ces données et nous présenterons les résultats que donnent nos approches sur ces dites données.



# Chapitre 4

## Application sur des données de fluorescence simulées et de laboratoire

Dans ce chapitre, il est question d'évaluer les performances des algorithmes décrits au chapitre précédent. Pour cela, nous les testons sur des données synthétiques de fluorescence simulées pour la CPD en ligne d'une part et d'autre part sur des données de fluorescence réelles acquises en ligne lors d'une expérience en laboratoire. Le but des simulations synthétiques est de calibrer les différents hyperparamètres de nos algorithmes afin d'améliorer leurs performances avant de les appliquer à des données de fluorescence réelles contrôlées. Dans les deux types de données (synthétiques ou réelles contrôlées), nous simulons les phénomènes réels se reproduisant *in situ* comme la disparition et l'apparition d'un ou de plusieurs fluorophores (ou composants). Durant la simulation sur les données synthétiques, nous utilisons uniquement les approches proposées au chapitre précédent. Par contre, nous étendons la comparaison à d'autres algorithmes pour les données réelles contrôlées [166].

## 4.1 Simulation des données synthétiques de fluorescence

Cette partie décrit les étapes de génération des données synthétiques de fluorescence. Nous simulons en tout trois différentes séries de 1000 MEEFs avec chacune une particularité différente en fonction des hypothèses : "Apparition" et "Disparition" de fluorophores. Toutes les MEEFs synthétiques suivent strictement le modèle trilineaire (les pics de diffusion ne sont pas présents et aucun bruit n'altère les MEEFs) et correspondent donc à l'addition de  $R$  composants fluorescents qui dépendent uniquement de  $R$  spectres d'excitation et d'émission (matrices  $\mathbf{A}$  et  $\mathbf{B}$  de dimensions respectives  $I \times R$  et  $J \times R$ ) et qui sont pondérés par des concentrations (contenues dans la matrice  $\mathbf{C}$  de dimensions  $K \times R$ ). Le tenseur  $\mathcal{T}$  généré a donc pour rang  $R$  et a pour dimensions  $I \times J \times K$ . Dans le cas des simulations présentées ici, le nombre  $K$  de MEEF est égal à 1000 et le rang  $\tilde{R}$  est inférieur ou égal à 5. Ci-dessus, nous présentons les différents spectres d'excitation et d'émission ainsi que les profils de concentrations simulés.

### 4.1.1 Présentation de l'acquisition des données synthétiques de fluorescence

#### Spectres

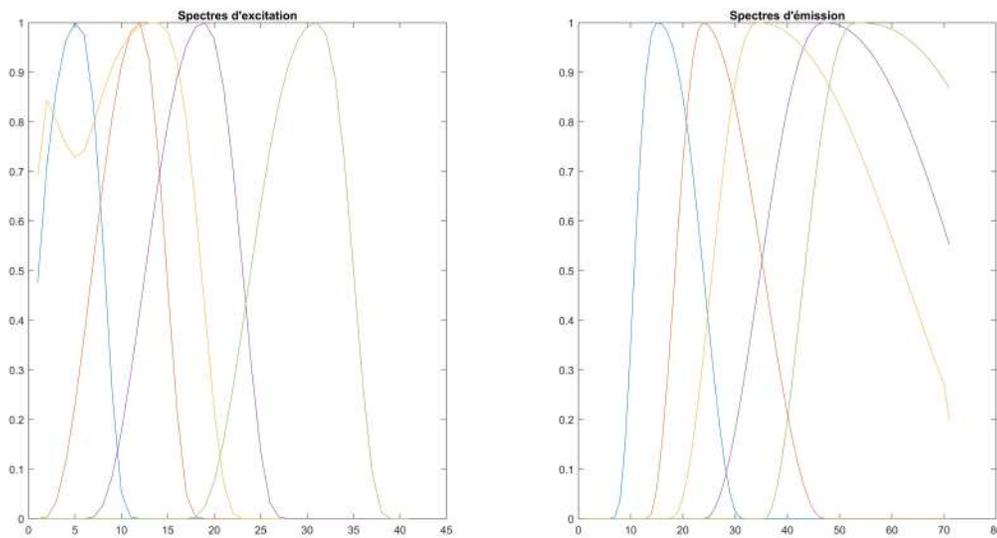


FIGURE 4.1 – Spectres excitation-émission

Les spectres d'excitation simulés comptent 41 points et les spectres d'émission 71 points. Comme les spectres réels de fluorophores naturels, les spectres d'excitation et d'émission synthétiques sont asymétriques. Dans ces simulations, tous les spectres d'émission sont monomodaux alors que les spectres d'excitation peuvent être soit monomodaux soit bimodaux (4.1). Dans le cas particulier de ces premières simulations, chaque spectre monomodal est défini par cinq paramètres : la longueur d'onde centrale associée au maximum du spectre, les largeurs gauche et droite du support à partir de la longueur d'onde centrale, et deux autres paramètres permettant de modifier la forme du spectre. Un spectre d'excitation bimodal est modélisé simplement en combinant linéairement deux spectres monomodaux. Tous les spectres sont normalisés par rapport à la norme  $L_\infty$  (leur maximum est égal à un) avant d'être placés dans les matrices  $\mathbf{A}$  et  $\mathbf{B}$ .

## Profils de concentrations simulés

Les concentrations qui constituent les matrices  $C$  des 3 séries sont générées en sommant trois signaux  $s_k(t)$  qui dépendent de trois distributions aléatoires différentes  $D_k(t)$  qui sont de la forme suivante :

$$D_k(t) = \sum_i a_{ij} \delta(t - t_i) \quad (4.1)$$

où  $\delta(t - t_i)$  représente la distribution de Dirac centrée sur l'instant  $t_i$  pondéré par l'amplitude  $a_i$  ( $t_i$  et  $a_i$  sont des variables aléatoires de distribution uniforme). La Figure 4.2 montre trois exemples de distributions  $D_k(t)$  (tracés rouges).

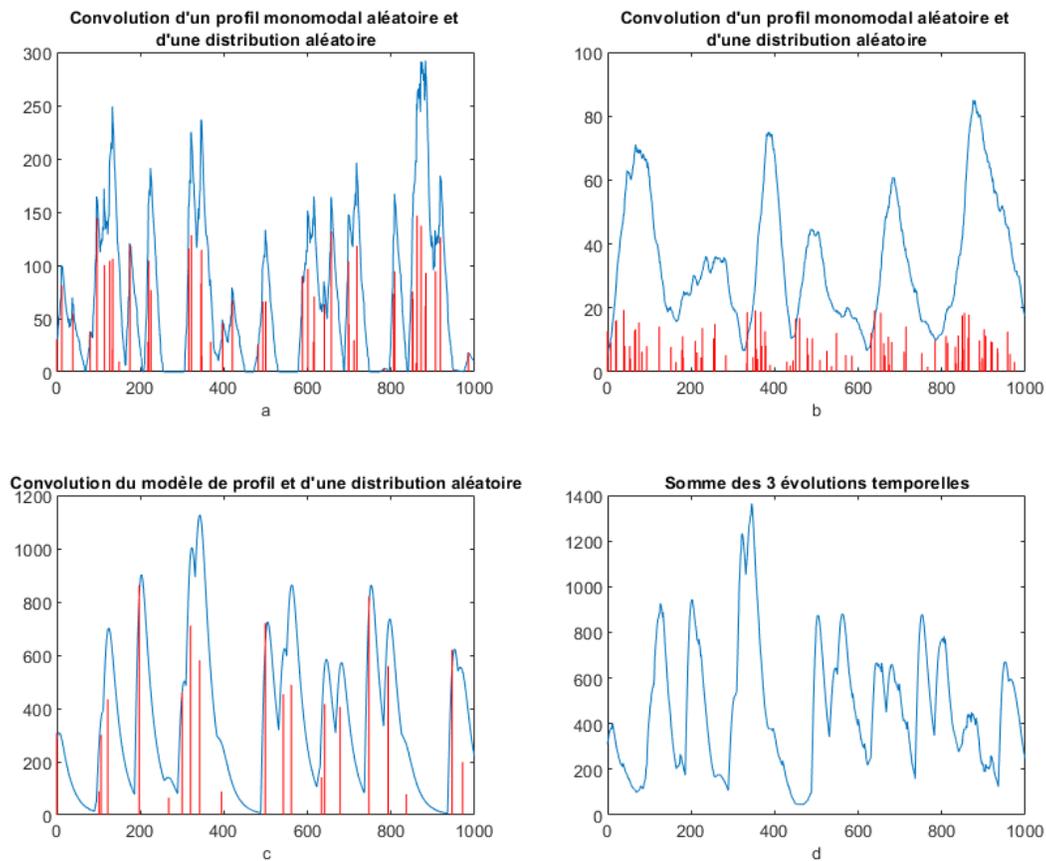
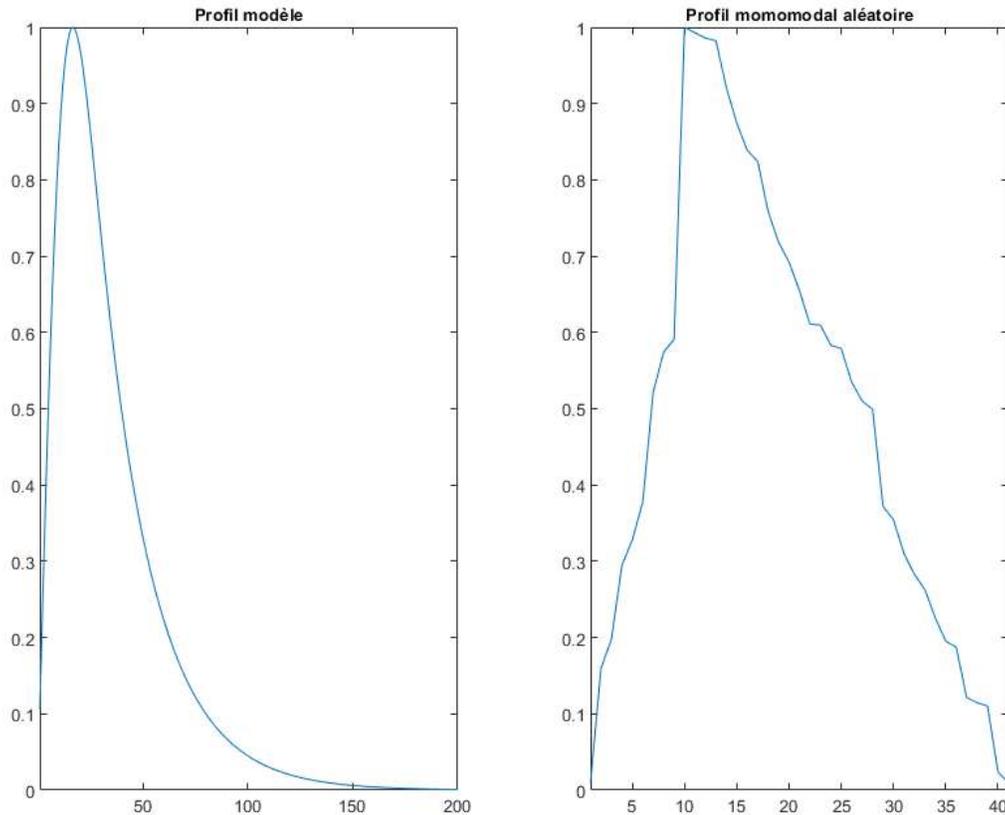


FIGURE 4.2 – Exemples de profils temporels.

Le premier signal  $s_1(t)$  correspond au produit de convolution de  $D_1(t)$  avec le profil modèle  $p_m(t)$  défini par le produit de convolution des fonctions  $g$  et  $d$  :

$$s_1(t) = (D_1 * g * d)(t) \quad (4.2)$$



**FIGURE 4.3 – Exemple d'un profil modèle.**

Un exemple de profil modèle est illustré par le tracé à gauche de la Figure 4.3 et le signal  $s_1(t)$  associé est présenté par le tracé bleu en bas à gauche (Cf image c) de la Figure 4.2. Les deux signaux  $s_2(t)$  et  $s_3(t)$  de la Figure 4.2 (Cf image a, b) sont respectivement le produit de convolution des distributions  $D_2(t)$  et  $D_3(t)$  avec des profils monomodaux aléatoires dont un exemple est présenté par le tracé à droite de la Figure 4.3. Chaque profil monomodal aléatoire  $p_a(t)$  est généré à partir de valeurs aléatoires triées par ordre croissant pour le front montant, le front descendant correspondant à des valeurs aléatoires triées par ordre décroissant. Le signal  $s_2(t)$  est caractérisé par de fortes amplitudes et par un support étroit afin de simuler des variations aléatoires rapides à forte dynamique. Quant au signal  $s_3(t)$ , ses amplitudes peuvent être 3 à 5 fois plus faibles que celles du signal  $s_2(t)$  et son support peut être 2 à 4 fois plus important afin de simuler une "ligne de base" à évolution lente et de faible dynamique. Le tracé en bas à droite de la Figure 4.2 (Cf image d) montre un exemple de profil temporel de concentration résultant de la somme de la composante semi-déterministe  $s_1(t)$  et des deux composantes aléatoires  $s_2(t)$  et  $s_3(t)$ . Les fonctions "front descendant"  $f_d$  et "front montant"  $f_m$  permettent de simuler la disparition ou l'apparition d'un composant au cours du temps en multipliant l'une ou l'autre par le profil temporel de concentration du composant. Ces deux fonctions sont basées sur la fonction test  $\varphi(t)$  de L. Schwartz dont l'expression est :

$$\varphi : t \rightarrow \begin{cases} e^{-\frac{1}{1-t^2}} & \text{si } |t| < 1 \\ 0 & \text{sinon} \end{cases} \quad (4.3)$$

La fonction  $f_d(t, t_1, t_2)$  correspondant à un front descendant qui commence à l'instant  $t_1$  et qui

se termine à l'instant  $t_2$  est définie par :

$$f_d(t, t_1, t_2) =: \begin{cases} \varphi\left(-\frac{t-t_1}{t_2-t_1}\right) & \text{si } t > t_1 \\ 1 & \text{sinon} \end{cases} \quad (4.4)$$

De la même façon, on définit la fonction  $f_m(t, t_1, t_2)$  pour un front montant sur l'intervalle temporel  $[t_1, t_2]$  :

$$f_m(t, t_1, t_2) =: \begin{cases} \varphi\left(-\frac{t-t_2}{t_2-t_1}\right) & \text{si } t < t_2 \\ 1 & \text{sinon} \end{cases} \quad (4.5)$$

Pour obtenir la disparition d'un composant entre les instants  $t_1$  et  $t_2$  suivie de sa réapparition entre les instants  $t_3$  et  $t_4$ , il suffit de multiplier son profil de concentration par la somme  $f_d(t, t_1, t_2) + f_m(t, t_3, t_4)$ . La Figure 4.4 illustre ces trois situations.

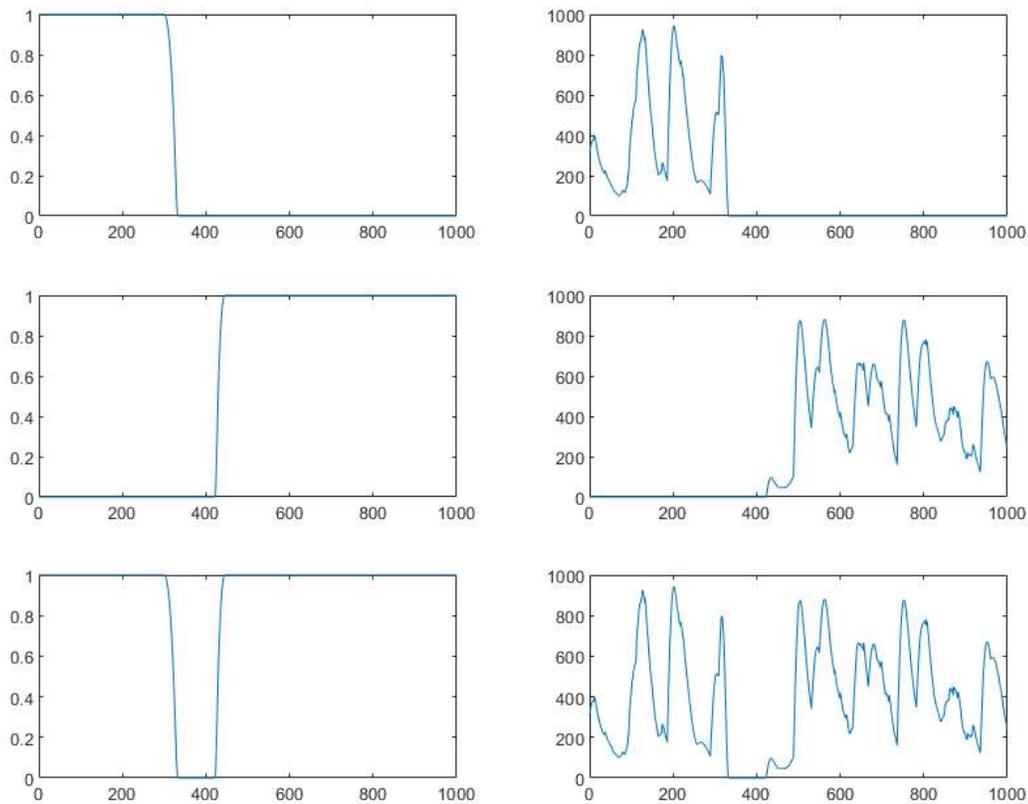


FIGURE 4.4 – Exemple d'apparition et de disparition.

## Séries

Trois séries de MEEFs sont générées en suivant les modèles décrits dans les paragraphes précédents. Dans la série 1, les  $\hat{R}$  composants fluorescents sont présents dès la première MEEF jusqu'à la dernière. Dans la série 2 (Figure 4.5), seul un composant disparaît et réapparaît (scénario à une disparition).

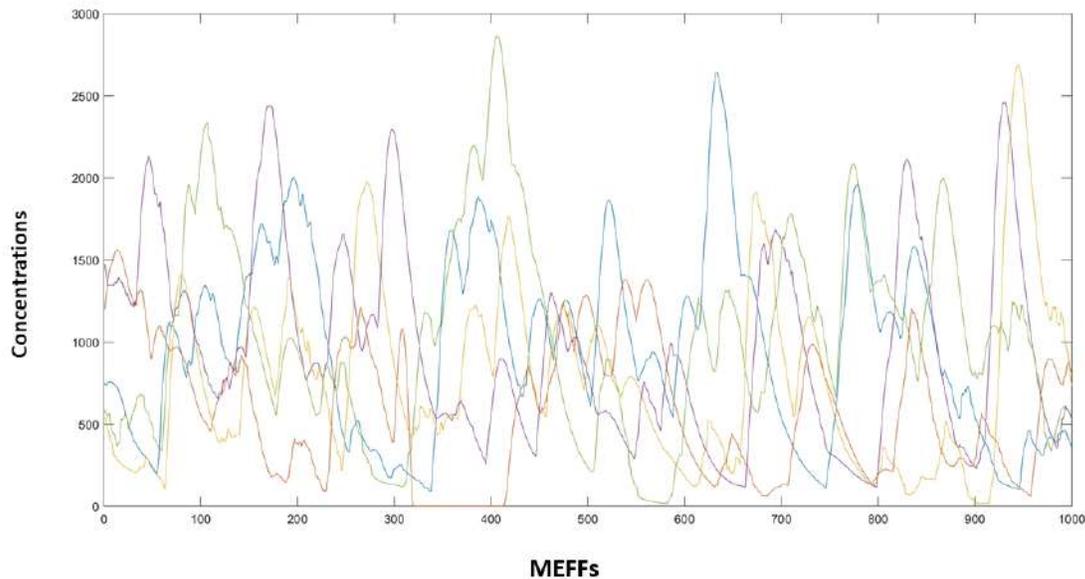


FIGURE 4.5 – Profil concentration série 2 (disparition d'un composant entre les MEEFs 315-415 )

La série 3 est particulière puisqu'elle ne comporte qu'une phase d'apparition : les  $(\tilde{R} - 1)$  premiers composants sont présents dans les  $K$  MEEFs alors que le  $\tilde{R}$ -ième composant apparaît à partir d'un certain instant. Dans l'ordre chronologique, les premières MEEFs sont donc de rang  $\tilde{R} - 1$  et les dernières sont de rang  $\tilde{R}$ . Dans cette série, il n'y a aucune phase de disparition.

#### 4.1.2 Résultats sur les données synthétiques de fluorescence

Après génération des séries, nous testons les différentes approches évoquées dans le chapitre précédent. Durant cette expérimentation, le tenseur de chaque série a pour dimensions  $41 \times 71 \times 1000$  et son rang  $\tilde{R}$  est inconnu a priori. De plus, ce tenseur n'est ni altéré par des pics de diffusion ni par du bruit.

Les 1000 MEEFs sont décomposées en plusieurs sous-tenseurs disjoints de  $41 \times 71 \times 10$  comme si, à chaque nouvel intervalle de temps, dix nouvelles MEEFs étaient acquises sans chevauchement de données. L'objectif de cette expérimentation est de simuler une décomposition en ligne c'est-à-dire de suivre périodiquement les concentrations relatives des différents fluorophores au cours du temps. Pour cela chaque série est traitée de la façon suivante :

- Le premier sous-tenseur est extrait du tenseur de départ de la dite série.
- Une première décomposition grâce à l'algorithme SNNCPD est réalisée à partir du sous-tenseur avec un rang surestimé  $R = 6$ . On suppose que  $R > \tilde{R}$ ,  $\tilde{R}$  étant le rang exact du tenseur. Dans un premier temps, nous faisons une décomposition. Le but de cette première décomposition est de fabriquer notre dictionnaire de départ à l'instant  $t = 0$  (phase d'initialisation).
- Une fois que notre dictionnaire est établi, les algorithmes OSNDTL, OSNNCPD1, OSNNCPD2 et ROSNNCPD sont utilisés en fonction des hypothèses à savoir disparition et apparition sur le reste des sous-tenseurs. En d'autres termes, la série 1 et la série 2 ayant pour hypothèse "la disparition de composant", l'algorithme OSNDTL est utilisé pour ce cas de figure. Tandis

que la série 3 présente le cas de figure "disparition-apparition". Pour cela, les algorithmes utilisés sont OSNNCPD1, OSNNCPD2 et ROSNNCPD.

Nous présentons des figures d'erreurs comme, (i) l'erreur quadratique normalisée sur les différentes matrices facteurs donnée par :

$$E_A = \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}, E_B = \frac{\|\mathbf{B} - \widehat{\mathbf{B}}\|_F}{\|\mathbf{B}\|_F} \text{ et } E_C = \frac{\|\mathbf{C} - \widehat{\mathbf{C}}\|_F}{\|\mathbf{C}\|_F} \quad (4.6)$$

(ii) l'erreur de reconstruction du tenseur définie par :

$$E_{\mathcal{T}} = \frac{\|\mathcal{T} - \widehat{\mathcal{T}}\|_F}{\|\mathcal{T}\|_F} \quad (4.7)$$

Chaque figure est composée de deux graphiques. Le premier correspond à l'erreur quadratique normalisée sur les matrices facteurs et le deuxième correspond à l'erreur de reconstruction sur les 1000 MEEFs. On peut considérer que les résultats sont bons, lorsque les points sont proches de l'origine. Il est important de rappeler que les tests ont été faits à l'aveugle c'est-à-dire que l'on ne connaît ni le rang  $\tilde{R}$ , ni le moment de la disparition ou de l'apparition d'un composant dans les 1000 MEEFs.

## Décomposition de la Série 1

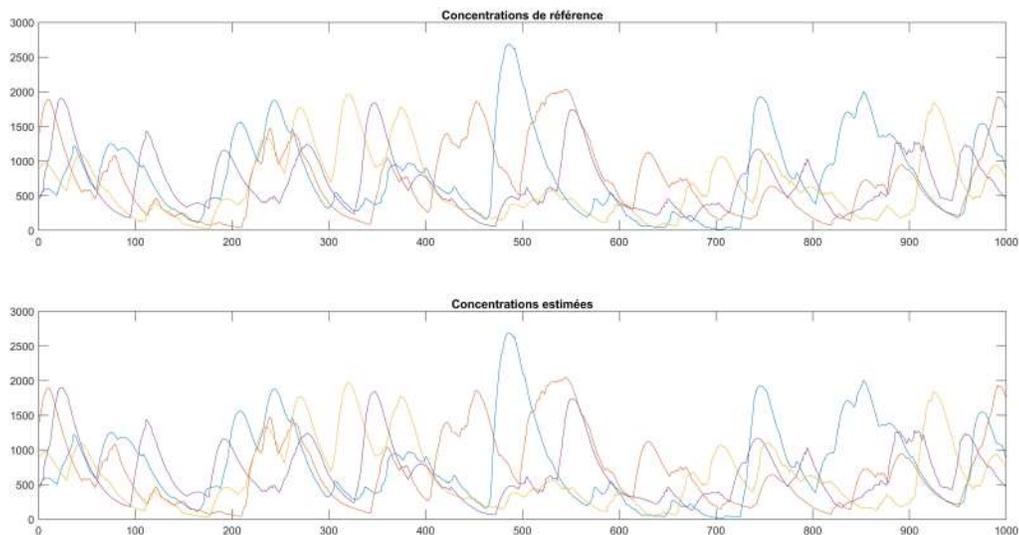
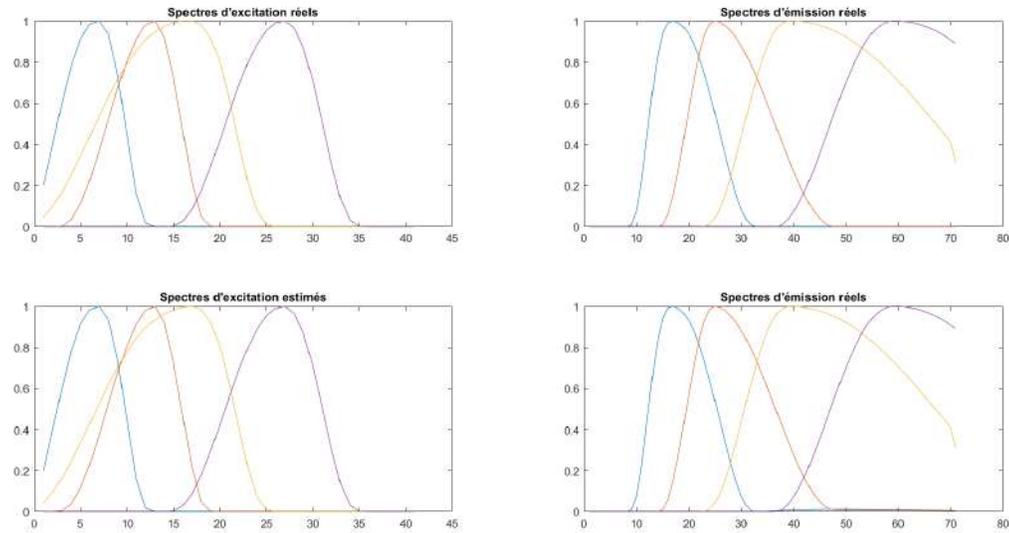
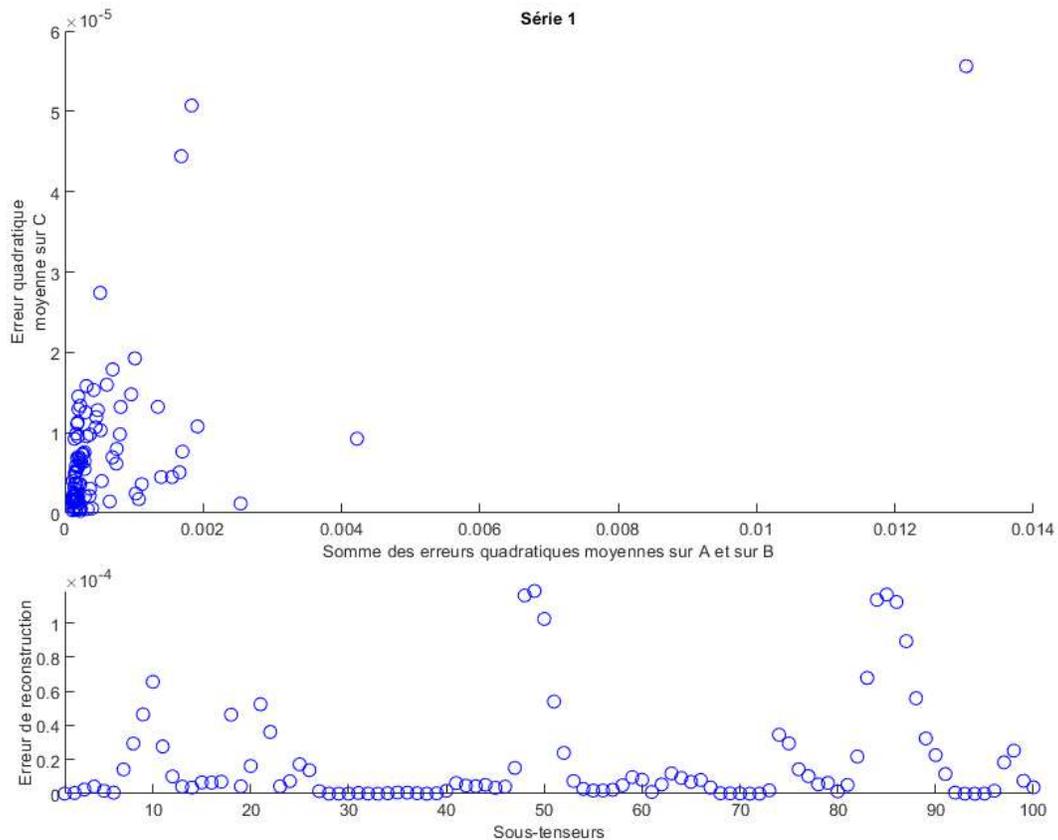


FIGURE 4.6 – Concentrations de référence et concentrations estimées de la série 1



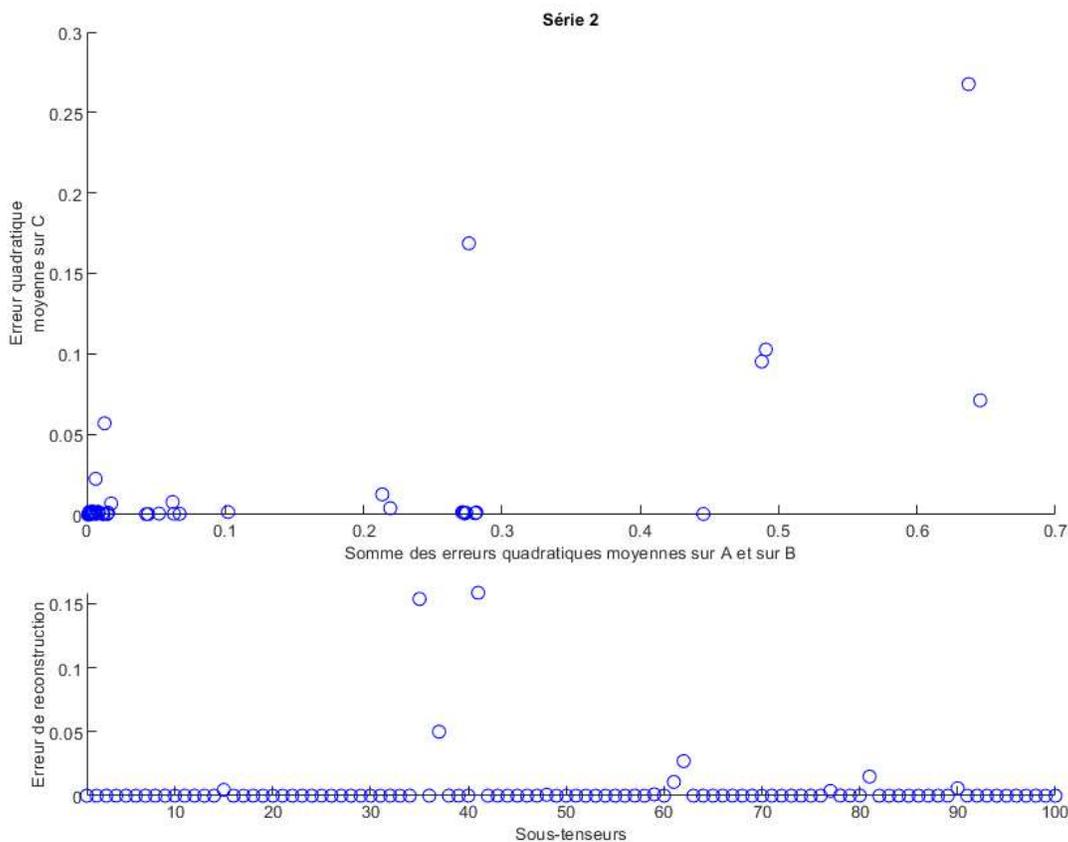
**FIGURE 4.7 – Spectres d’excitation et d’émission de référence et spectres d’excitation et d’émission estimés de la série 1**



**FIGURE 4.8 – Erreur quadratique moyenne sur les matrices facteurs et sur l’ensemble des MEEFs de la série 1 : variations des concentrations sans disparition de composant**

L'algorithme utilisé pour cette série est l'algorithme OSNDTL. Cet algorithme convient pour cette série car il n'y a pas d'apparition. On peut voir que l'algorithme arrive à bien estimer les concentrations sur les 100 blocs. On peut le constater sur la Figure 4.6 et aussi sur le Figure 4.7 qui montre les spectres d'excitations et d'émission réels et les spectres d'émission et d'émission estimées. On peut difficilement faire la différence entre les deux concentrations à savoir celle attendue et celle estimée par l'algorithme. Cela sous-entend que l'algorithme converge sur l'ensemble des 100 sous-tenseurs. En effet une bonne estimation de la concentration, signifie un bon suivi des composants en temps réel. Aussi, on peut voir sur la Figure 4.8 une bonne estimation de l'erreur de reconstruction.

## Décomposition de la Série 2

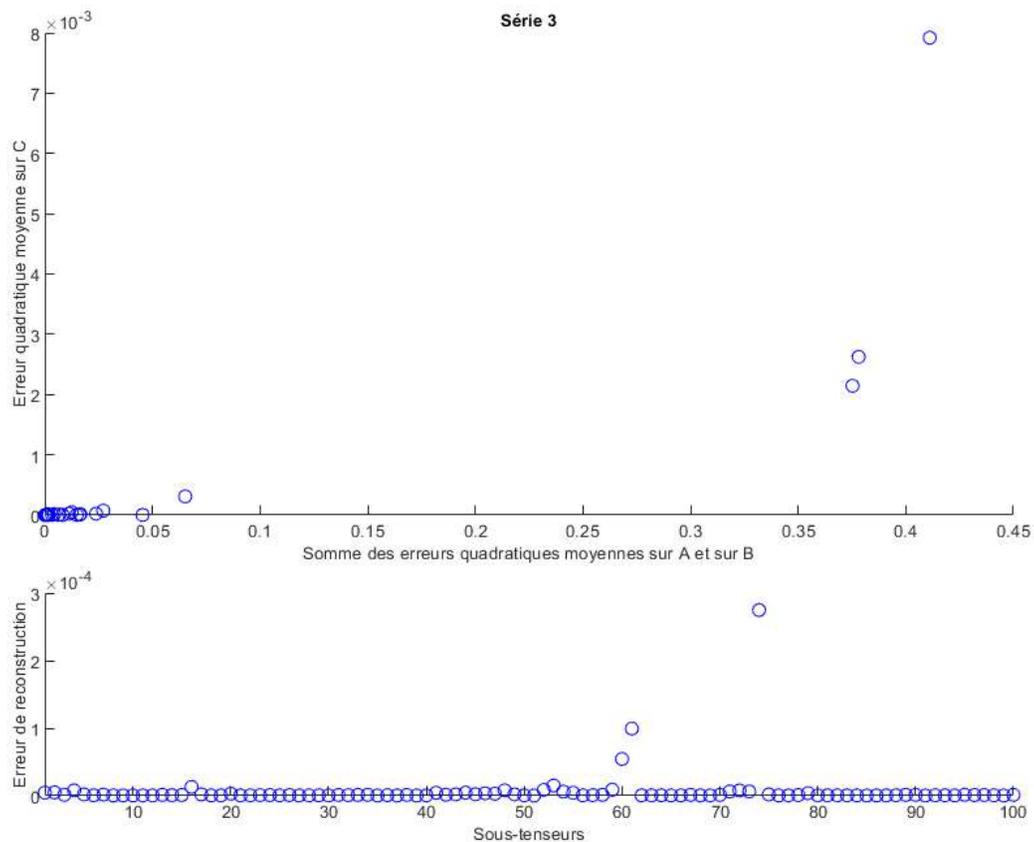


**FIGURE 4.9 – Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 2 : variations de concentrations avec une disparition d'un composant**

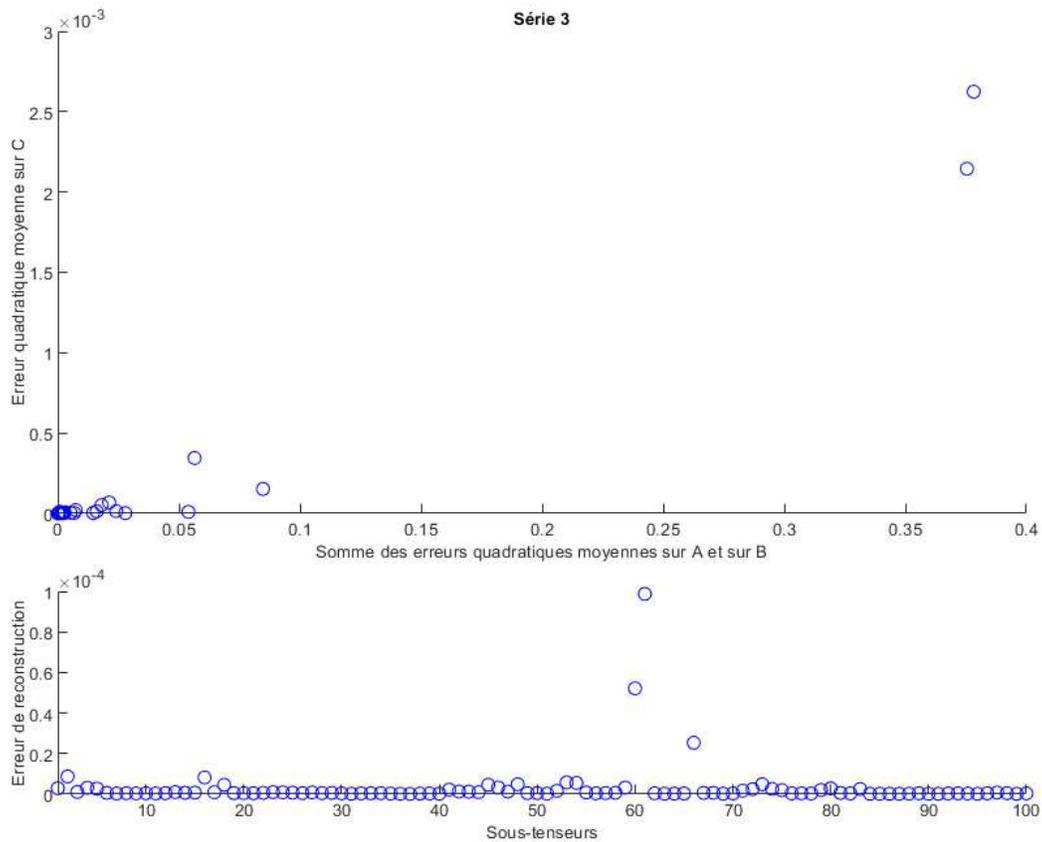
L'algorithme utilisé pour cette série est l'algorithme OSNTDL. L'erreur sur la matrice **A** reste faible, ce qui implique que l'algorithme gère la disparition et la réapparition du composant. L'algorithme arrive aussi à bien converger comme nous le montre la Figure 4.9.

## Décomposition de la Série 3

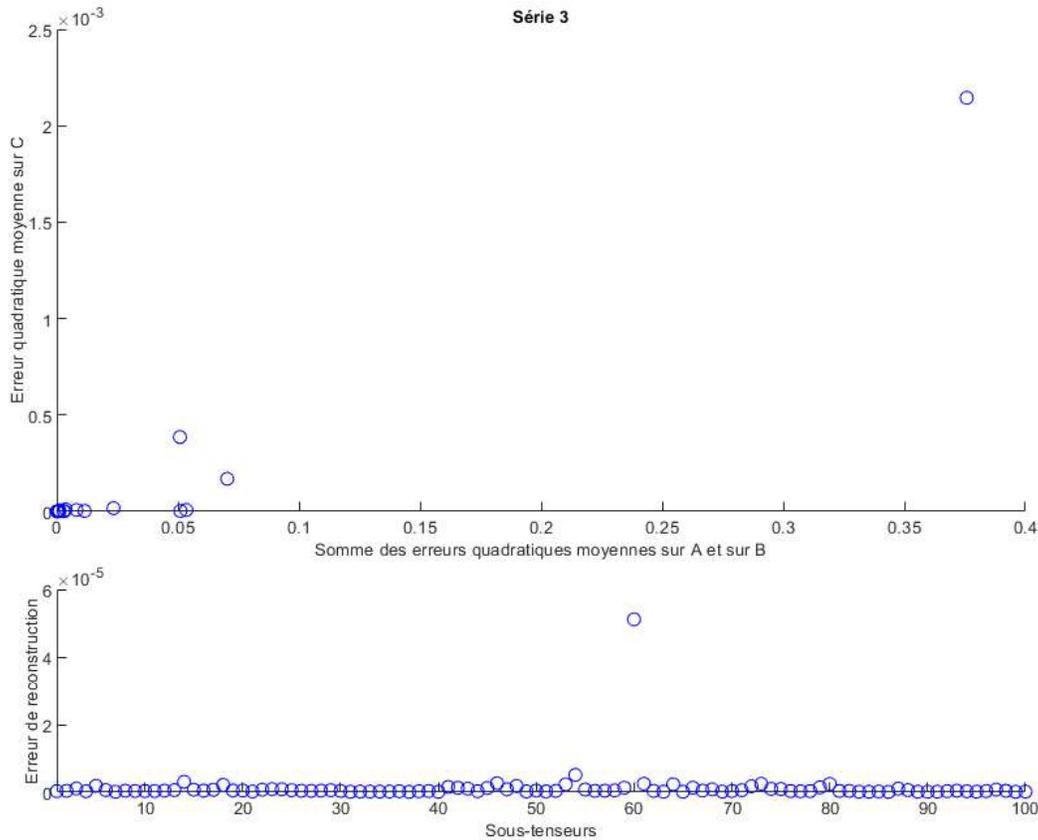
La série 3 est l'hypothèse qui est la plus difficile parce qu'il y a apparition d'un nouveau fluorophore au cours du temps c'est-à-dire une augmentation du rang. La série 3 est traitée par les algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD avec un rang  $R$  choisi supérieur à  $\tilde{R}$ . On constate une faible erreur de reconstruction du tenseur sur les 100 blocs.



**FIGURE 4.10** – Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 3 obtenue avec l'algorithme OSNNCP1 : cas apparition



**FIGURE 4.11 – Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 3 obtenue avec l'algorithme OSNNCP2 : cas apparition**



**FIGURE 4.12 – Erreur quadratique moyenne sur les matrices facteurs et sur l'ensemble des MEEFs de la série 3 obtenue avec l'algorithme ROSNNCPD : cas apparition**

Les Figures 4.10, 4.11 et 4.12 nous montrent l'erreur quadratique moyenne sur les matrices facteurs des algorithmes OSNNCP1, OSNNCP2 et ROSNNCPD. Sur la Figure 4.12, seul un bloc est associé à une grande valeur d'erreur, ce qui signifie que l'algorithme n'a pas réussi à trouver le bon rang sur ce bloc. Par contre, on voit quelques valeurs élevées sur le premier tracé de la Figure 4.10. Cela s'explique par un mauvais choix des paramètres pris au départ. Étant donné que c'est une simulation à l'aveugle, nous ne modifierons pas les hyperparamètres pour éviter une attache aux données. Mais globalement, les résultats sont bons. Comme indication d'amélioration des résultats, nous pouvons augmenter le coefficient de pénalité. Cependant sur le deuxième tracé de de la Figure 4.10, on constate que l'algorithme a une faible erreur de reconstruction de chaque sous-tenseur.

Ces résultats sur les données synthétiques sont intéressants et nous permettent de calibrer nos hyperparamètres (tableau 4.1). Il faut rappeler que le pas d'apprentissage  $\gamma$  change de valeur au cours du processus d'optimisation. Afin de choisir un bon hyperparamètre, il est conseillé d'abord de choisir un pas d'apprentissage en fonction de la taille du tenseur. Par exemple avec un tenseur ou un sous-tenseur de petite taille, il est conseillé de prendre  $\gamma \in [0.1; 1]$  d'après les simulations réalisées. Ensuite,  $\beta_1$  et  $\beta_2$  sont par défaut mis à 0.9 mais pour éviter des minima locaux, il est conseillé de diminuer leur valeur d'après les simulations réalisées. Cependant, l'inconvénient d'une valeur plus faible est que le nombre d'itération augmente.

Après une simulation sur les données synthétiques de fluorescence, il faut mettre en place une

Algorithmes	$\beta_1$	$\beta_2$	$\gamma$	$\alpha$	$\epsilon$
SNNCPD	[0.1;0.9]	[0.1;0.99]	$10^{-1}$	[0.1;1]	$10^{-5}$
OSNDTL	[0.1;0.9]	[0.1;0.99]	$10^{-3}$	[0.5;1]	$10^{-5}$
OSNNCPD1	[0.1;0.9]	[0.1;0.99]	$10^{-2}$	[0.1;1]	$10^{-5}$
OSNNCPD2	[0.1;0.9]	[0.1;0.99]	$10^{-6}$	[0.5;2]	$10^{-5}$
ROSNCPD	[0.1;0.9]	[0.1;0.99]	$10^{-1}$	[0.1;0.5]	$10^{-5}$

**TABLEAU 4.1 – Les différents hyperparamètres utilisés par chaque algorithme.**

expérimentation sur des données réelles de fluorescence. Les algorithmes ont été testés dans les mêmes conditions que nous pouvons rencontrer dans un milieu naturel. Nous présentons différentes hypothèses comme la disparition ou l'apparition ou encore les deux. Chaque algorithme est adapté à une des hypothèses décrites. OSNDTL étant un algorithme qui ne traite que l'hypothèse de la disparition n'est pas forcément compatible pour une surveillance des composants *in situ* tandis que OSNNCPD1, OSNNCPD2 et ROSNNCPD peuvent prendre en compte toutes les hypothèses.

## 4.2 Acquisition des données réelles contrôlées de fluorescences en ligne

Il est assez difficile de simuler des données de fluorescence bruitées réalistes. Plusieurs algorithmes CPD qui fonctionnent bien sur des données simulées bruitées échouent dans des conditions réelles ou semi-réelles [112]. Nous parlons de conditions réelles contrôlées lorsque le tenseur de données est construit à partir de mesures de fluorescence réelles obtenues à partir de mélanges connus fait en laboratoire. C'est le cas du dispositif expérimental décrit dans cette section et qui permet non seulement de valider les approches proposées mais aussi de les comparer aux autres algorithmes de la littérature. Dans cette partie, l'algorithme 7 n'est pas inclus dans les comparaisons car l'expérimentation consiste à simuler l'apparition, la disparition et la réapparition de composants alors que cet algorithme prend en compte uniquement la disparition et la réapparition de composants.

### 4.2.1 Matériel et dispositif expérimental



FIGURE 4.13 – Photo du Dispositif expérimental pour l'acquisition de données de fluorescence en ligne.

Nous avons réalisé en laboratoire des séries d'injections de 4 solutions de fluorophore de concentrations connues (Fluorescéine (F), Tryptophane (TRP), 5-Sulfate-8-HydroxyQuinine (5S8HQ) et Rhodamine (RHO) à  $1 \text{ g.L}^{-1}$ ). Au cours du temps d'expérience à intervalles de temps réguliers les injections ont été faites afin d'obtenir l'apparition et la disparition de certains fluorophores au cours du temps. Ces fluorophores ont été choisis en raison de leurs caractéristiques de chevauchement des pics et de leur rendement quantique de fluorescence élevé. Il faut aussi noter que le Tryptophane peut être un indicateur de la présence d'une activité biologique [2].



**FIGURE 4.14 – Raspberry model pi 4.**

Les fluorophores sont injectés dans un réservoir d'eau ( $V=26,8 L$ ) à renouvellement constant ( $D=114 mL.min^{-1}$ ). Quatre pompes correspondant aux 4 fluorophores permettent de contrôler les injections suivant un scénario pré-établi. Les injections (volume et temporisation) ont été contrôlées par un Raspberry Pi modèle 4 (Figure 4.14) comme le montre la Figure 4.15. Au-dessus du réservoir, un mélangeur a été utilisé pour agiter tout le volume. Le mélange est acheminé de manière continue vers le spectrofluorimètre équipé d'une cellule à circulation. L'acquisition des MEEFs est ensuite faite à intervalles de temps réguliers.

Les MEEFs sont enregistrées avec un spectrofluorimètre *Hitachi F7000* (Figure 4.16) doté d'une cellule à circulation en quartz de  $0,5 cm$  de chemin optique dans laquelle le mélange circule à l'aide d'une pompe au débit de  $1 mL.min^{-1}$ . La tension du tube PM est de  $700 V$  et la vitesse de balayage est de  $30\ 000 nm.min^{-1}$ . Les MEEFs sont mesurées dans la gamme de longueurs d'onde d'excitation de  $220 nm$  à  $650 nm$  (fente  $10 nm$ ) et dans la gamme de longueurs d'onde d'émission de  $220 nm$  à  $700 nm$  (fente  $10 nm$ ). Un pas de  $10 nm$  est utilisé à la fois en excitation et en émission. La taille des MEEFs est donc de  $(49, 44)$ . Le temps d'acquisition d'une MEEF dans ces conditions est inférieur à  $4 min$ .

## 4.2.2 Calibration

Afin de relier le signal de fluorescence à la concentration des fluorophores, il est procédé à une calibration. L'étape de calibration consiste à calculer la CPD de tenseurs contenant plusieurs MEEFs d'un fluorophore unique à différentes concentrations connues. L'objectif est triple :

- obtenir les spectres d'excitation et d'émission de référence de chaque fluorophore ;
- établir la relation linéaire entre les concentrations réelles et la réponse obtenue ;
- détecter d'éventuelles impuretés et un possible effet d'écran interne.

Pour cela, de petits volumes  $v$  du fluorophore sont injecté dans l'eau, le tout est dilué et acheminé vers le spectrofluorimètre pour l'acquisition des MEEFs. Le processus est répété plusieurs fois pour

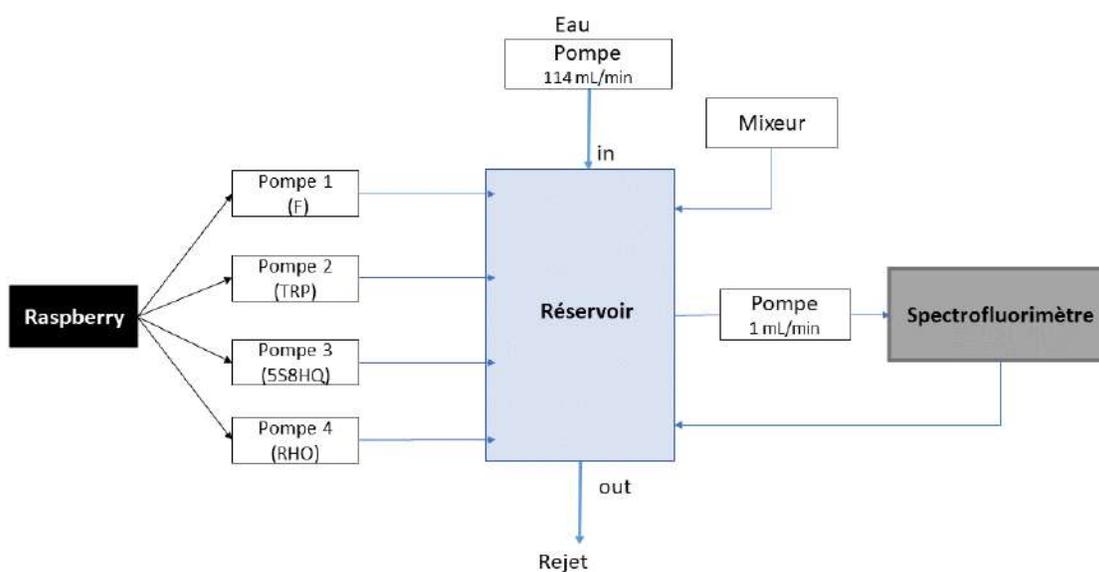


FIGURE 4.15 – Dispositif expérimental pour l'acquisition de données de fluorescence en ligne.



FIGURE 4.16 – Le spectrofluorimètre Hitachi F7000.

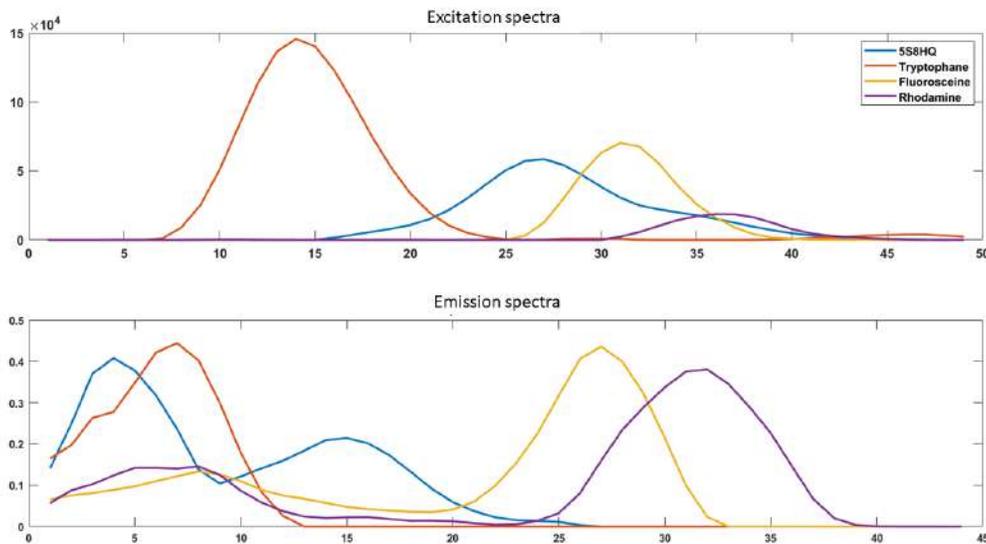


FIGURE 4.17 – Spectres d’excitation et d’émission des fluorophores estimés par l’algorithme SNNCPD.

obtenir la plage de concentration de chaque composant. On doit alors obtenir un tenseur de rang  $\tilde{R} = 1$  correspondant au fluorophore injecté. Le même protocole est répété pour chaque fluorophore. Il est important de noter qu’après chaque acquisition de tenseur, le système est totalement nettoyé afin d’éviter le mélange de fluorophores. De plus, on s’assure que le  $pH$  de l’eau ne subit aucune variation au cours de l’acquisition car ce paramètre peut influencer grandement sur le rendement quantique de fluorescence. Dans [167], il est montré en particulier l’effet du  $pH$  sur la réponse du TRP. Aussi dans [168], il est constaté cet effet sur certains composants comme le seminaphthofluoresceine (SNAFL) ou le seminaphthorhodafluors (SNARF). Nous obtenons finalement 4 tenseurs de fluorophores “purs”.

Le traitement des données se fait sous l’environnement Matlab avec le programme progMEEF développé par le MIO<sup>1</sup>. Les diffusions Raman et Rayleigh sont supprimées numériquement de chaque MEEF, selon la méthode proposée dans [126] combinée à un filtre linéaire pour éliminer les distorsions créées par la suppression. La CPD de chaque tenseur est d’abord calculée à l’aide de l’algorithme ALS non négatif [52] pour différentes valeurs de rang de 1 à 5 puis le CORE CONSistency DIAgnostic [52] est effectué pour chaque valeur de rang. Ensuite, la CPD est calculée avec notre algorithme SNNCPD pour  $R = 5$ . Nous recensons les spectres d’excitation et d’émission de chaque fluorophore (Figure 4.17) et nous les sauvegardons pour qu’ils nous servent de référence au cours de la phase d’évaluation des algorithmes.

### 4.2.3 Bruit de fond

Pour chacun des 4 tenseurs, le CORCONDIA et SNNCPD indiquent tous les 2 un rang physique de  $\tilde{R} = 2$  au lieu du  $\tilde{R} = 1$ .

Par ailleurs, les spectres d’excitation et d’émission du composant supplémentaire ne varient pas d’un tenseur à l’autre. De plus, sa contribution est assez élevée et constante. Enfin, les spectres d’excitation, les spectres d’émission et les profils de concentration de chaque fluorophore ont été

1. <http://woms18.univ-tln.fr/progmeeef/>

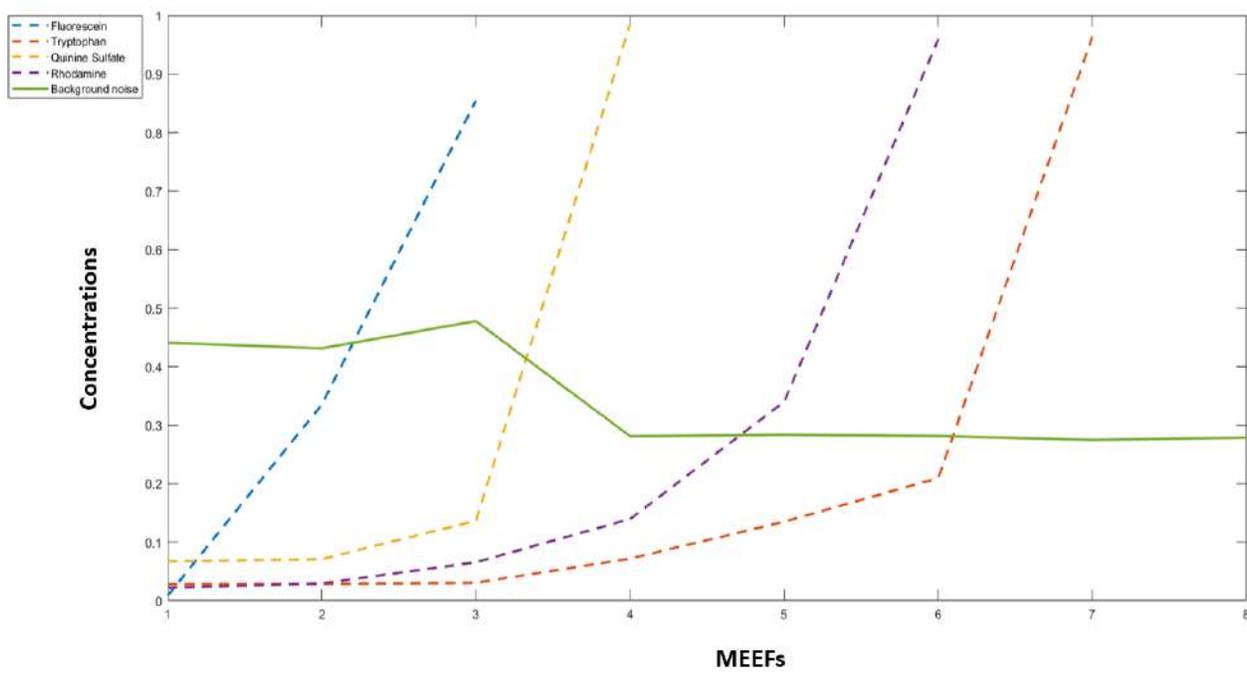


FIGURE 4.18 – Ligne verte : Profil de contribution du bruit de fond. En pointillés : les 4 fluorophores décrits ci-dessus. Tous les spectres sont normalisés.

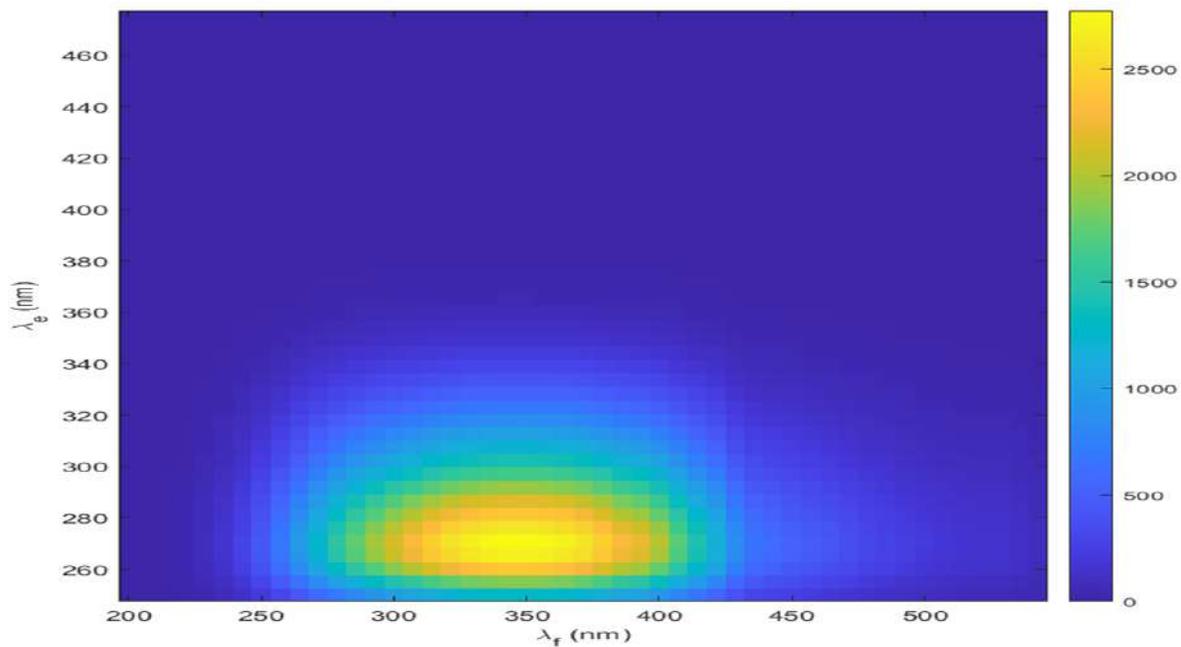


FIGURE 4.19 – Image du bruit de fond obtenue après CPD.

correctement estimés par la CPD. Aucun effet d'écran interne n'a été observé. En conséquence, le composant supplémentaire est considéré comme un cinquième fluorophore présent dans toutes les MEEFs mesurées. Il est référencé par la suite comme le bruit de fond. Sa MEEF reconstruite après la CPD est représentée par la Figure 4.19.

#### 4.2.4 Modèle théorique des concentrations

Le calcul des concentrations théoriques nous permet d'avoir les profils de concentration théoriques de référence afin de comparer avec les profils de concentration obtenus après CPD.

Notons  $V_0$  le volume du réservoir et  $C$  la concentration en fluorophore. Le nombre de moles  $N$  présentes dans le volume constant  $V_0$  du réservoir est donc égal à  $N = CV_0$ . Pendant un intervalle de temps élémentaire  $dt$ , un volume élémentaire  $dV$  d'eau et de fluorophores a été remplacé par le même volume  $dV$  d'eau. La variation correspondante du nombre de moles présentes dans le nouveau mélange est donc donnée par  $dN = -CdV$ . Le volume  $V_0$  restant constant on a aussi  $dN = V_0dC$ . En combinant les deux relations précédentes, on obtient donc :

$$dC = -\frac{1}{V_0}CdV, \quad (4.8)$$

Le débit constant  $D_0$  de l'eau rejetée dans le réservoir est défini par :

$$D_0 = \frac{dV}{dt}, \quad (4.9)$$

Ce qui donne :

$$dC = -\alpha_0 C dt, \quad (4.10)$$

où  $\alpha_0 = \frac{D_0}{V_0}$ . Notons maintenant  $C_0$ , la concentration en fluorophore à l'injection. On peut ainsi exprimer la variation de la concentration avec le temps :

$$C(t) = C_0 H(t) e^{-\alpha_0 t}, \quad (4.11)$$

où  $H(t)$  est la fonction échelon de Heaviside. Il faut alors tenir compte du fait que l'acquisition d'une MEEF n'est pas instantanée. Notons  $\Delta t$  le temps d'acquisition et  $t_k$  l'instant de début d'acquisition de la MEEF  $k$ . On estime donc finalement la concentration correspondante  $C_k$  en faisant la moyenne de la variation de concentration de  $t_k$  à  $t_k + \Delta t$  :

$$C_k = \frac{1}{\Delta t} \int_{t_k}^{t_k + \Delta t} C_0 e^{-\alpha_0 t} dt = \frac{C_0}{\alpha_0 \Delta t} (1 - e^{-\alpha_0 \Delta t}) (e^{-\alpha_0 t_k}) \quad (4.12)$$

#### 4.2.5 Métriques de comparaisons

Trois critères de performance sont utilisés pour l'évaluation des algorithmes, à savoir :

- l'estimation du rang physique,
- la moyenne des erreurs normalisées sur les matrices facteurs,
- l'erreur de reconstruction relative des sous-tenseurs.

Tout d'abord, concernant l'estimation du rang physique, les résultats d'algorithmes utilisant les méthodes de détermination du rang avant décomposition comme l'algorithme SeekAndDestroy ou NLS sont retenus. Pour les algorithmes imposant une contrainte de parcimonie sur une ou toutes les matrices factorielles, nous comptons les colonnes nulles ou proches de zéros (on considère une colonne nulle si tous ses coefficients sont inférieurs à  $10^{-5}$ ).

Ensuite, après avoir supprimé les colonnes nulles de  $\widehat{\mathbf{A}}$ ,  $\widehat{\mathbf{B}}$  et les colonnes correspondantes de  $\widehat{\mathbf{C}}$  pour les algorithmes imposant la parcimonie contrainte, les vrais facteurs sont comparés aux matrices de facteurs estimées grâce à la moyenne des erreurs quadratiques normalisées ou Mean Square Error (MSE) que nous rappelons :

$$E_A = \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}, E_B = \frac{\|\mathbf{B} - \widehat{\mathbf{B}}\|_F}{\|\mathbf{B}\|_F} \text{ et } E_C = \frac{\|\mathbf{C} - \widehat{\mathbf{C}}\|_F}{\|\mathbf{C}\|_F} \quad (4.13)$$

Ici, les matrices de référence  $\mathbf{A}$ ,  $\mathbf{B}$  ont été obtenues lors de l'étape de calibration et la matrice  $\mathbf{C}$  à travers l'équation (4.12). Le deuxième critère est alors :

$$E_M = (E_A + E_B + E_C)/3 \quad (4.14)$$

Les extra-facteurs estimés sont considérés comme des composants dupliqués. Pour détecter des composants dupliqués, il suffit de calculer la corrélation entre chaque couple de spectres. Parmi les composants estimés considérés comme dupliqués, on retient les composants qui ont une forte corrélation avec les composants réels et avec une concentration plus élevée.

Troisièmement, la qualité de reconstruction relative utilisée dans [111, 119] et cela nous permet de juger la convergence de chaque algorithme. Elle est définie par :

$$E_{\mathcal{T}}(\%) = 100\left(1 - \frac{\|\mathcal{T} - \widehat{\mathcal{T}}\|_F}{\|\mathcal{T}\|_F}\right) \quad (4.15)$$

## 4.2.6 Mise en place des bases de données de test

### Premier jeu de test : sous-tenseurs de petite taille

Sous-tenseur $\mathcal{T}_n$	$\mathcal{T}_0$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$
$\widetilde{R}_n$	3	4	5	5
RHO	✓	✓	✓	✓
5S8HQ	✓	✓	✓	✓
TRP	×	✓	✓	✓
F	×	×	✓	✓
Background noise	✓	✓	✓	✓

TABLEAU 4.2 – Les différents sous-tenseurs  $\mathcal{T}_n$  avec leur rang vrai correspondant de la première base de donnée.

Pour ce premier jeu de données, 50 MEEFs de taille (49, 44) ont été enregistrées en ligne et le tenseur de fluorescence correspondant a été partitionné en 4 sous-tenseurs successifs ( $\mathcal{T}_0 \dots \mathcal{T}_3$ ) comme l'indique le tableau 4.2 avec les rangs physiques  $\widetilde{R}_n$ . Les paramètres d'acquisition des données sont les mêmes que pour l'étape d'étalonnage et la diffusion est corrigée de la même manière. Nous avons considéré deux types de découpage. Dans la première partition (sans chevauchement), les sous-tenseurs n'ont aucune MEEF en commun :  $\mathcal{T}_0$  contient 20 MEEFs et les autres sous-tenseurs ne

contiennent que 10 MEEFs. Dans la deuxième partition (recouvrement à 50%), chaque sous-tenseur contient 20 MEEFs et deux sous-tenseurs consécutifs ont 10 MEEFs en commun. Intuitivement, la première partition devrait rendre la décomposition plus difficile car les sous-tenseurs ne partagent pas de MEEFs. Cependant, elle est plus intéressante d'un point de vue pratique puisqu'elle nécessite moins de calculs car la phase en ligne traite de plus petits tenseurs. Nous avons choisi pour la décomposition  $R = 10$  afin de surestimer le rang.

## Second jeu de données : sous-tenseurs de grande taille

Un deuxième jeu de données comprenant 1000 MEEFs a été acquis de la même manière. On obtient alors un tenseur de données  $\mathcal{T}$  de taille (49, 44, 1000). Le tenseur  $\mathcal{T}$  est ensuite divisé en 10 sous-tenseurs de taille (49, 44, 100) sans chevauchement entre les sous-tenseurs. Le tableau 4.3

Sous-tenseur $\mathcal{T}_n$	$\mathcal{T}_0$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	$\mathcal{T}_7$	$\mathcal{T}_8$	$\mathcal{T}_9$
$\tilde{R}_n$	3	4	5	3	5	4	5	4	5	5
RHO	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5S8HQ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TRP	×	✓	✓	×	✓	✓	✓	✓	✓	✓
F	×	×	✓	×	✓	×	✓	×	✓	✓
Background noise	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLEAU 4.3 – Les différents sous-tenseurs  $\mathcal{T}_n$  avec leur rang vrai correspondant.

indique les fluorophores présents dans chaque sous-tenseur  $\mathcal{T}_n$  ainsi que les rangs physiques  $\tilde{R}_n$ . Rappelons que le bruit de fond n'est pas supprimé et qu'il est considéré comme un extra-fluorophore présent dans toutes les MEEFs.  $\tilde{R}_n$  est alors égal à 5 au maximum et au minimum égal à 3. Ensuite, trois découpages des données sont considérés :

- sans chevauchement de données entre deux sous-tenseurs consécutifs,
- 50% de chevauchement de données entre deux sous-tenseurs consécutifs,
- 90% de chevauchement de données entre deux sous-tenseurs consécutifs.

## 4.3 Résultats sur les données réelles contrôlées de fluorescence

### 4.3.1 Premier jeu de données

Nous comparons d'abord les valeurs des rangs physiques estimées par OSNNCPD1, OSNNCPD2 et ROSNNCPD avec celles estimées par AutoTen [54] et l'algorithme NN-CPD proposé dans [93] et les valeurs réelles. Ensuite nous montrons le deuxième critère qui est la moyenne sur l'estimation des matrices facteurs. Ici, nous ne montrons pas la qualité de reconstruction du fait de la taille des données. Nous indiquons également la valeur du terme du coefficient de pénalité ( $\alpha$ ) utilisé dans nos deux algorithmes et dans NN-CPD.

#### 4.3.1.1 Cas sans chevauchement de données

Sous-tenseur	0	1	2	3	$\alpha$
Rang physique	3	4	5	5	-
AutoTen	3	2	2	2	-
NN-CPD	3	4	5	10	0.1 - 0.5
OSNNCPD1, OSNNCPD2, ROSNNCPD	3	4	5	5	0.5

TABLEAU 4.4 – Estimation du rang dans le cas sans chevauchement.

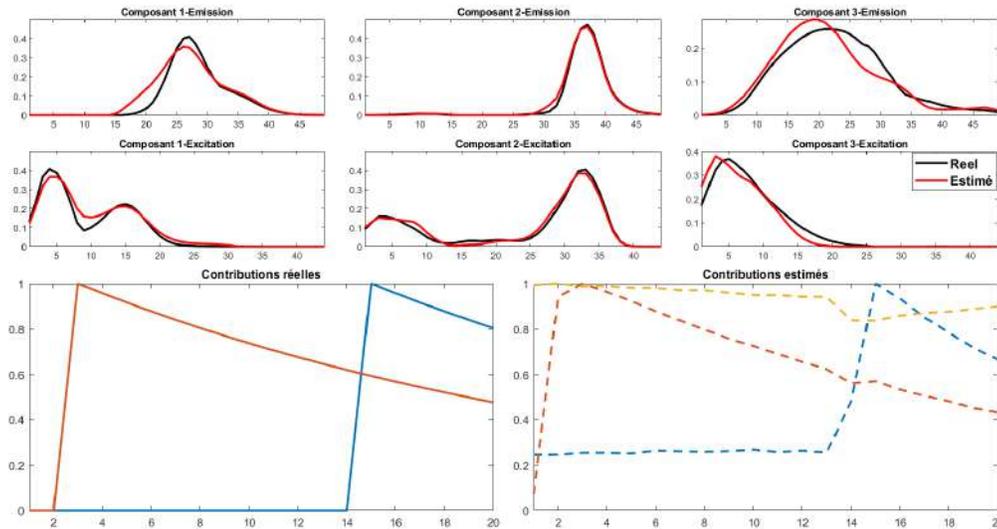
**Estimation du rang** Les résultats sont rapportés dans le tableau 4.4 pour le cas sans chevauchement. OSNNCPD 1, OSNNCPD 2 et ROSNNCPD trouvent le rang correct de chaque sous-tenseur. En revanche, AutoTen sous-estime le rang pour 3 sous-tenseurs sur 4 tandis que NN-CPD arrive à trouver le bon rang des 3 premiers sous-tenseurs seulement.

	Algorithme 8 (OSNCPD1)				Algorithme 9 (OSNCPD2)				Algorithme 10 (ROSNNCPD)			
Sous-tenseur	0	1	2	3	0	1	2	3	0	1	2	3
Moyenne $E_A$	0.17	0.28	0.35	0.35	0.17	0.18	0.21	0.33	0.17	0.25	0.33	0.32
Moyenne $E_B$	0.14	0.15	0.31	0.32	0.14	0.16	0.19	0.24	0.14	0.14	0.31	0.30
Moyenne $E_C$	0.25	0.32	0.31	0.32	0.25	0.31	0.32	0.29	0.25	0.29	0.31	0.3

TABLEAU 4.5 – Erreur moyenne sur les matrices facteurs dans le cas sans chevauchement.

**Erreur moyenne sur les matrices facteurs** Les résultats sont donnés dans le tableau 4.5. Dans le cas sans chevauchement, OSNNCPD2 surpasse systématiquement OSNNCPD1 et ROSNNCPD, en ce qui concerne l'estimation des 2 matrices spectrales A et B en particulier pour les sous-tenseurs  $\mathcal{T}_2$  et  $\mathcal{T}_3$ . Ici, il apparaît que la valeur de  $E_A$  obtenue avec OSNNCPD1 sur  $\mathcal{T}_1$  est déjà assez élevée et

ensuite l'erreur se propage aux sous-tenseurs successifs. Ceci est probablement dû à la petite taille des sous-tenseurs. Dans ce contexte, la plus grande flexibilité d'OSNNCPD2 atténue ce problème. Enfin l'erreur sur  $E_c$  obtenue avec les 3 algorithmes est comparable et relativement élevée.



**FIGURE 4.20 – Spectres d’émission et d’excitation des fluorophores en phase d’initialisation ( $\mathcal{T}_0$ ).** En haut : spectres d’émission, milieu : spectres d’excitation. Lignes rouges : spectres estimés, lignes noires : spectres réels. Tous les spectres sont normalisés. En bas : concentration de fluorophores dans la phase d’initialisation ( $\mathcal{T}_0$ ). Ligne pleine : contributions « vraies » (la contribution du bruit de fond est inconnue), ligne pointillée : les contributions estimées. Toutes les concentrations sont normalisées.

Afin de donner un sens physique à ces termes d’erreur qui peuvent sembler élevés, la Figure 4.20 présente les facteurs normalisés issus de notre approche en phase d’initialisation (sous-tenseur  $\mathcal{T}_0$ ) avec les vrais facteurs ainsi que les contributions. Nous observons un bon accord entre les facteurs réels et estimés. Cependant, il y a une compensation dans les contributions estimées. Cet écart peut être expliqué par le fait que les contributions réelles ne sont pas exactement connues puisqu’elles ont été établies théoriquement (cf section 4.2.4) à partir des caractéristiques des dispositifs d’injection et d’évacuation. De plus, ce modèle théorique ne prend pas en compte le fluorophore additionnel. Ce décalage explique les valeurs plus élevées de  $E_C$  par rapport à  $E_A$  et  $E_B$ . Au final, la contribution estimée du fluorophore supplémentaire, considéré comme un bruit de fond, est presque constante comme cela était attendu.

#### 4.3.1.2 Cas 50% chevauchement de données

**Estimation du rang** Les résultats sont rapportés dans le tableau 4.6 pour le cas avec chevauchement. AutoTen échoue pour deux sous-tenseurs. Cela s’explique par la contribution plus faible de plusieurs fluorophores dans les dernières MEEFs et la corrélation des spectres. A l’inverse, NN-CPD surestime le rang des 2 derniers sous-tenseurs. Il est probable qu’un réglage de  $\alpha$  soit nécessaire. En revanche, OSNNCPD1, OSNNCPD2 et ROSNNCPD trouvent le rang correct de chaque sous-tenseur.

<b>Sous-tenseur</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	$\alpha$
<b>Rang réel</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>5</b>	-
AutoTen	3	4	4	4	-
NN-CPD	3	4	7	10	0.5 - 1
<b>OSNNCPD1, OSNNCPD2, ROSNNPCD</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>5</b>	<b>1</b>

TABLEAU 4.6 – Estimation du rang dans le cas sans chevauchement.

<b>Sous-tenseur</b>	<b>Algorithme 8 (OSNCPD1)</b>				<b>Algorithme 9 (OSNCPD2)</b>				<b>Algorithme 10 (ROSNNPCD)</b>			
	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
Moyenne $E_A$	0.17	0.19	0.19	0.2	0.17	0.18	0.2	0.34	0.17	0.18	0.2	0.21
Moyenne $E_B$	0.14	0.15	0.15	0.16	0.14	0.16	0.19	0.24	0.14	0.15	0.17	0.16
Moyenne $E_C$	0.25	0.25	0.16	0.20	0.25	0.26	0.18	0.29	0.25	0.23	0.2	0.18

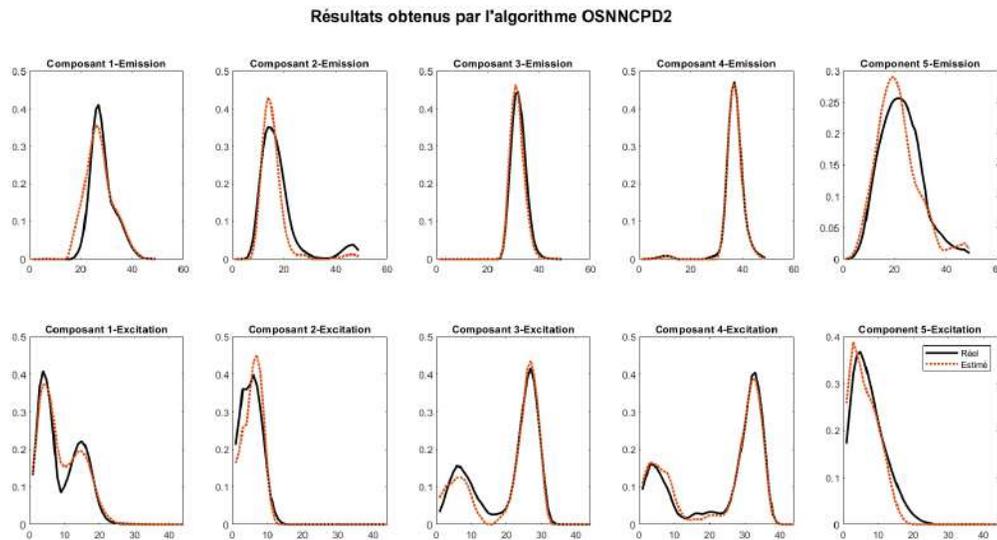
TABLEAU 4.7 – Erreur moyenne sur les matrices facteurs dans le cas de chevauchement.

**Erreur moyenne sur les matrices facteurs** Les résultats sont rapportés dans tableau 4.7 pour le cas de chevauchement à 50%. Ici, OSNCPD1 et ROSNNCPD donnent les meilleurs résultats du fait de la superposition des données entre deux sous-tenseurs consécutifs. En particulier, pour le dernier sous-tenseur, on observe également que le chevauchement a permis une meilleure estimation des profils de concentration. En ce qui concerne l'estimation de **A** et **B**, il convient de mentionner que les résultats OSNNCPD2 ne varient pas du cas avec chevauchement au cas sans chevauchement. Les hyperparamètres utilisés sont donnés dans le tableau 4.8.

<b>Algorithmes</b>	$\beta_1$	$\beta_2$	$\gamma$	$\epsilon$
SNNCPD	0.9	0.99	$10^{-1}$	$10^{-5}$
OSNNCPD1	0.9	0.99	$10^{-1}$	$10^{-5}$
OSNNCPD2	0.9	0.99	$10^{-5}$	$10^{-5}$
ROSNNPCD	0.9	0.99	$10^{-3}$	$10^{-5}$

TABLEAU 4.8 – Les différents hyperparamètres utilisés par chaque algorithme pour ce dataset.

La Figure 4.21 compare les spectres des 4 composants que nous avons injectés et le bruit de fond avec ceux estimés dans le cas du chevauchement de données obtenus avec l'algorithme OSNNCPD1. Aussi, OSNNCPD1 et ROSNNCPD donnent également le même résultat. La figure montre que l'on arrive à retrouver les bons fluorophores injectés malgré la surestimation du rang et la corrélation des spectres. Après une comparaison sur un petit jeu de données, il est important d'augmenter la quantité de données mais aussi d'utiliser d'autres algorithmes de l'état de l'art, tout en gardant le même processus d'acquisition.



**FIGURE 4.21 – Emission et excitation et spectres des fluorophores en phase d’initialisation ( $\mathcal{T}_3$ ). En haut : Spectres d’émission, en bas : Spectres d’excitation. Points rouges : spectres estimés, lignes noires : spectres réels. Tous les spectres sont normalisés.**

### 4.3.2 Second jeu de données : 1000 MEEFs

Dans cette partie, nous effectuons une comparaison plus large par rapport à la comparaison précédente. Les algorithmes proposés OSNNCPD1, OSNNCPD2 et ROSNNCPD sont comparés à SeekAndDestroy [51], Block-Coordinate Variable Metric Forward-Backward (BCVMFB) proposé dans [64], Nonlinear Least Squares update of the canonic polyadic decomposition (NLS) [78] et Robust Canonical Polyadic Decomposition (R-CPD) [102]. Les algorithmes BCVMFB et R-CPD sont également basés sur la surestimation du rang et peuvent être utilisés en ligne (même s’ils n’ont pas été prévus pour cette utilisation) en surestimant le rang au départ et en calculant la CPD pour chaque nouveau sous-tenseur. SeekAndDestroy et NLS sont des algorithmes de CPD en ligne. NLS a déjà été comparé aux algorithmes SDT et RLST dans [78] mais il ne gère pas la variation du rang. Par contre, SeekAndDestroy gère la variation du rang et il a été comparé à OnlineCP et à SamBaTen. dans [51].

En plus des métriques mentionnées plus haut, cette partie montre les temps d’exécution moyens de chaque algorithme proposé pour le traitement des sous-tenseurs. Pour chaque métrique, nous comparons les algorithmes pour un rang surestimé égal à 7, 10 ou 20.

#### 4.3.2.1 Cas sans chevauchement de données

**Estimation du rang** Le cas sans chevauchement est difficile à traiter car le vrai rang peut varier d’un sous-tenseur à l’autre. Ici, le tenseur  $\mathcal{T}$  est découpé en 10 sous-tenseurs  $\mathcal{T}_n$ . Le tableau 4.9 montre le rang estimé pour chaque sous-tenseur par chaque algorithme en comparaison avec le vrai rang  $\tilde{R}$ . Les algorithmes SeekAndDestroy et NLS permettent de déterminer le rang exact  $\tilde{R}$  pour quelques sous-tenseurs mais ont tendance à sous-estimer la valeur du rang pour la plupart des sous-tenseurs (à l’exception d’une surestimation avec SeekAndDestroy sur le sous-tenseur  $\mathcal{T}_3$ ). Inversement, R-CPD et BCVMFB surestiment le rang pour la plupart des sous-tenseurs. D’autre part, les algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD trouvent le vrai rang pour la plupart

Sous-tenseur $\mathcal{T}_n$		$\mathcal{T}_0$	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	$\mathcal{T}_7$	$\mathcal{T}_8$	$\mathcal{T}_9$
<b>Rang réel <math>\tilde{R}</math></b>		<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
SeekAndDestroy	R=7	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	4	4	4
	R=10	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	4	4	4
	R=20	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	4	4	4
NLS	R=7	<b>3</b>	<b>4</b>	4	2	4	3	4	3	4	4
	R=10	<b>3</b>	<b>4</b>	4	2	4	3	4	3	4	4
	R=20	<b>3</b>	<b>4</b>	4	2	4	3	4	3	4	4
R-CPD	R=7	<b>3</b>	<b>4</b>	6	5	6	5	6	6	6	6
	R=10	<b>3</b>	<b>4</b>	6	5	6	5	6	6	6	6
	R=20	<b>3</b>	<b>4</b>	6	5	6	5	6	6	6	6
BCVMFB	R=7	<b>3</b>	<b>4</b>	4	4	6	6	6	6	6	6
	R=10	<b>3</b>	<b>4</b>	4	4	6	6	6	6	6	6
	R=20	<b>3</b>	<b>4</b>	4	4	6	8	10	8	10	10
OSNNCPD1	R=7	<b>3</b>	<b>4</b>	4	4	<b>5</b>	<b>4</b>	4	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=20	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	4	5	6	6
OSNNCPD2	R=7	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=20	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
ROSNNCPD	R=7	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=20	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	10	<b>5</b>	8	10	10

TABLEAU 4.9 – Estimation du rang dans le cas sans chevauchement.

des sous-tenseurs, en particulier OSNNCPD2 qui trouve le rang exact  $\tilde{R}$  pour chaque sous-tenseur et quel que soit le rang CPD utilisé. ROSNNCPD ne fait pas d'erreur pour  $R=7$  et  $R=10$  et enfin OSNNCPD1 ne fait pas d'erreur que pour  $R=10$ . La surestimation de rang observée avec R-CPD et BCVMFB s'explique notamment par la faible valeur seuil que nous avons utilisée ici pour définir la colonne nulle dans les matrices factorielles estimées. Cependant, cela indique également que nos algorithmes sont moins sensibles à cette valeur et que les facteurs avec la plus petite contribution peuvent être clairement identifiés comme des facteurs supplémentaires.

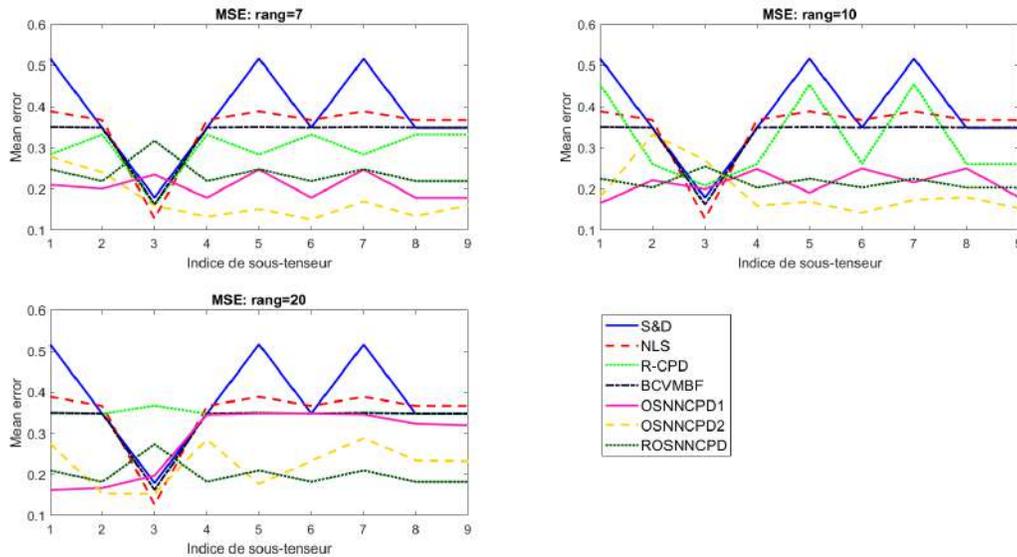


FIGURE 4.22 – Erreur moyenne ( $E_A, E_B, E_C$ ) sur les matrices facteurs dans le cas sans chevauchement.

**Erreur moyenne sur les matrices facteurs** La Figure 4.22 montre l'évaluation de l'erreur sur les matrices facteurs pour les 3 valeurs différentes de  $R$ . Pour les 3 rangs de la CPD choisis en entrée et pour la plupart des sous-tenseurs, les algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD permettent d'obtenir des erreurs moyennes plus faibles que les autres algorithmes. Ceci est une conséquence directe de la sous-estimation ou de la surestimation du rang obtenu par les autres approches.

**Estimation de la qualité de reconstruction** La Figure 4.23 montre la qualité de reconstruction relative (équation 4.15) pour les 3 différentes valeurs de  $R$ . Tous les algorithmes testés ont une qualité de reconstruction élevée, ce qui implique que tous ces algorithmes ont correctement convergé. Concernant SeekAndDestroy, NLS et BCVMFB, la qualité relative de reconstruction est stable par rapport au rang  $R$ . Ce rapport de qualité augmente avec la valeur de  $R$  pour les algorithmes R-CPD, OSNNCPD1 et OSNNCPD2 alors que le rapport diminue dans le cas de ROSNNCPD. On peut remarquer que les algorithmes OSNNCPD1 et OSNNCPD2 obtiennent la meilleure qualité de reconstruction relative quel que soit le rang  $R$  choisi en entrée.

Il faut noter qu'une bonne qualité de reconstruction ne signifie pas que l'algorithme a bien fonctionné car il peut reconstruire le bruit ou des composants différents de la référence. On considère une bonne qualité de reconstruction lorsque sa valeur est supérieure à 90. La Figure 4.22 indique que certains algorithmes de la littérature ont une erreur moyenne un peu élevée, mais une bonne

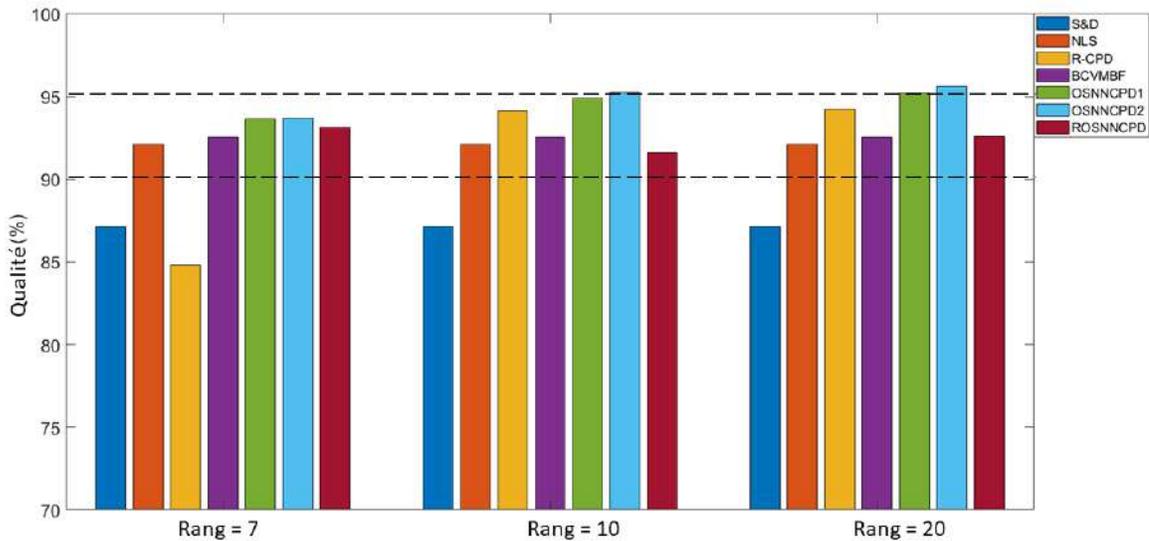


FIGURE 4.23 – Qualité de reconstruction relative pour le cas sans chevauchement.

qualité de reconstruction comme c'est le cas de NLS où R-CPD. Aussi, il faut prendre en compte l'estimation du rang qui joue un rôle important selon que l'algorithme le surestime ou le sous-estime par rapport au rang physique. C'est le cas de BCVMFB, qui garde un équilibre entre l'estimation du rang, l'erreur moyenne et ce qui conduit une qualité de reconstruction assez bonne même si l'algorithme a tendance à surestimer d'un facteur le rang dans certains sous-tenseurs. Tandis que, OSNNCPD1, OSNNCPD2 et ROSNNCPD ont de bons résultats sur les métriques de comparaison.

#### 4.3.2.2 Cas avec 50% chevauchement de données

**Estimation du rang** Dans ce cas, il y a au total 19 sous-tenseurs mais seuls 10 sous-tenseurs principaux sont montrés pour l'estimation du rang. Ces 10 sous-tenseurs principaux ( $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_3, \mathcal{T}_7, \mathcal{T}_9, \mathcal{T}_{11}, \mathcal{T}_{13}, \mathcal{T}_{15}, \mathcal{T}_{17}, \mathcal{T}_{19}$ ) sont choisis en fonction de la variation du rang pour être en accord avec les sous-tenseurs du cas de non chevauchement. Les résultats sur l'estimation du rang des différents algorithmes sont donnés dans le tableau 4.10.

Pour l'algorithme BCVMFB, le rang estimé est toujours égal au rang initial du CPD ( $R = 7, 10$  ou 20) cela est dû est une fabrication de doublons. Pour le rang initial  $R$  égal à 7, R-CPD donne un rang  $R$  surestimé de un. Pour les autres valeurs de rang, le rang estimé par cet algorithme est égal à  $R$ .

Les algorithmes R-CPD et SeekAndDestroy ne donnent respectivement que 30% et 40% de rangs estimés exacts pour les différentes valeurs de rang CPD.

Auparavant, OSNNCPD2 donnait la meilleure estimation du rang des sous-tenseurs quel que soit le rang CPD initial. Avec 50% de recouvrement, cet algorithme échoue dans 20% des cas. Cependant, les algorithmes OSNNCPD1 et ROSNNCPD donnent le rang exact des sous-tenseurs dans tous les cas.

Sous-tenseur $\mathcal{T}_n$		$\mathcal{T}_0$	$\mathcal{T}_1$	$\mathcal{T}_3$	$\mathcal{T}_7$	$\mathcal{T}_9$	$\mathcal{T}_{11}$	$\mathcal{T}_{13}$	$\mathcal{T}_{15}$	$\mathcal{T}_{17}$	$\mathcal{T}_{19}$
<b>Rang physique <math>\tilde{R}</math></b>		<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
SeekAndDestroy	R=7	3	4	4	4	5	3	3	7	7	7
	R=10	3	4	4	4	5	3	3	10	10	10
	R=20	3	4	4	4	5	3	3	20	20	20
NLS	R=7	3	4	5	4	5	7	7	4	5	5
	R=10	3	4	4	4	5	3	3	10	10	10
	R=20	3	4	4	4	5	3	3	20	20	20
R-CPD	R=7	3	4	6	5	6	5	6	6	6	6
	R=10	10	10	10	10	10	10	10	10	10	10
	R=20	20	20	20	20	20	20	20	20	20	20
BCVMFB	R=7	7	7	7	7	7	7	7	7	7	7
	R=10	10	10	10	10	10	10	10	10	10	10
	R=20	20	20	20	20	20	20	20	20	20	20
OSNNCPD1	R=7	3	4	5	3	5	4	5	4	5	5
	R=10	3	4	5	3	5	4	5	4	5	5
	R=20	3	4	5	3	5	4	5	4	5	5
OSNNCPD2	R=7	3	4	4	4	5	4	5	4	5	5
	R=10	3	4	5	3	5	4	5	4	5	5
	R=20	3	4	5	3	5	5	4	4	7	7
ROSNNCPD	R=7	3	4	5	3	5	4	5	4	5	5
	R=10	3	4	5	3	5	4	5	4	5	5
	R=20	3	4	5	3	5	4	5	4	5	5

TABLEAU 4.10 – Estimation du rang avec 50% de chevauchement.

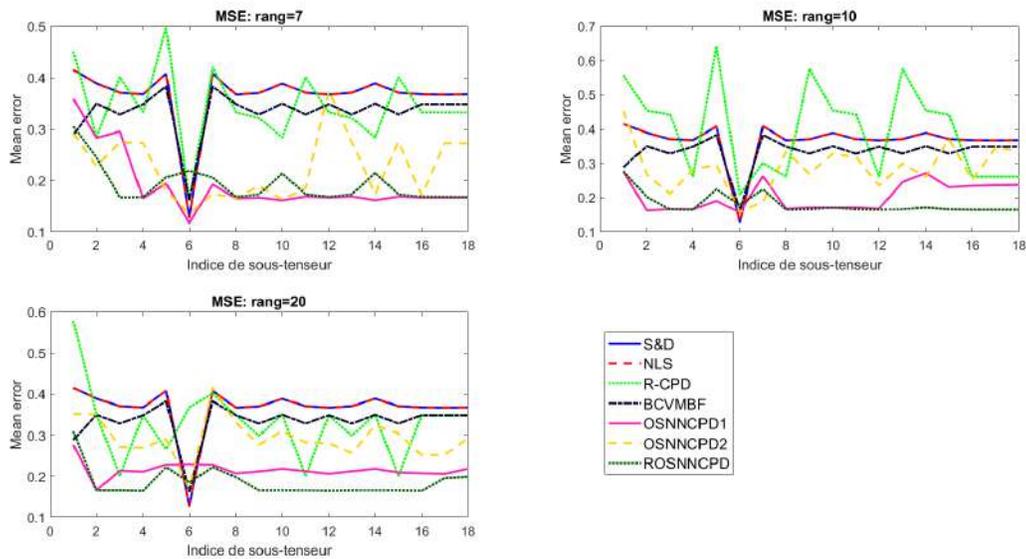


FIGURE 4.24 – Erreur moyenne ( $E_A, E_B, E_C$ ) pour le cas avec 50% de chevauchement.

**Erreur moyenne sur les matrices facteurs** Comme les rangs estimés par les algorithmes SeekAndDestroy, NLS, R-CPD et BCVMFD sont différents des rangs réels des sous-tenseurs, la Figure 4.24 montre que les erreurs moyennes sur les matrices de facteurs sont importantes dans la plupart des cas. A l'inverse, OSNNCPD1, OSNNCPD2 et ROSNNCPD produisent de faibles erreurs sur les matrices de facteurs car les rangs estimés par ces algorithmes sont globalement exacts (tableau 4.10).

Alors que les erreurs obtenues par OSNNCPD1 et OSNNCPD2 peuvent varier, ROSNNCPD donne des erreurs moyennes assez stables pour un nombre important de sous-tenseurs.

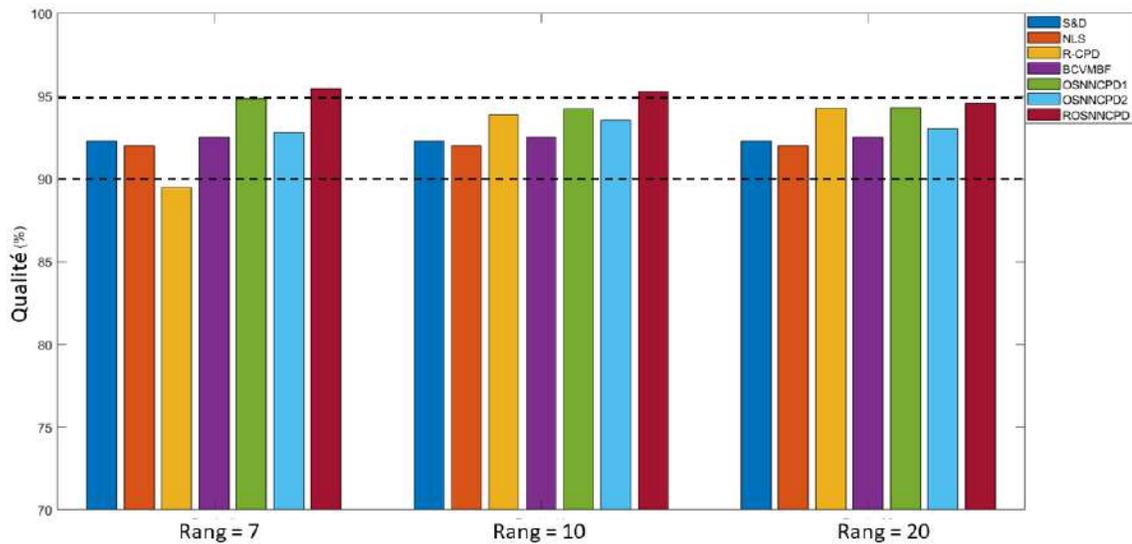


FIGURE 4.25 – Qualité de reconstruction relative pour le cas avec 50% de chevauchement.

**Estimation de la qualité de reconstruction** Les algorithmes OSNNCPD1, ROSNNCPD ont les meilleures qualités de reconstruction relative probablement parce qu'ils utilisent les matrices obtenues à l'instant  $t_{n-1}$ . Dans ce cas, ROSNNCPD obtient les meilleurs résultats. La stabilité des résultats par rapport à  $R$  confirme ici le bon comportement de la contrainte de parcimonie.

### 4.3.2.3 Cas avec 90% chevauchement de données

**Estimation du rang** Dans ce cas, les algorithmes sont appliqués à 91 sous-tenseurs mais pour les mêmes raisons évoquées précédemment, seuls 10 sous-tenseurs principaux ( $\mathcal{T}_0, \mathcal{T}_{11}, \mathcal{T}_{21}, \mathcal{T}_{31}, \mathcal{T}_{41}, \mathcal{T}_{51}, \mathcal{T}_{61}, \mathcal{T}_{71}, \mathcal{T}_{81}, \mathcal{T}_{91}$ ) sont présentés pour l'estimation du rang. Les résultats des différents algorithmes sont assez similaires (sauf pour R-CPD) par rapport au cas précédent, comme le montre le tableau 4.11.

Les algorithmes de l'état de l'art (SeekAndDestroy, NLS et BCVMBF) tendent à surestimer le rang pour chaque sous-tenseur en favorisant l'apparition de composants dupliqués. Inversement ROSNNCPD et OSNNCPD1 trouvent le bon rang pour les sous-tenseurs et quelle que soit la valeur

Sous-tenseur $\mathcal{T}_n$		$\mathcal{T}_0$	$\mathcal{T}_{11}$	$\mathcal{T}_{21}$	$\mathcal{T}_{31}$	$\mathcal{T}_{41}$	$\mathcal{T}_{51}$	$\mathcal{T}_{61}$	$\mathcal{T}_{71}$	$\mathcal{T}_{81}$	$\mathcal{T}_{91}$
<b>Rang physique <math>\tilde{R}</math></b>		<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
SeekAndDestroy	R=7	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	10	10	10
	R=20	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	20	20	20
NLS	R=7	<b>3</b>	<b>4</b>	<b>5</b>	4	<b>5</b>	3	3	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	10	10	10
	R=20	<b>3</b>	<b>4</b>	4	4	<b>5</b>	3	3	20	20	20
R-CPD	R=7	<b>3</b>	<b>4</b>	4	4	3	3	3	3	3	3
	R=10	<b>3</b>	<b>4</b>	4	4	3	3	3	3	3	3
	R=20	<b>3</b>	<b>4</b>	4	4	3	3	3	3	3	3
BCVMFB	R=7	7	7	7	7	7	7	7	7	7	7
	R=10	10	10	10	10	10	10	10	10	10	10
	R=20	20	20	20	20	20	20	20	20	20	20
OSNNCPD1	R=7	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=20	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
OSNNCPD2	R=7	<b>3</b>	<b>4</b>	4	4	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=20	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	5	4	<b>4</b>	<b>5</b>	7
ROSNNCPD	R=7	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=10	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>
	R=20	<b>3</b>	<b>4</b>	<b>5</b>	<b>3</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>4</b>	<b>5</b>	<b>5</b>

TABLEAU 4.11 – Estimation du rang avec 90% de chevauchement.

de  $R$ . C'est aussi le cas de OSNNCPD2 pour  $R = 10$ , mais des erreurs sont observées pour  $R = 7$  et  $R = 20$ .

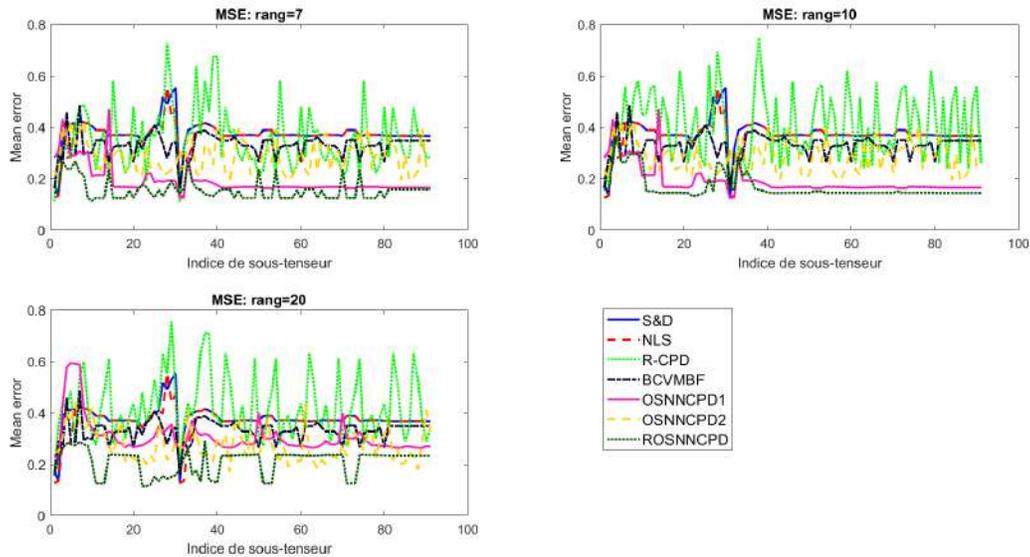


FIGURE 4.26 – Erreur moyenne ( $E_A, E_B, E_C$ ) pour le cas avec 90% de chevauchement.

**Erreur moyenne sur les matrices facteurs** La Figure 4.26 montre l'évolution de l'erreur moyenne sur les matrices facteurs pour les trois valeurs différentes de  $R$ . Avec  $R = 7$  ou  $R = 10$ , ROSNNCPD et OSNNCPD1 présentent les erreurs les plus faibles pour la plupart des sous-tenseurs. On note aussi que la qualité d'estimation augmente avec le temps. OSNNCPD2 est moins efficace et moins stable mais il fournit toujours de meilleurs résultats que ROSNNCPD et OSNNCPD1. Pour  $R = 20$ , ROSNNCPD fournit clairement les meilleurs résultats et OSNNCPD2 est meilleur que OSNNCPD1.

**Estimation de la qualité de reconstruction** Les qualités de reconstruction sont tracées sur la Figure 4.27 pour les différents algorithmes. Nous observons une légère amélioration de la qualité relative de la reconstruction par rapport au cas de chevauchement à l'exception de R-CPD dont les paramètres ont manifestement été mal choisis ici. Les trois algorithmes proposés permettent de reconstruire correctement les données surtout pour  $R = 20$ .

Cependant, il n'y a pas de corrélation claire entre la qualité relative de la reconstruction et l'erreur moyenne sur les matrices factorielles estimées. Ici, l'intérêt est d'obtenir une qualité de reconstruction stable. Par exemple, OSNNCPD1 estime assez mal les matrices facteurs mais reconstruit très bien le tenseur. OSNNCPD2 estime bien les matrices facteurs et reste stable sur la qualité de reconstruction. Egalement, ROSNNCPD reste stable sur la qualité de reconstruction et il estime bien les matrices facteurs.

### 4.3.3 Temps d'exécution

En plus des critères d'estimation, nous donnons le temps d'exécution de nos trois algorithmes pour chaque cas de chevauchement de données.

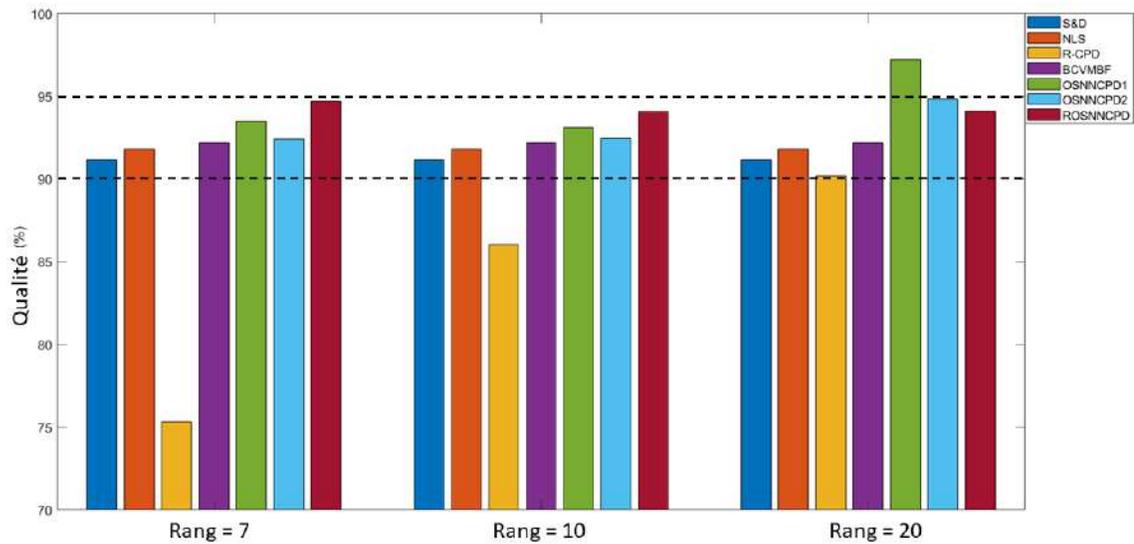


FIGURE 4.27 – Qualité de reconstruction relative pour le cas avec 90% de chevauchement.

Rang	7	10	20	$\alpha$
OSNCPD1	3.6	3.6	3.6	5
OSNCPD2	4.91	5.3	5.3	1
ROSNCPD	3.1	2.5	3.4	5

TABLEAU 4.12 – Moyenne des temps d'exécution (s) des 3 algorithmes dans le cas sans chevauchement.

**Cas sans chevauchement de données** Le tableau 4.12 montre que l'algorithme ROSNNCPD est plus rapide que les algorithmes OSNNCPD1 et OSNNCPD2 du fait du calcul récursif du gradient. OSNNCPD2 est plus lent que OSNNCPD1 de fait de sa complexité numérique supérieure (les dimensions des matrices à estimer sont plus importantes).

Rang	7	10	20	$\alpha$
OSNNCPD1	3.6	3.6	3.6	5
OSNNCPD2	4.9	5.3	5.3	1
ROSNNCPD	1.1	1.8	2.6	5

TABLEAU 4.13 – Moyenne des temps d'exécution (s) des 3 algorithmes dans le cas de recouvrement à 50%.

**Cas avec 50% de chevauchement** Le tableau 4.13 montre que les temps d'exécution de ROSNNCPD diminuent : cela s'explique par le fait que les algorithmes convergent plus rapidement grâce à l'algorithme récursif. Avec un chevauchement, les MEEFs partagent des facteurs en commun et ainsi, ROSNNCPD, en se basant des facteurs estimés précédemment, converge rapidement.

Rang	7	10	20	$\alpha$
OSNNCPD1	2.1	3.4	3.6	5
OSNNCPD2	4.9	5.3	5.3	8
ROSNNCPD	1.1	1.2	2.4	5

TABLEAU 4.14 – Moyenne des temps d'exécution (s) des 3 algorithmes dans le cas de recouvrement à 90%.

**Cas avec 90% de chevauchement** Le tableau 4.14 montre que les temps d'exécution de ROSNNCPD diminuent comme dans le cas précédent. Ceci s'explique par le fait que ROSNNCPD converge plus rapidement car plusieurs sous-tenseurs ont des MEEFs en commun. On peut aussi mentionner que OSNNCPD1 converge plus vite car, avec l'annulation du gradient des matrices facteurs estimés à l'instant précédent et aussi avec un taux de chevauchement élevé, il y a de fortes chances d'avoir les mêmes facteurs à estimer. Les hyperparamètres utilisés sont donnés dans le tableau 4.15.

Dans un problème inverse comme le nôtre, le coefficient de pénalité  $\alpha$  joue un rôle non négligeable car il permet d'accentuer ou non la parcimonie. Le paramètre  $\alpha$  agit sur la parcimonie de la solution, de sorte qu'habituellement, il est souvent nécessaire de trouver le bon  $\alpha$  en procédant par tâtonnement, c'est-à-dire, de tester différentes valeurs pour trouver le chemin de la solution au fur et à mesure que  $\alpha$  tend vers 0. Des travaux se sont intéressés à la recherche de la meilleure valeur du coefficient de pénalité [169, 170] et ces travaux ont montré que la solution optimale du problème suit un chemin linéaire par morceaux.

Dans notre cas, ce coefficient diffère lorsqu'on utilise les données de fluorescence synthétiques des données de fluorescence réelles (cf. tableau 4.1). Nos algorithmes en ligne présentés ici sont légèrement moins sensibles à ce coefficient. Notons que plus ce coefficient est grand, plus la partie régularisation de la fonction de coût devient prioritaire et cela conduit à une annulation des fluorophores avec une concentration faible. D'autre part, le fait d'utiliser l'algorithme d'optimisation

Nadam est un avantage car l'algorithme Nadam assure une meilleure convergence que la descente de gradient classique, aussi, la mise à jour du gradient (Cf chapitre 3) est faite de façon simultanée. Afin, on peut ajouter la contrainte de non-négativité appliquée. Nous pouvons ajouter aussi que nos algorithmes utilisent les matrices facteurs estimées précédemment.

<b>Algorithmes</b>	$\beta_1$	$\beta_2$	$\gamma$	$\epsilon$
SNNCPD	0.9	0.99	$10^{-3}$	$10^{-6}$
OSNNCPD1	0.9	0.99	$10^{-1}$	$10^{-6}$
OSNNCPD2	0.9	0.99	$10^{-5}$	$10^{-6}$
ROSNNCPD	0.9	0.99	$10^{-1}$	$10^{-6}$

**TABLEAU 4.15** – Les différents hyperparamètres utilisés par chaque algorithme pour ce jeu de données.

## 4.4 Conclusion du chapitre

Dans ce chapitre, nous avons généré et utilisé des séries de MEEFs simulées afin de valider nos approches sur des données entièrement contrôlées. Une première expérience sur ces données synthétiques a été réalisée pour calibrer les hyperparamètres des algorithmes et pour connaître leur comportement. Puis, une expérience en conditions réelles contrôlées de spectroscopie de fluorescence en ligne a été menée en laboratoire afin de comparer nos algorithmes avec des approches connues.

Les algorithmes proposés permettent de suivre correctement les variations de rang dans la plupart des situations considérées contrairement aux approches de référence comme, par exemple, R-CPD et AutoTen. En spectroscopie de fluorescence, on sait maintenant que l'erreur moyenne sur les facteurs estimés permet d'améliorer l'identification des fluorophores. En d'autres termes, une faible erreur moyenne sur les matrices facteurs signifie que l'on peut visuellement identifier les fluorophores. Concernant ce critère, notre approche donne généralement de meilleurs résultats que les autres stratégies en ligne. Plus précisément, lorsqu'un recouvrement entre sous-tenseurs consécutifs est autorisé OSNNCPD1 et ROSNNCPD fournissent clairement les meilleurs résultats. Dans ce cas, les facteurs sont également estimés avec plus de précision et les algorithmes sont plus stables. A l'inverse, dans le cas sans chevauchement, OSNNCPD2 apparaît comme la meilleure option (mais il est plus judicieux de surestimer le rang). Finalement, dans la plupart des situations, ROSNNCPD fournit des résultats légèrement meilleurs que OSNNCPD1. C'est aussi l'algorithme le plus rapide. Nous recommandons donc de combiner chevauchement des sous-tenseurs et ROSNNCPD dans le cas de la décomposition en ligne des mesures de matrice d'excitation et d'émission de fluorescence.

Après ces tests sur des données synthétiques et contrôlées en laboratoire, il sera nécessaire de voir comment ces algorithmes se comportent en milieu naturel dans une expérience en ligne et *on field*.

# Chapitre 5

## Application sur des données de fluorescence *on field*

Ce chapitre va nous permettre de mettre à l'épreuve les algorithmes de CPD en ligne proposés sur des données mesurées *on field* pour la surveillance d'un environnement aquatique utilisant la technique de fluorescence 3D en ligne. L'idée sera de suivre en temps réel les composants détectés lors de la phase d'initialisation et de voir s'il y a des apparitions ou des disparitions de composants au cours du temps. Cette surveillance en temps réel permet de caractériser le milieu à surveiller et de tirer des conclusions sur la présence de certains contaminants biologiques ou organiques.

Il faut savoir que les résultats d'une décomposition CPD peuvent être une indication sur la présence ou non de contaminants biologiques et chimiques dans le milieu [15] mais d'autres travaux montrent qu'une conclusion est difficile à tirer à partir de ces seuls résultats [26]. A titre d'exemple, le signal du tryptophane indique généralement une contamination microbienne possible dans un échantillon d'eau [171] mais il a aussi été associé à une source autochtone de matière fraîche [172]. Pour faire la différence, il est important de pouvoir décomposer le signal de fluorescence 3D au cours du temps. Pour cela, une expérience d'acquisition de données de fluorescence en ligne est mise en place sur laquelle nous testerons une décomposition après acquisition. Il faut noter que, contrairement aux données des chapitres précédents, nous travaillons à l'aveugle sans connaissance des composants de référence. Dans un milieu aquatique complexe, nous nous attendons à observer principalement la MOD et les composés fluorescents habituels que l'on peut observer comme la tyrosine ou le tryptophane.

La MOD constitue un mélange complexe et hétérogène de macromolécules dont la taille peut varier entre 100 et 100000 Daltons [173]. Celle-ci peut être issue de l'activité des microorganismes, de débris végétaux ou de résidus d'animaux. Elle peut être également issue de la production du phytoplancton dans les eaux, comme le tryptophane cité précédemment. Il est encore difficile à l'heure actuelle de définir une liste de la totalité de sa composition. La MOD peut être divisée en deux groupes de composés tels que les substances humiques (SH) et les substances non-humiques (SNH) [174].

D'une manière générale, les fluorophores observés dans les MEEFs en environnement sont localisés en fonction de leur couple de longueur d'onde d'excitation et d'émission de leur maximum depuis [175]. Une représentation non exhaustive des régions de fluorescence a été dressée dans [4] permettant une caractérisation des principaux groupes de fluorescence présents dans un milieu naturel (Figure 5.1).

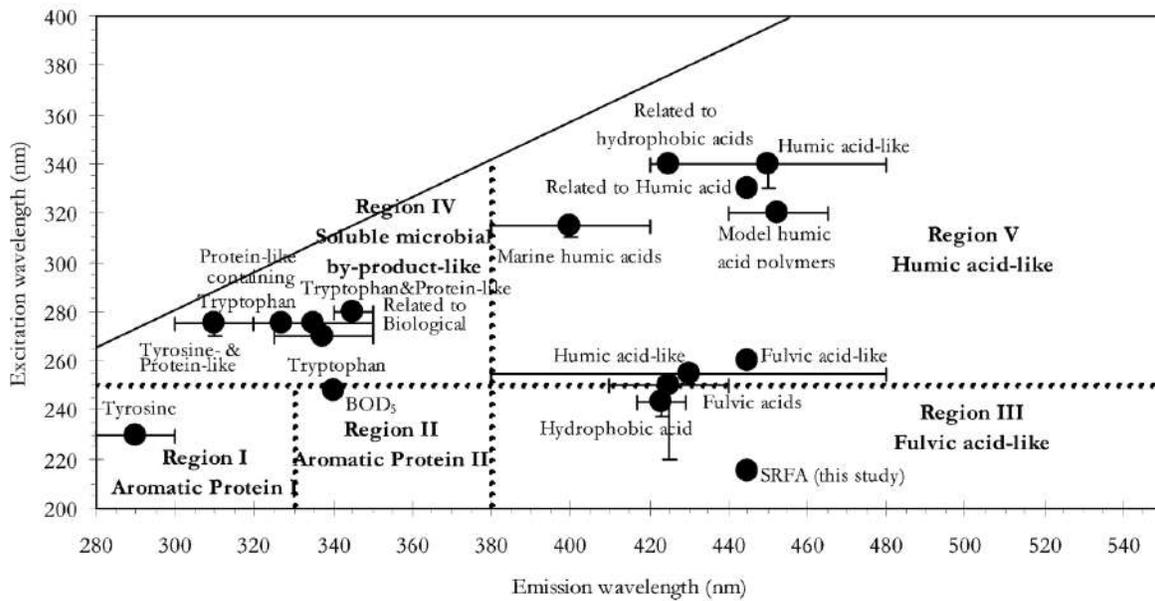


FIGURE 5.1 – Cartographie des principaux fluorophores en environnement. Image tirée de [4].

## 5.1 Dispositif expérimental

### 5.1.1 Descriptif du site



FIGURE 5.2 – Arrivé de l'eau de mer au sein du laboratoire SSL@MM.

L'acquisition a été faite dans le Sea Water Sensing Laboratory @ MIO Marseille (SSL@MM). Le SSL@MM est un laboratoire marin situé à la station marine historique d'Endoume dans la ville de Marseille comme l'indique la Figure 5.3.

Ce laboratoire a été entièrement pensé pour amener l'eau de mer de la baie vers les laboratoires et assurer la représentativité des conditions *in situ* en termes de communauté microbienne, de propriétés chimiques et d'hydrologie (<https://sslamm.mio.osupytheas.fr/>) (Figure 5.2). Les écosystèmes pélagiques marins sont complexes et dynamiques. Ils sont influencés soit par des événements physiques déclenchant des distributions biologiques et chimiques à des échelles infimes, soit par des processus liés aux réactions rapides des microorganismes planctoniques grâce à l'adaptation des activités cellulaires intrinsèques. Pour cela, il faut acheminer l'eau de la mer à l'intérieur du laboratoire avec une certaine précaution. L'eau de mer est préfiltrée à l'aide d'une crépine à eau située à 60 cm de haut et à mailles de 2 mm positionnée à l'entrée du pompage. Les tuyaux sont en polypropylène. Les systèmes hydrauliques à l'intérieur du laboratoire sont nettoyés chimiquement deux fois par mois avec 1%  $H_2O_2$  pendant 4 heures et à l'extérieur avec de l'eau



FIGURE 5.3 – Situation géographique du site (Endoume).

douce à haute pression en utilisant des buses de nettoyage de tuyaux spéciales qui éliminent la saleté des tuyaux.

Pour optimiser l'intégrité des micro-organismes et réduire les turbulences dans l'eau de mer pompée, une pompe péristaltique massive a été installée à environ 5 m au-dessus du niveau de la mer. La pompe péristaltique est utilisée à 30% de sa capacité (Figure 5.4), ce qui donne un pompage continu avec un scénario d'utilisation stable et à long terme. Le tube péristaltique est en EPDM et doit être changé tous les 3-4 mois d'utilisation.

### 5.1.2 Dispositif de mesure des signaux de fluorescence 3D

Le dispositif de mesure de fluorescence est similaire à l'acquisition de données expliquée dans le chapitre précédent. Nous disposons d'un spectrofluorimètre Hitachi F7000, d'une pompe péristaltique et d'un Raspberry équipé d'une caméra. D'abord, l'eau de mer est acheminée via la pompe péristaltique, dans une éprouvette de 500 mL permettant la décantation de particules éventuelles. Ensuite, à l'aide d'un tube, nous utilisons notre pompe péristaltique pour pomper l'eau de l'éprouvette vers le spectrofluorimètre pour effectuer l'acquisition comme le montre la Figure 5.5.

Les MEEFs sont alors enregistrées avec le spectrofluorimètre *Hitachi F7000* (Figure 4.16) chaque 8 mn avec une cellule à circulation en quartz de 0,5 cm de chemin optique alimenté par un débit de 1 mL.min<sup>-1</sup>. La tension du tube PM est de 950 V et la vitesse de balayage est de 12000 nm.min<sup>-1</sup>. Les MEEFs sont mesurées dans la gamme de longueurs d'onde d'excitation de 220 nm à 900 nm (fente 10 nm) et dans la gamme de longueurs d'onde d'émission de 220 nm à 900 nm (fente 10 nm). Un pas de 10 nm est utilisé à la fois en excitation et en émission. La taille des MEEFs est donc de (69, 69) (Figure 5.6).

Un Raspberry Pi 4 (Figure 4.14) muni d'une caméra nous permet de surveiller le bon fonctionnement de l'ensemble de l'expérimentation à distance et en ligne. Pour cela, un programme en Python de détection de goutte d'eau à travers le module caméra en temps réel a été développé dans le cadre d'un projet étudiant de 2ème année de l'école d'ingénieur Seatech par Asmae Majdoub et Zineb Bzair sous ma supervision, le tout pouvant être visionné à distance et à chaque instant (Figure 5.7). En parallèle, nous comptons le nombre de gouttes par heure à travers la détection afin de surveiller le débit du circuit.



FIGURE 5.4 – La pompe péristaltique.

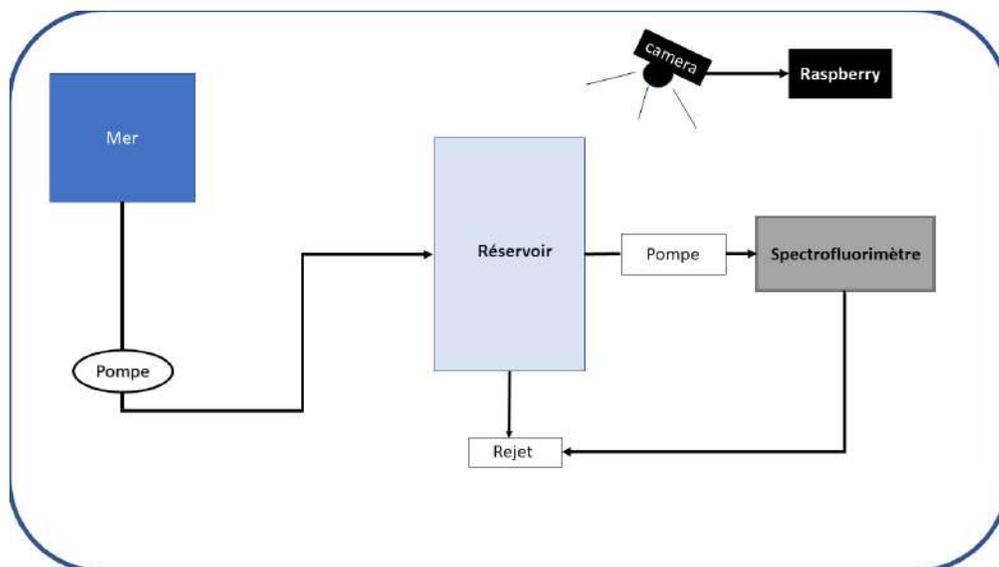


FIGURE 5.5 – Schéma du dispositif d'acquisition en ligne.



**FIGURE 5.6 – Schéma du dispositif d'acquisition en ligne.**



**FIGURE 5.7 – Surveillance du dispositif en ligne.**

## 5.2 Décomposition en ligne des signaux

### 5.2.1 CPD en ligne

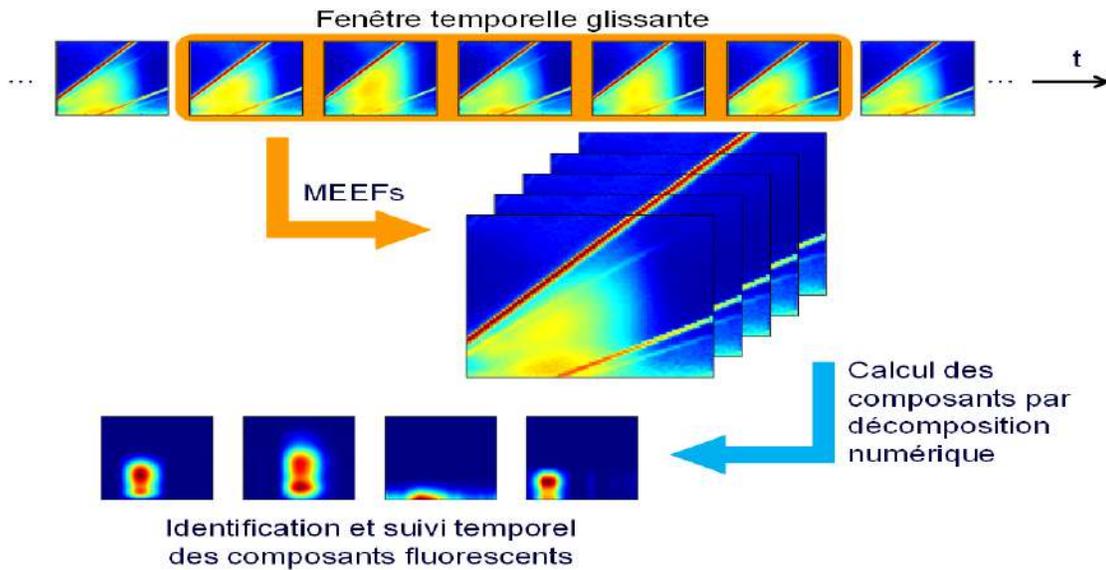


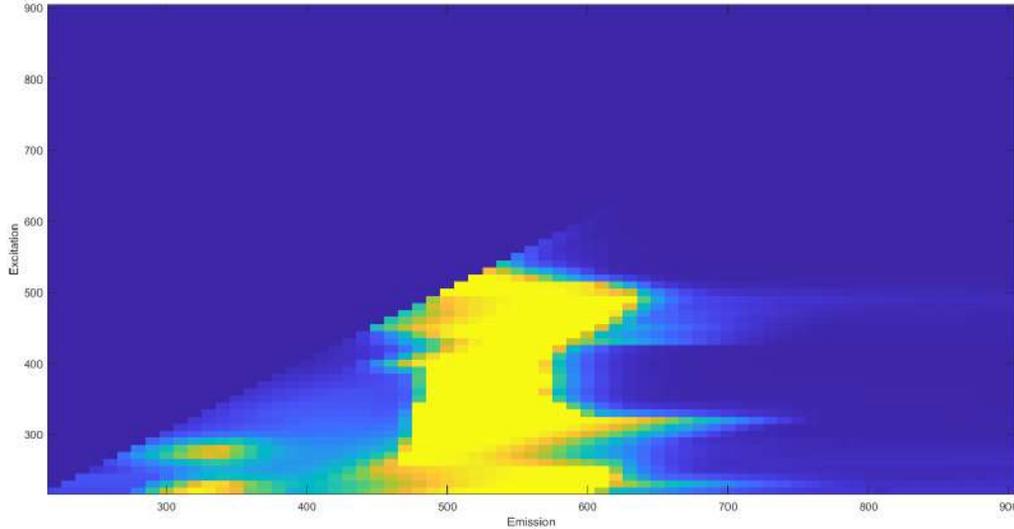
FIGURE 5.8 – Rappel du principe du traitement des MEEFs en ligne et in situ.

L'expérimentation nous a permis d'acquérir en tout 1396 MEEFs en ligne. Pour des questions pratiques, nous enregistrons ces MEEFs dans un premier temps et dans un second temps nous choisissons une fenêtre glissante pour le traitement numérique en utilisant nos différents algorithmes comme l'indique la Figure 5.8.

Avant d'appliquer nos différents algorithmes, nous devons faire les pré-traitements d'usage comme la suppression de la diffusion Raman et Rayleigh. La méthode de Zepp a été utilisée pour ce traitement. Nous ne tenons pas compte ici des autres effets non linéaires peu probables comme l'effet d'écran (absorbance faible de l'eau de mer) ou la présence de particules (filtrées en amont). Un "tri" (manuel, pour l'instant) est nécessaire avant de lancer la phase de l'initialisation sur ce type de données réelles afin de bien prendre en compte une bonne initialisation avant de démarrer la phase en ligne. Ce tri consiste à parcourir les MEEFs manuellement et à supprimer les MEEFs anormales comme le montre la Figure 5.9

Deux fenêtres de chevauchement des données pour former les sous-tenseurs sont utilisées durant l'application de nos algorithmes, à savoir : (i) cas sans chevauchement des données, (ii) 99% de chevauchement des données. Dans le deuxième cas, cela signifie que chaque nouveau sous-tenseur correspond à l'acquisition d'une seule nouvelle MEEF et à l'élimination de la première MEEF du précédent sous-tenseur.

OSNNCPD1, OSNNCPD2 et ROSNNCPD sont les algorithmes que nous retenons pour le traitement en ligne. Le choix est dicté par les résultats obtenus sur les données simulées du chapitre 4 et les données semi-réelles car ces algorithmes surclassent les algorithmes de la littérature. Ces



**FIGURE 5.9 – Image de la MEEF anormale trouvée lors du tri manuel (qui a été supprimée) avant la CPD et après suppression des phénomènes de diffusion.**

approches prennent en compte la possibilité d'apparition et de disparition d'un ou de plusieurs composants d'un sous-tenseur à l'autre. Rappelons que pour chaque cas de chevauchement, nous avons toujours deux phases, la phase d'initialisation suivie de la phase en ligne.

Pour cette surveillance en ligne, nous donnons comme critère l'erreur de reconstruction (5.1) pour chaque sous-tenseur, ce qui permet une représentation globale du comportement de chaque algorithme sur l'ensemble des sous-tenseurs :

$$E_{\mathcal{T}} = \frac{\|\mathcal{T} - \widehat{\mathcal{T}}\|_F}{\|\mathcal{T}\|_F} \quad (5.1)$$

avec  $\mathcal{T}$  le tenseur de départ et  $\widehat{\mathcal{T}}$  le tenseur reconstruit. Nous posons comme critère de performance une erreur de reconstruction pour un sous-tenseur inférieure à 0.25.

**Phase d'initialisation** Afin d'initialiser les matrices facteurs nous utilisons l'algorithme SNNCPD avec un rang fixé  $R=7$  sur le premier sous-tenseur  $\mathcal{T}_0(69, 69, 10)$  dont les MEEFs sont données sur l'image 5.10. Le pas de l'algorithme SNNCPD est égal à  $\gamma = 10^{-3}$  au début de processus de la convergence, le coefficient de pénalité est fixé à  $\alpha = 2$ , les hyperparamètres de l'algorithme Nadam  $\beta_1 = 0,9$  et  $\beta_2 = 0,99$ .

Nous trouvons trois composants significatifs comme l'indique la Figure 5.11. Leurs spectres d'excitation et d'émission sont représentés sur la Figure 5.12. Contrairement à l'expérimentation réalisée en laboratoire, nous ne connaissons pas les composants présents dans l'eau de mer prélevée. De ce fait, nous allons caractériser grossièrement les spectres trouvés dans cette phase d'initialisation grâce au tableau 5.1 qui rassemble les caractéristiques des principaux fluorophores identifiés dans la littérature [4].

D'après ce tableau, le premier composant en rouge peut être identifié comme la tyrosine, le deuxième composant en bleu foncé est vu comme une substance de type humique et enfin le dernier

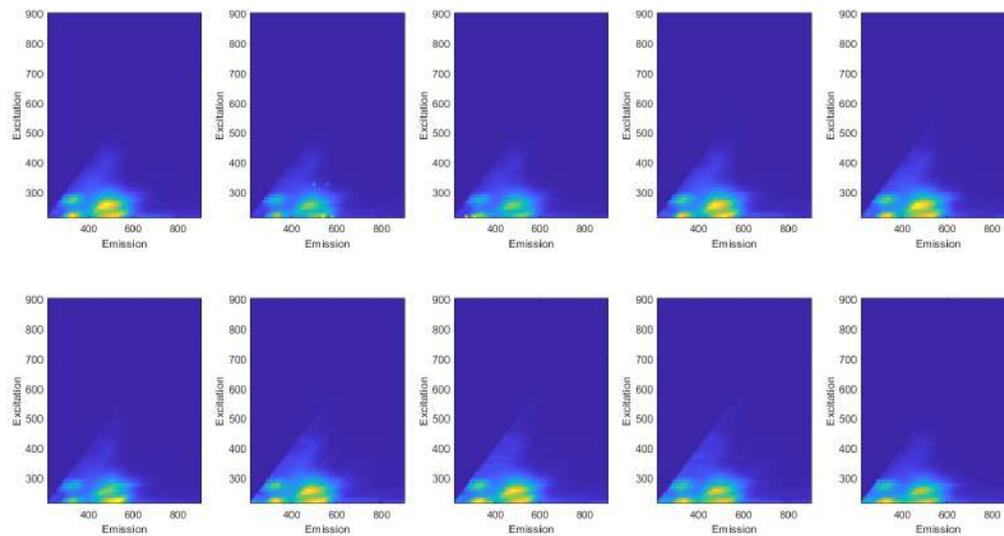


FIGURE 5.10 – Image des MEEFs du sous-tenseur  $\mathcal{T}_0$ .

Fluorophores [176]	Fluorophores [177]	Longueur d'onde d'excitation (nm)	Longueur d'onde d'émission (nm)	Type de composés
A	$\alpha$	230 – 260	380 - 480	Substances type humiques
C	$\alpha'$	330 – 370	420 - 480	Substances type humiques
M	$\beta$	310 – 320	380 - 420	composante biologique
B	$\gamma$	270 – 280	300 - 320	Tyrosine
T	$\delta$	270 – 280	320 - 380	Tryptophane

TABLEAU 5.1 – Résumé des fluorophores décrits dans la littérature.

composant est de la matière organique. Ces composants sont ceux que l'on peut observer lors d'une surveillance de l'eau de mer et qui indiquent la possible présence d'une activité bactérienne.

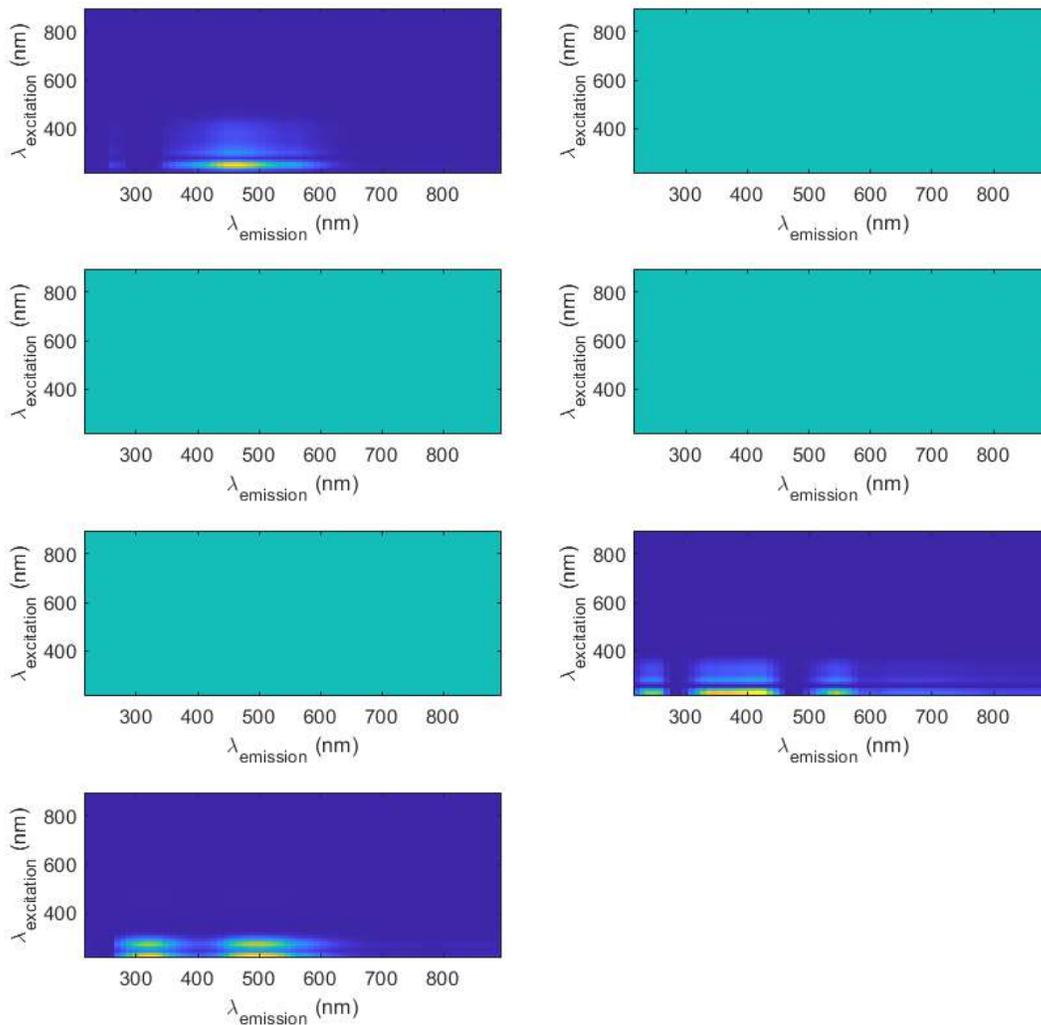


FIGURE 5.11 – Image des fluorophores présents dans le sous-tenseur  $\mathcal{T}_0$ .

**Cas sans chevauchement de données** Dans ce cas, chaque sous-tenseur est de taille (69,69,10) et nous surestimons le rang de la CPD  $R$  qui est fixé à 7 lors de l'expérimentation. Après la phase d'initialisation, nous utilisons nos algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD pour la phase en ligne.

La Figure 5.13 nous montre l'erreur de reconstruction sur les sous-tenseurs produite par les algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD. Nous remarquons que les erreurs de reconstruction de l'algorithme OSNNCPD1 sur les sous-tenseurs sont relativement faibles et varient entre 0,1 et 0,2 à l'exception du sous-tenseur 57 qui a une erreur de reconstruction supérieur à 0,3. En

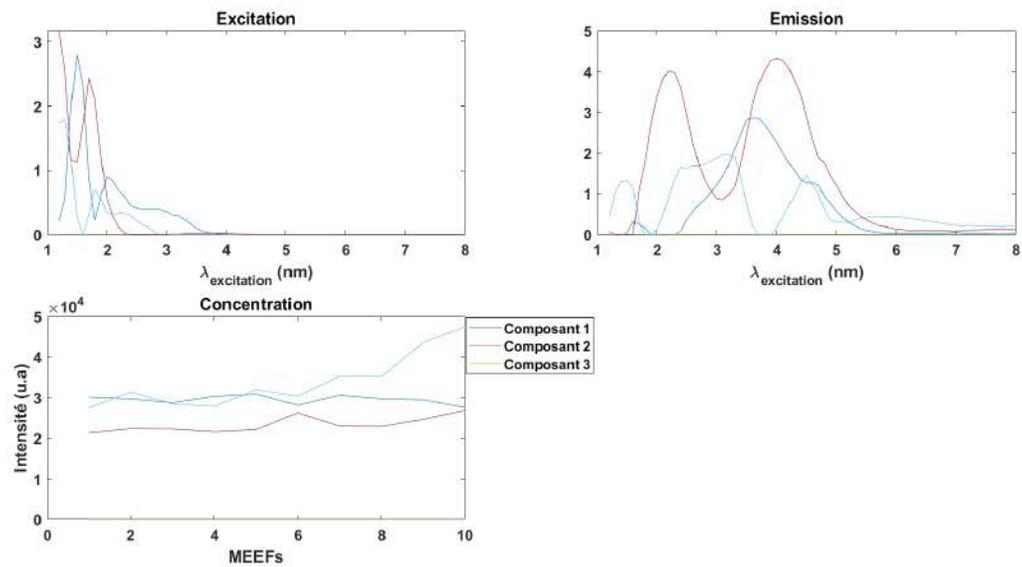


FIGURE 5.12 – Spectres d’excitation et d’émission des 3 fluorophores présents dans le sous-tenseur  $\mathcal{T}_0$ .

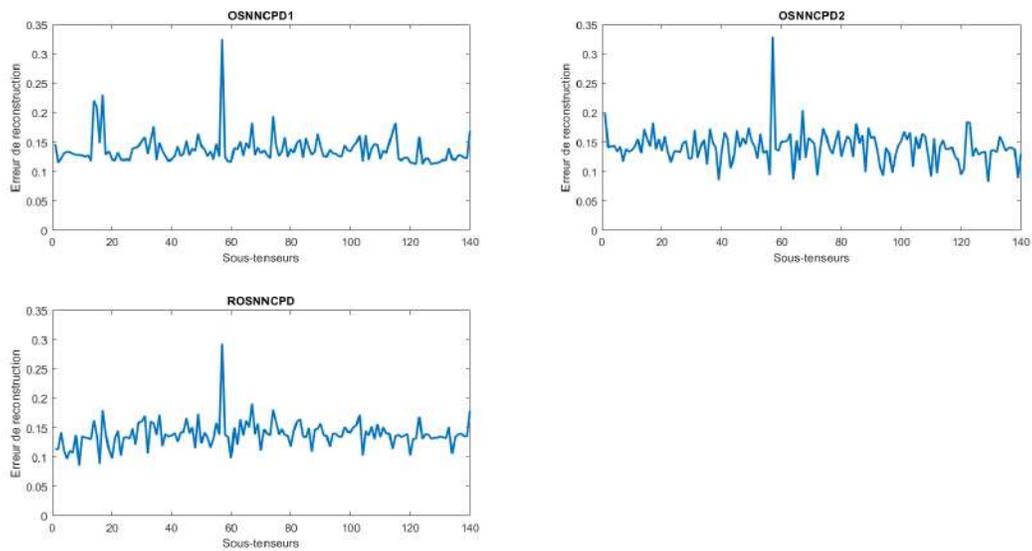


FIGURE 5.13 – Erreurs de reconstruction des algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD sur les sous-tenseurs. Cas sans chevauchement.

ce qui concerne l'algorithme OSNNCPD2, les erreurs de reconstruction sont globalement comprises entre 0,1 et 0,15 sauf pour les sous-tenseurs 57 et 138 qui ont une erreur de reconstruction supérieur à 0,2. Ces résultats semblent légèrement meilleurs que ceux obtenus précédemment avec l'algorithme OSNNCPD1. Cela s'explique par le fait qu'il n'y a pas de chevauchement de données qui pourrait aider OSNNCPD1. De même, l'algorithme ROSNNCPD produit des erreurs de reconstruction du même ordre que celles de OSNNCPD1. Le sous-tenseur 57 a toujours une erreur de reconstruction supérieur à 0,3.

Ce sous-tenseur donne une erreur de reconstruction élevée pour chaque algorithme, il est donc nécessaire de visualiser les MEEFs correspondantes. La Figure 5.14 nous montre les 10 MEEFs constituant ce sous-tenseur et nous remarquons des anomalies sur plusieurs d'entre elles ce qui corrobore notre hypothèse. Ces anomalies sont probablement dues à la présence de bulles d'air dans la cellule à circulation lors de l'acquisition de ces MEEFs.

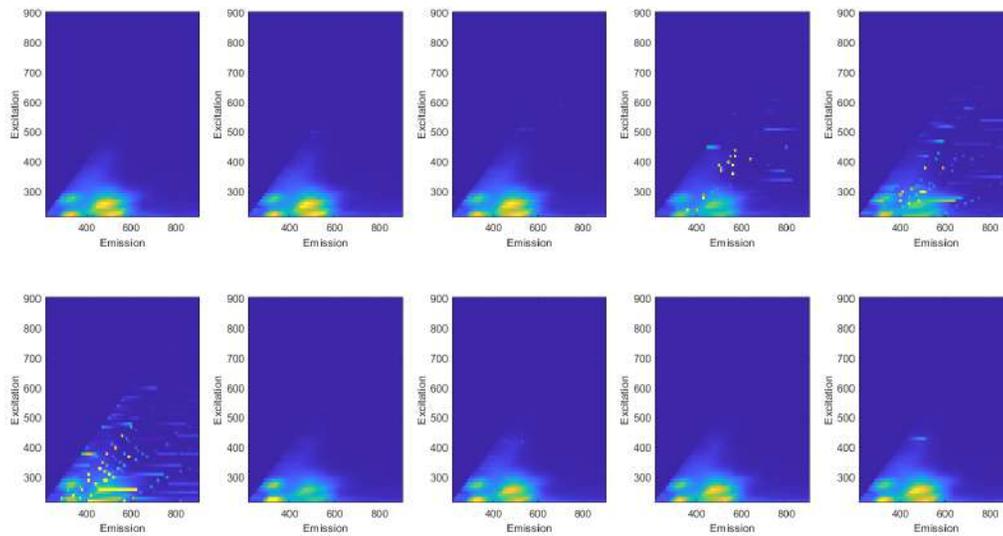


FIGURE 5.14 – Sous-tenseur 57 qui est constitué de MEEFs anormales.

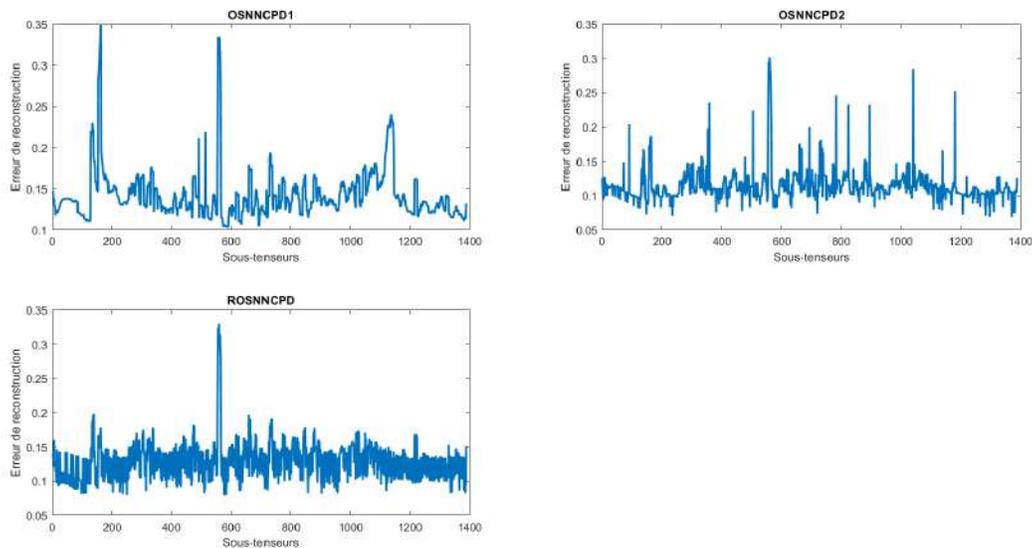
Algorithmes	Moyennes sur l'erreur de reconstruction	$\beta_1$	$\beta_2$	$\gamma$	$\alpha$	$\epsilon$	Temps d'exécution en seconde
OSNNCPD1	0.13	0.9	0.99	$10^{-2}$	1	$10^{-5}$	10.47
OSNNCPD2	0.11	0.9	0.99	$10^{-6}$	0.5	$10^{-5}$	9.21
ROSNNCPD	0.13	0.9	0.99	$10^{-1}$	1	$10^{-5}$	10.36

TABLEAU 5.2 – Les différents hyperparamètres utilisés par chaque algorithme.

Globalement, nos algorithmes convergent bien malgré la difficulté de l'expérience, parce que nous avons une erreur de reconstruction assez basse pour des données réelles. Le tableau 5.2 nous donne l'erreur moyenne de reconstruction de chaque algorithme, le temps d'exécution pour

chaque sous-tenseur et chaque algorithme ainsi que les différents hyperparamètres. Nous constatons que OSNNCPD2 donne une moyenne d'erreur de reconstruction plus faible que OSNNCPD1 et ROSNNCPD pour un même temps d'exécution. Il est donc l'algorithme à privilégier. Une perspective d'amélioration des résultats est de diminuer le coefficient de pénalité ainsi que le pas d'apprentissage.

**Cas avec 99 % chevauchement de données** Dans ce cas, cela revient à faire une décomposition après acquisition de chaque MEEF. L'avantage de ce cas est que chaque couple de sous-tenseurs consécutifs partage une grande quantité d'informations. Par contre sur une durée donnée, le nombre de sous-tenseurs à traiter sera beaucoup plus important. Dans le cas présent, au lieu de 140 sous-tenseurs sans chevauchement, nous nous retrouvons avec 1400 sous-tenseurs. Rappelons que chaque sous-tenseur est de taille (69,69,10) et nous surestimons le rang de la CPD  $R$  qui est fixé à 7 lors de l'expérimentation. Après la phase d'initialisation, nous comparons nos algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD pour la phase en ligne.



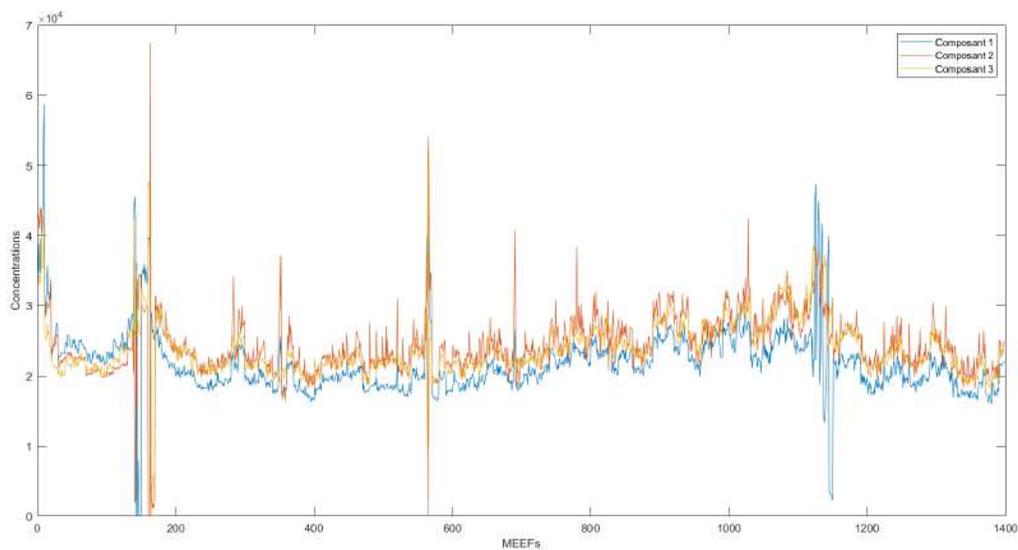
**FIGURE 5.15 – Erreur de reconstruction des algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD sur tous les sous-tenseurs. Cas avec chevauchement.**

La Figure 5.15 montre l'erreur de reconstruction sur les sous-tenseurs produite par les algorithmes OSNNCPD1, OSNNCPD2 et ROSNNCPD. Nous remarquons une sensible amélioration des résultats surtout pour ROSNNCPD qui voit son erreur de reconstruction comprise entre 0.1 et 0.15. C'est aussi le cas de OSNNCPD1 et OSNNCPD2 mais nous constatons une erreur élevée sur le sous-tenseur 570 pour chaque algorithme qui correspond bien au sous-tenseur avec les MEEFs anormales détectées précédemment (Figure 5.14). OSNNCPD1 et ROSNNCPD arrivent à stabiliser l'erreur de reconstruction après avènement des MEEFs anormales à la différence de OSNNCPD2.

Le tableau 5.3 donne l'erreur moyenne de reconstruction de chaque algorithme, le temps d'exécution moyen pour chaque sous-tenseur et chaque algorithme ainsi que les différents hyperparamètres. Nous constatons que cette fois-ci, ce sont OSNNCPD1 et ROSNNCPD qui donnent la moyenne d'erreur de reconstruction la plus faible par rapport à OSNNCPD2. Cela confirme que ce dernier est surtout utile pour les cas sans chevauchement.

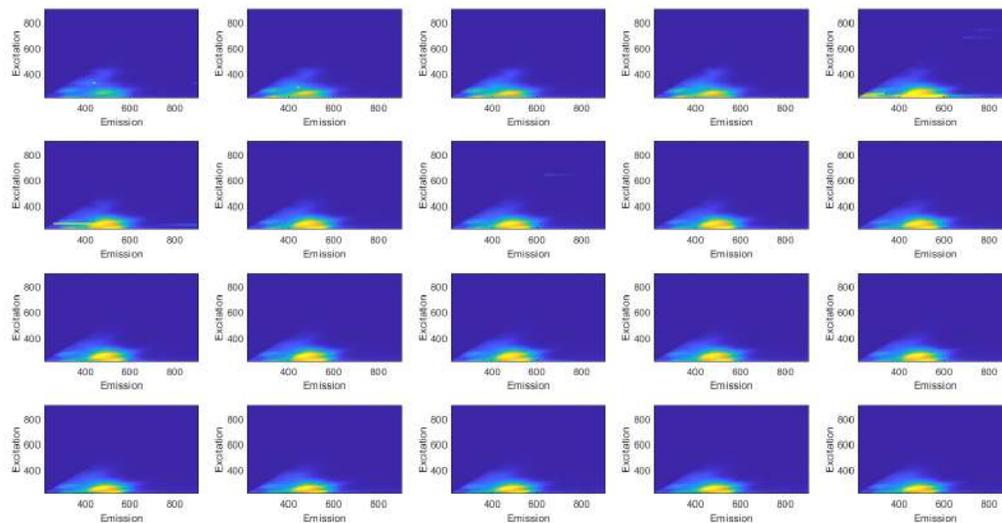
Algorithmes	Moyennes sur l'erreur de reconstruction	$\beta_1$	$\beta_2$	$\gamma$	$\alpha$	$\epsilon$	Temps d'exécution en seconde
OSNNCPD1	0.11	0.9	0.99	$10^{-2}$	1	$10^{-5}$	10.47
OSNNCPD2	0.13	0.9	0.99	$10^{-6}$	0.5	$10^{-5}$	9.21
ROSNNCPD	0.11	0.9	0.99	$10^{-1}$	1	$10^{-5}$	8.52

**TABLEAU 5.3 – Résumé de l'expérience avec les différents hyperparamètres utilisés par chaque algorithme. Cas avec chevauchement.**



**FIGURE 5.16 – Profil de concentration de chaque composant au cours du temps avec R=7.**

La Figure 5.16 montre les profils des concentrations de chaque composant au cours du temps avec  $R=7$ . Ici nous affichons directement les 7 profils de concentrations correspondant au rang de départ. Nous remarquons que nous avons 4 pics avec un composant en trop. Le premier pic se situe au niveau de la 147<sup>ème</sup> MEEF cela ressemble à une apparition d'un composant, ensuite, vers 170<sup>ème</sup> MEEF, vers 560<sup>ème</sup> MEEF, et enfin autour de 1150<sup>ème</sup> MEEF. Mais après investigation, il s'agit d'une anomalie de MEEFs (Figure 5.17) qui produit des composants doublons. Si nous regardons le deuxième pic, correspond à l'anomalie du sous-tenseur 57 décrite précédemment. Le dernier pic est une anomalie. Cette investigation, nous permet de dire que nous avons 4 sous-tenseurs où le rang n'est pas trouvé et que nous n'avons pas de disparition ni d'apparition de composant sur cette période.



**FIGURE 5.17 – Planche de MEEFs prises entre 140 et 160 MEEFs.**

### 5.3 Conclusion du chapitre

Dans ce chapitre, nous avons proposé une expérimentation visant à faire une surveillance aquatique *on field*. Il faut noter que nous n'avons pas d'informations a priori et pour cela, nous utilisons notre algorithme 6 (SNNCPD) pour l'initialisation. Après cette phase d'initialisation, nous avons suivi les différents composants détectés au cours du temps. Cette mise en pratique de nos algorithmes nous permet de faire une surveillance en ligne d'un milieu aquatique. Cependant, les conditions réelles sont nettement plus complexes que les conditions semi-réelles décrites dans le chapitre précédent : de nombreuses MEEFs sont altérées par des pics de diffusion dus à des passages de bulles d'air dans la cellule à circulation du spectrofluorimètre. Il est nécessaire de supprimer ces MEEFs avant tout traitement car nous pouvons avoir des erreurs sur la CPD comme nous l'avons montré. Une idée d'amélioration est d'ajouter à notre dispositif un appareil pouvant supprimer les bulles avant passage dans le spectrofluorimètre.

Nous avons pu observer durant l'expérimentation, la présence de composés comme la tyrosine qui donnent des indices sur la présence d'une activité bactérienne.

# Conclusion

## 5.4 Contributions

Dans ce travail, nous nous sommes intéressés à la décomposition nonnégative polyadique canonique en ligne de tenseurs. Nous avons proposé une approche originale pour calculer la décomposition en ligne de flux de sous-tenseurs qui prend en compte d'éventuelles variations de rang inconnues. En effet, cette approche basée sur l'apprentissage par dictionnaire parcimonieux permet de surestimer le rang (ou nombre de composants approprié) de la CPD en privilégiant des extra-facteurs à contribution nulle.

Nous avons d'abord présenté quelques notions mathématiques et introduit les modèles de décomposition de tenseurs. Par la suite, nous rappelons la relation existant entre le modèle CPD et l'intensité de fluorescence pour un couple de longueurs d'onde d'excitation et d'émission dans le cas particulier d'échantillons liquides à faible absorbance. L'intensité de fluorescence ayant toujours une valeur positive, une contrainte de nonnégativité est imposée dans notre modèle. Nous présentons également comment étendre le modèle CPD au modèle CPD en ligne dans le but de l'appliquer à la surveillance environnementale en temps réel à l'aide de la spectroscopie de fluorescence. Toutefois, dans un milieu aquatique, de nombreux phénomènes physiques peuvent faire varier le nombre de fluorophores présents, ce qui peut rendre le problème de la CPD difficile. En d'autres termes, ces phénomènes peuvent entraîner la disparition d'un ou de plusieurs composants ou bien l'apparition d'un ou de plusieurs composants.

Dans la littérature, il existe certains algorithmes de CPD en ligne comme Seek&Destroy, NLS, RLST, mais aussi des algorithmes dits hors ligne pouvant être adaptés en ligne (mais difficiles à mettre en place) comme R-CPD, BCVMBF. Pour un certain nombre d'algorithmes de CPD en ligne, le problème de variation n'est pas pris en compte ce qui limite leur usage dans le domaine de la fluorescence en ligne. Une méthode pour résoudre le problème du rang est d'imposer une contrainte de parcimonie au modèle trinéaire comme, par exemple, la norme  $L_{1,1}$ , la norme mixte, ou encore, la norme nucléaire. L'imposition de cette contrainte permet de déduire le nombre de composants après la convergence de l'algorithme de CPD. Cette approche peut être vue comme une méthode a posteriori de la détection du nombre de composants. Contrairement à cette approche, il existe des approches dites a priori qui consistent à estimer le nombre de composants avant chaque CPD par exemple CORCONDIA, AutoTen mais ces approches ne permettent pas toujours de décider clairement entre plusieurs valeurs de classement possibles.

Dans cette thèse, nous nous sommes focalisés sur l'approche parcimonieuse en appliquant la norme  $L_{1,1}$  comme contrainte en plus de la contrainte de nonnégativité et cela nous a permis de développer plusieurs algorithmes.

Nous avons ensuite retenu trois algorithmes principaux distincts de la CPD nonnégative en ligne après les résultats obtenus sur des données synthétiques de fluorescence simulées dans un premier temps, dans un second temps sur des données semi-réelles de fluorescence et enfin des données réelles

prélevées *in situ*. Dans le premier algorithme (OSNNCPD1), on suppose que les facteurs obtenus à partir de la CPD du premier sous-tenseur sont correctement estimés (étape hors ligne) avant de rechercher de nouveaux facteurs lors de l'analyse des sous-tenseurs suivants (étape en ligne). Bien sûr, cela permet aussi d'annuler les facteurs qui disparaîtraient au cours de l'étape en ligne tout en les gardant en mémoire. Le deuxième algorithme (OSNNCPD2) permet de mettre à jour les facteurs estimés précédemment en utilisant des combinaisons linéaires de grands dictionnaires. Cet algorithme offre plus de flexibilité mais le coût de calcul augmente par rapport à OSNNCPD1. Le dernier algorithme (ROSNNCPD) est basé sur une réécriture de l'algorithme OSNNCPD1 de manière récursive afin de réduire le temps d'exécution. De plus, il prend en compte tous les sous-tenseurs passés dans sa fonction de coût d'optimisation. Nous proposons aussi une extension de l'algorithme 2 (OOSCPD) en imposant à la fonction de coût une contrainte d'orthogonalité afin d'avoir un gain de temps durant le processus d'optimisation. Nos algorithmes présentent des avantages par rapport aux algorithmes de la littérature :

- La prise en compte des matrices facteurs obtenues à l'instant précédent afin de mettre à jour les matrices facteurs à l'instant  $t$ . Cela permet un gain dans la rapidité de convergence et cela évite de stocker en mémoire toutes les matrices facteurs ;
- La prise en compte de la variation du rang au cours du temps ;
- La flexibilité des algorithmes à incorporer d'autres contraintes.

D'un point de vue pratique, une simulation de données synthétique de fluorescence a été réalisée avec le phénomène de variation de rang afin de vérifier la convergence de nos algorithmes et de calibrer leurs différents hyperparamètres. Après une évaluation et une comparaison des algorithmes sur des données de fluorescence numérique simulées, nous avons réalisé en laboratoire une expérimentation dans le but d'acquérir des données réelles de fluorescence dans des conditions contrôlées. Pour cela, nous avons réalisé des séries d'injections de 4 fluorophores dilués de concentrations connues (Fluorescéine (F), Tryptophane (TRP), 5-Sulfate-8-HydroxyQuinine (5S8HQ) et Rhodamine (RHO)) à des intervalles de temps réguliers et contrôlés afin de simuler l'apparition et la disparition de certains fluorophores au cours du temps. Nous avons réalisé l'expérimentation de façon à intégrer le phénomène de la variation de rang. Deux jeux de données sont tirées de cette expérience, le premier jeu de donnée est un tenseur de petite taille. L'intérêt ici c'est de voir si sur un petit jeu de donnée nos algorithmes convergent bien comme dans le cas de la simulation numérique. Le second jeu de donnée correspond naturellement à un tenseur de grande taille. Avec ces deux jeux de données, une comparaison a été réalisée avec des algorithmes de CPD en ligne et des algorithmes de CPD hors ligne pouvant être adapté au cas en ligne avec différents critères. Nos algorithmes donnent les meilleurs résultats dans les deux situations. Nous retenons qu'en spectroscopie de fluorescence, l'erreur moyenne sur les facteurs estimés est un critère important qui permet d'améliorer l'identification des fluorophores. Concernant ce critère également, notre approche donne généralement de meilleurs résultats que les autres stratégies en ligne. Plus précisément, lorsqu'un chevauchement entre sous-tenseurs consécutifs est autorisé, OSNNCPD1 et ROSNNCPD fournissent clairement les meilleurs résultats. Dans ce cas, les facteurs sont également estimés avec plus de précision et les algorithmes sont plus stables. A l'inverse, dans le cas sans chevauchement, OSNNCPD2 apparaît comme la meilleure option (mais il est plus judicieux de surestimer le rang). Finalement, dans la plupart des situations, ROSNNCPD fournit des résultats légèrement meilleurs que OSNNCPD1. C'est aussi l'algorithme le plus rapide. Nous recommandons donc de combiner chevauchement des sous-tenseurs et ROSNNCPD.

Pour finir, une expérimentation similaire a été réalisée pour une évaluation des algorithmes *on field* à Endoume dans la ville de Marseille. Cette expérimentation nous a permis de mettre en œuvre nos algorithmes sur des données réelles dont nous ne maîtrisons pas les facteurs comme ce fut le cas

de l'expérimentation en laboratoire ou des données simulées. Notons que cette expérimentation a été difficile à mettre en place et aussi que le traitement des données est difficile par rapport aux données en laboratoire et aux données simulées. Cela s'explique par le fait que nous n'avons connaissance d'informations a priori et aussi que nous avons observé des perturbations lors de l'acquisition des matrices de fluorescence comme l'apparition de bulles d'air dans le volume de mesure. En dépit de ces difficultés, nos algorithmes arrivent à bien converger et sont capables d'estimer des MEEFs de composants fortement altérées lors de l'acquisition.

Enfin, même si nos algorithmes sont présentés dans le cas particulier de la NN-CPD en ligne des tenseurs de fluorescence du troisième ordre, ils ne se limitent pas à ce domaine d'application et peuvent être facilement étendus à des tenseurs d'ordre supérieur. Les trois algorithmes proposés permettent de suivre correctement les variations de rang dans la plupart des situations considérées contrairement aux approches de référence. Les travaux effectués nous donnent de bons résultats mais peuvent probablement être améliorés.

## 5.5 Perspectives

Nous proposons ici deux pistes d'amélioration pour les approches proposées.

**Prise en compte des phénomènes pouvant altérer le modèle trilinéaire** Une perspective est de prendre en compte les effets non-linéaires liés à l'acquisition des données de fluorescence et à la nature des échantillons dans les différents algorithmes en ligne proposés. Cela permettra de supprimer l'étape du prétraitement des données de fluorescence. Il est en effet envisageable de supprimer les diffusions de Rayleigh et Raman de façon automatique en utilisant des méthodes de deep learning, ces dernières pouvant s'intégrer dans l'algorithme de décomposition en ligne. Le deep learning est une méthode qui a du succès depuis une dizaine d'année et donne des résultats prometteurs en dépit de sa complexité et de la grande quantité de données nécessaires. Une des techniques du deep learning est l'Auto-Encodeur (AE) qui est utilisée dans le débruitage, l'augmentation des données, le compression... L'idée serait d'utiliser cette technique afin de supprimer les diffusions Rayleigh et Raman de façon automatique et rapide. Des premiers travaux ont été réalisés et donnent des résultats prometteurs (Cf Annexe B). D'autres tests sont en cours pour confirmer ces premiers résultats.

On peut aussi remplacer la CPD par la NLFD dans le modèle pour tenir compte des effets d'écran

**Extensions des principaux algorithmes** Une extension naturelle de nos approches est d'incorporer une information a priori à nos dictionnaires et cela reviendrait à traiter un problème d'apprentissage supervisé. Une autre extension est de réécrire de façon récursive l'algorithme OSNNCPD2 ainsi que OOSCPD (Cf chapitre 3) afin d'avoir un gain de temps lors du processus d'optimisation. Étant donné que le terme du coefficient de pénalité joue un rôle principal dans l'estimation du rang, il est possible d'envisager l'ajout dans les travaux futurs de la méthode de  $L - curve$  pour le choisir automatiquement. On peut aussi essayer de modifier des algorithmes basés sur la diagonalisation conjointe pour faire de la CPD en ligne rapide.

D'autres investigations devront être programmées pour tester d'autres mélanges contrôlés et d'autres pourcentages et stratégies de chevauchement de données, mais également pour implémenter directement dans un système de surveillance ou de monitoring les algorithmes de décomposition en ligne. A long terme, ces résultats encourageants permettent d'envisager un suivi en milieu naturel couplé avec d'autres données comme les paramètres physico-chimiques, voire même les informations de conditions météorologiques dans une modèle de plus large échelle, pour une meilleure détection ou prévision des contaminations chimique et biologique.

# Annexe A

## Détection des contaminants biologiques et Chimiques

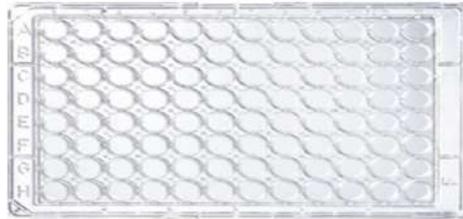
### A.1 Détection des contaminants biologiques : Cas du *Escherichia coli*

*Escherichia coli*, également connue sous le nom d'*E. coli*, est une bactérie intestinale très commune chez l'Homme. *E. coli* est un excellent indicateur de contamination fécale. En effet, cette bactérie n'est pas adaptée au milieu marin et est donc beaucoup moins résistante à l'environnement que de nombreux germes pathogènes et même d'autres "indicateurs de pollution fécale" tels que les streptocoques fécaux. Leur présence dans l'eau ou la nourriture témoigne donc du risque de propagation de germes transmissibles dans les selles, tels que les salmonelles. La consommation d'eau ou d'aliments contenant de grandes quantités d'*E. coli* peut présenter un danger pour l'Homme. Ces bactéries sont le plus souvent liées à l'apparition d'infection ou d'intoxication dans les eaux de baignade en eaux douces ou marines.

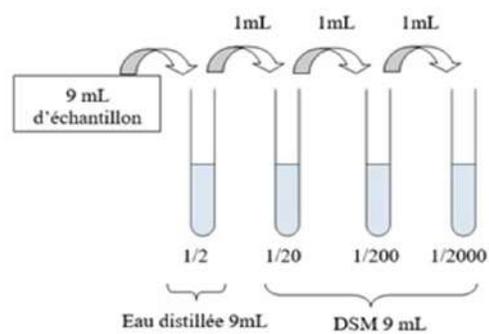
La technique de détection et de quantification des bactéries *E. coli* en biologie est réalisée en échantillonnant l'environnement pour fabriquer les microplaques (Figure A.1). Une dilution en cascade est effectuée dans un environnement stérile, de manière à obtenir un intervalle de dilution compris entre 1/2 et 1/20 pour les eaux. Un exemple est donné dans la Figure A.2. Cependant, les tests permettant de détecter les micro-organismes pathogènes sont longs (24 à 72 heures) et la qualité de l'eau peut alors changer radicalement au cours de cette période. C'est pourquoi les modèles statistiques prédictifs, basés sur un ensemble de données et utilisant également des données environnementales, peuvent désormais fournir des prévisions en temps réel sur les contaminations bactériennes. Aujourd'hui, des modèles prédictifs sont utilisés à Hong Kong, en Écosse et dans les grands lacs d'Amérique du Nord. Ces nouvelles méthodes sont plus rapides en évitant les contraintes des méthodes conventionnelles. Elles sont prometteuses en tant qu'outils de gestion pour les aquaculteurs.

#### A.1.1 Quantification d'*E. coli* dans les eaux aquacoles

Afin de faire les dilutions, un diluant spécial microplaques (DSM) a été réalisé à 22, 5g/L. Un volume de 9mL d'eau à analyser est placé dans un tube stérile contenant 9mL d'eau distillée stérile (dilution 1/2). Une dilution en cascade est effectuée dans du milieu DSM stérile ensuite, de manière à obtenir une gamme de dilution adaptée aux eaux de baignade.



**FIGURE A.1 – microplaques.**



**FIGURE A.2 – Dilution en cascade.**



**FIGURE A.3 – Ferme aquacole**

Les microplaques sont ensuite recouvertes d'un adhésif prévu à cet effet afin d'éviter l'évaporation et incubées au minimum 36 heures à 44°C. Les microplaques utilisées sont prêtes à l'emploi et stériles. Le milieu de culture déshydraté est fixé au fond de la microplaque. Il est réhydraté par l'échantillon d'eau à analyser. Un tube de dilution est vidé dans une boîte de Pétri et 200  $\mu\text{L}$  de la dilution sont répartis dans chaque puits à l'aide d'une pipette multicanaux. En présence de l'enzyme, le substrat libère un composé fluorescent pouvant être observé à l'aide d'une lampe *UV* à 366 *nm*. Le nombre de puits fluorescents est compté pour chaque dilution. A partir du nombre caractéristique obtenu, une analyse statistique basée sur la loi de Poisson permet de dénombrer les entérocoques dans l'eau analysée.

### **A.1.2 Prédiction des *E. coli* chez BIOCEANOR**

Tournée vers des innovations technologiques de pointe utilisant le Machine Learning et le big-data, Bioceanor a développé des services innovants pour la prédiction des paramètres critiques de la qualité de l'eau de nos océans ([https://www.francetvinfo.fr/france/ilsontla-solution/a-l-embouchure-du-var-des-bouees-connectees-pour-mesurer-la-pollution-des-eaux\\_4710677.html](https://www.francetvinfo.fr/france/ilsontla-solution/a-l-embouchure-du-var-des-bouees-connectees-pour-mesurer-la-pollution-des-eaux_4710677.html)). Bioceanor propose la première solution permettant de prévoir 48 heures à l'avance la teneur en oxygène dissous. Une bonne maîtrise de la teneur en oxygène est vitale pour les organismes aquatiques et pour la bonne stabilité des écosystèmes. Aujourd'hui, Bioceanor propose principalement ses solutions au secteur de l'aquaculture. Ce secteur est en pleine croissance avec l'augmentation de la pression des quotas sur la pêche de capture et l'augmentation exponentielle de la consommation des poissons et autres coquillages issus des océans. Les solutions proposées par Bioceanor permettent à l'aquaculture (Figure A.3) de produire de façon plus durable et responsable. Bioceanor s'engage également sur des projets environnementaux. Ces

solutions sont déployées sur des zones de baignades pour la surveillance et la prévention de fermeture de celles-ci en raison de l'état sanitaire du plan d'eau ; sur des lacs américains pour la surveillance des phénomènes anoxiques venant perturber les pompages réalisés pour la consommation humaine. . . La surveillance continue de la qualité de l'eau est basée sur l'intégration de l'IoT (Internet of Things, internet des objets) dans l'aquaculture. Il permet d'établir un état de la situation en temps réel dans les fermes. Généralement, les mesures sont sporadiques et elles ne reflètent pas ce qui se passe exactement dans les fermes. Des mesures continues permettent d'avoir une vision globale de la ferme et de pouvoir agir si nécessaire. Cette partie comprend le développement de sondes capables de mesurer un large éventail de paramètres physico-chimiques à moindre coût, faciles à utiliser et avec une capacité de mesure et de transmission à très haute fréquence.

Bioceanor est impliquée dans de nombreux projets d'innovation et de recherche au niveau national et européen. Depuis sa création en 2018, Bioceanor a développé 4 projets structurants pour la société :

- SENSITHAU (Avenir Littoral Région Occitanie ; <https://www.bioceanor.com/fr/lancement-du-projet-sensithau> ; 2020 - 2022). Ce projet soutenu par la région Occitanie est porté par Bioceanor. L'objectif du projet SENSITHAU est de mettre en place un réseau de surveillance sur la lagune Thau en opérant un réseau de capteurs et en mettant en place des services de prédictions des anoxies. Le second objectif est de coupler ces données à des données externes (telles que les données satellitaires) pour permettre d'anticiper les blooms de microalgues.
- iFishIENCi (H2020 ; <http://ifishienci.eu/> ; 2018-2022). Ce projet d'innovation, piloté par la société Aquabiotech vise à mettre en place des innovations pour l'aquaculture en recirculation. L'objectif principal de ce projet est de permettre une alimentation plus efficace et plus durable des animaux élevés dans ces systèmes via l'apport de technologies couplant l'internet des objets et le big-data.
- ASTRAL (H2020 ; [www.astral-project.eu](http://www.astral-project.eu) ; 2021- 2025). Ce projet de recherche, piloté par l'institut NORCE en Norvège est composé de 16 partenaires. Le projet dans son ensemble vise à apporter de nouvelles technologies dans le secteur de l'aquaculture multi-trophique. Bioceanor est responsable de la mise en place de solutions innovantes de captation de la donnée physico-chimique. Aussi, Bioceanor collabore sur la prévision des événements critiques tels que les blooms de microalgues et les contaminations par des pathogènes.
- I-LAB (ministère de l'enseignement supérieur et de la recherche ; 2019-2022). Ce projet d'innovation, monopartenaire est porté par Bioceanor et financé par le ministère de l'enseignement supérieur et de la recherche. Ce projet récompense la création de projet de société pour des personnes ayant exercé auparavant des activités dans le monde de la recherche. Ce projet a été remporté pour bâtir le service de prédiction de l'oxygène dissous de Bioceanor. A travers ce projet nous avons pu déployer des réseaux de capteurs pour construire notre base de données. Ce pool de données a été ensuite analysé en interne et nous a permis d'établir notre service de prédiction à 48 heures de l'oxygène dissous.

Outre l'aquaculture, d'autres applications sont possibles avec la technologie développée par Bioceanor, telles que :

- Surveillance des plages, des lacs et des rivières ;
- Surveillance de l'environnement et des ports ;
- Surveillance des fermes de spiruline et des aquaponies (<https://aquaeas.org/Program/PaperDetail/38955>).

Etant donné que le temps réel est nécessaire pour obtenir des informations directement de la ferme, la prédiction permettra d'agir pour éviter tout problème avant qu'il n'apparaisse. L'évolution d'un paramètre pourrait être corrélée à d'autres paramètres exogènes. Grâce aux mesures en temps



FIGURE A.4 – Image du capteur de Biocenor *in situ*

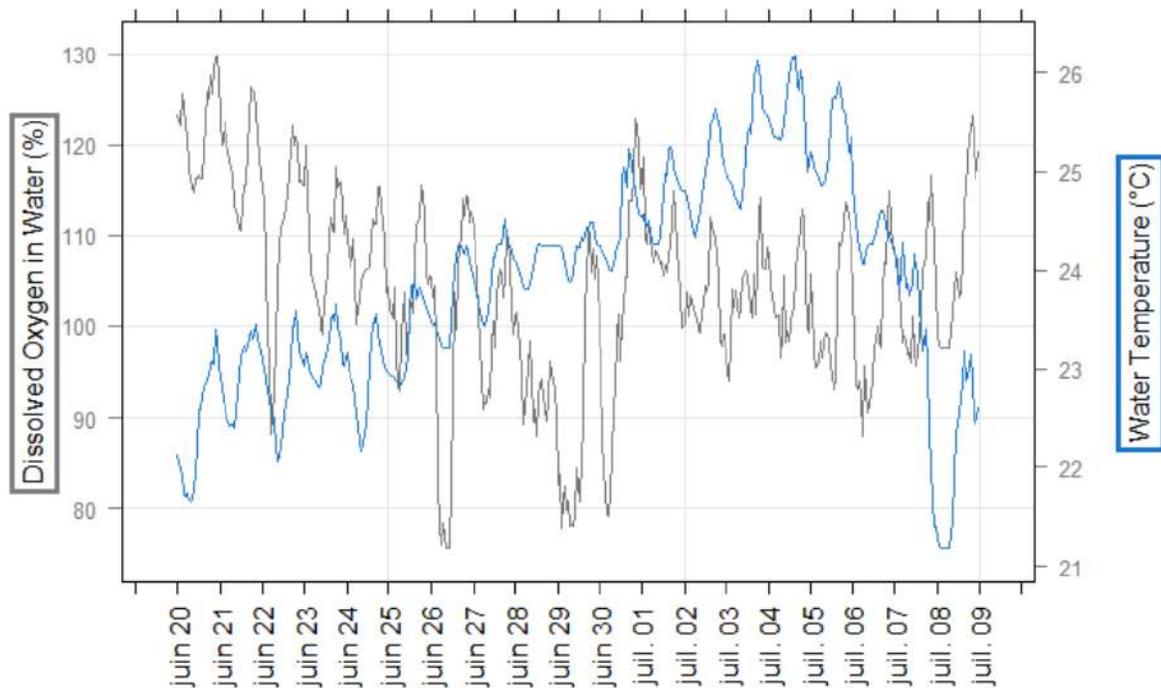


FIGURE A.5 – Tracé en gris : évolution sur 20 jours d’oxygène dissous dans l’eau (%). Tracé en bleu température de l’eau (° C).

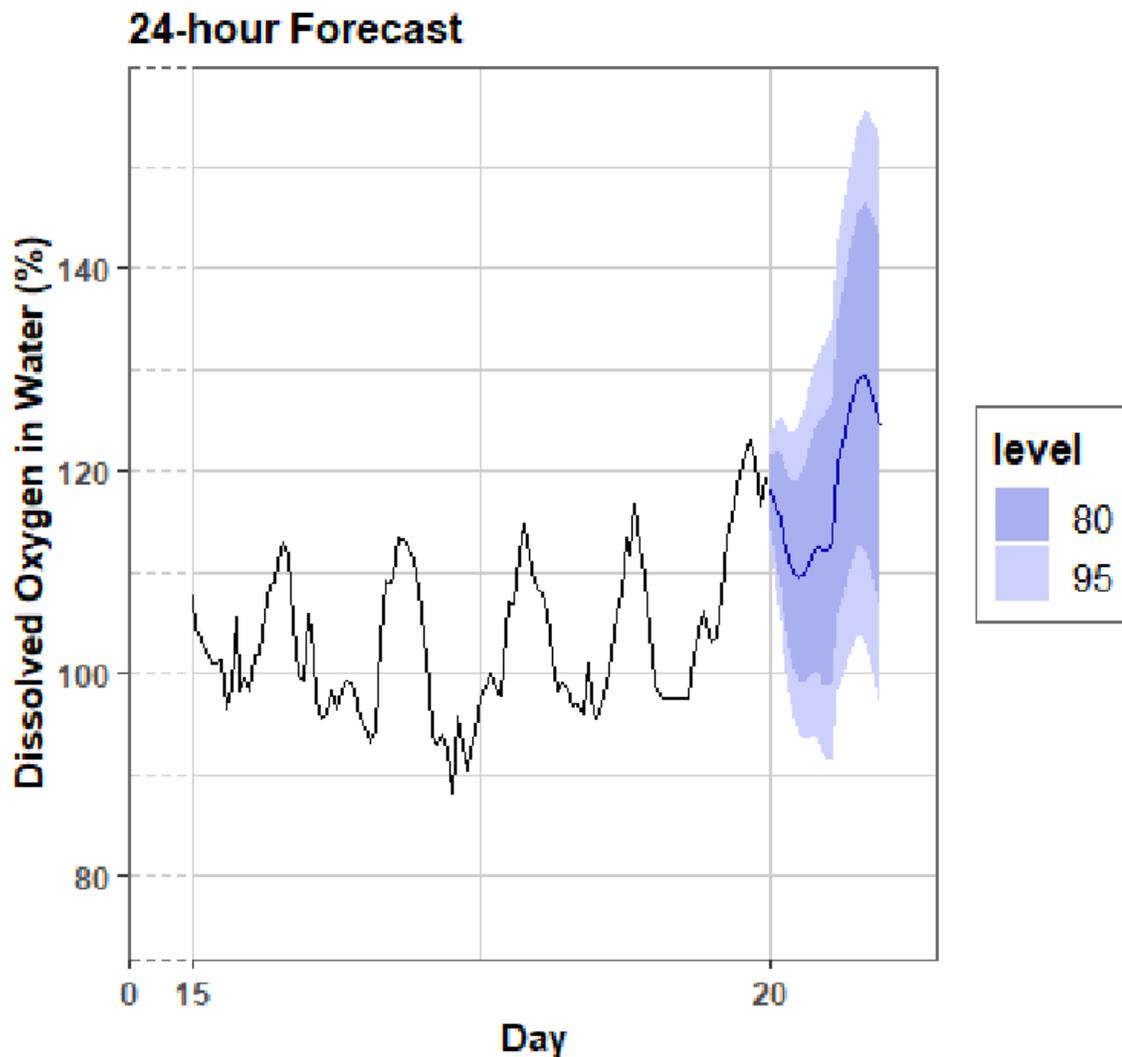


FIGURE A.6 – Valeurs prédites en 24 heures (ligne bleue) pour le 20e jour, basées sur des données collectées (ligne noire), avec 80% (zone remplie de bleu profond) et 95% (zone remplie de bleu clair).

réel, couplées à des données météorologiques et satellites, Bioceanor a étudié plusieurs méthodes (outils graphiques, tests de corrélation, régressions et plus) pour estimer les paramètres les plus importants expliquant le niveau d'oxygène dissous dans l'eau. Dans l'exemple donné dans la Figure A.5, les données proviennent d'une pisciculture (cage maritime) en mer Méditerranée (France). Nous avons vu que la température de l'eau est l'une des caractéristiques les plus importantes affectant les niveaux d'oxygène dissous dans cet essai (Figure A.5). Dans les systèmes d'aquaculture, de nombreux facteurs peuvent affecter les niveaux d'oxygène dissous et dépendent directement du type d'aquaculture. Ces facteurs dépendent du type de production, mais la plupart d'entre eux sont mesurables et pourraient être intégrés dans un modèle de prédiction basé sur l'IoT. Ces facteurs sont, par exemple, la qualité nutritionnelle des aliments, la composition chimique, la durée et le niveau d'alimentation des poissons dans n'importe quel système...

De plus, la modélisation prédictive peut informer un jour plus tôt les valeurs de la concentration en oxygène dissous avec des intervalles de prédiction de 80% et 95% (Figure A.6), permettant aux aquaculteurs de prendre des mesures (par exemple, adapter l'alimentation ou activer les aérateurs). Ceci est important, car les agriculteurs peuvent perdre toute leur production avec une mauvaise

gestion de l'oxygène. Prenons un exemple d'étude réalisé par Bioceanor qui consiste à mesurer en continu l'oxygène dissous sur trois semaines et à agréger les données au niveau horaire. Les concentrations d'oxygène dissous ont été analysées comme une série temporelle avec une saisonnalité quotidienne. Un modèle ARIMA (Autoregressive integrated moving average) optimal basé sur la valeur du critère d'information Akaike (en anglais Akaike information criterion ou AIC) a été ajusté aux données de prévision. Néanmoins, une approche de la série temporelle unique peut conduire à une mauvaise stabilité, et donc, ne suffit pas pour anticiper toute la dynamique en temps réel. La difficulté est que les processus prédictifs peuvent être spécifiques à un paramètre de qualité cible avec sa dynamique temporelle ou collective en établissant une relation entre plusieurs prédicteurs et la variable d'intérêt. Cela implique en grande partie à la fois des modèles d'apprentissage automatique (comme l'algorithme des forêts aléatoires, des machines à support de vecteurs (SVM) ou des réseaux de neurones), des prévisions de séries chronologiques (telles que des modèles de lissage exponentiels, d'autorégression vectoriel...) et, lorsque cela est nécessaire, combiner ces approches pour améliorer la qualité des prévisions. Dans les étapes suivantes, l'utilisation de méthodes sans hypothèse linéaire forte comme les réseaux de neurones, robustes au bruit, et avec des facteurs affectant l'oxygène dissous comme entrée, pourrait être pertinente [178].

Alors que la teneur en oxygène dissous est en continu mesurable et prévisible, certains paramètres restent impossibles à mesurer en temps réel dans l'eau et donc, sont plus difficiles à prévoir. C'est le cas, par exemple, de la contamination par des micro-organismes comme les bactéries. Certaines bactéries sont directement responsables des maladies dans les fermes tandis que d'autres rendent le poisson inapte à la consommation humaine comme les bactéries *Escherichia coli*. Bien que l'identification d'une contamination soit soumise à un long processus (plusieurs jours), Bioceanor a développé des modèles statistiques pour identifier les facteurs de qualité clés responsables des contaminations d'*E. Coli* enregistrées dans les fermes. En intégrant ces informations, Bioceanor sera bientôt en mesure de proposer des modèles pour prédire les risques de contamination d'*E. Coli* et par la suite pour de nombreux autres micro-organismes.



## **Annexe B**

# **Suppression des phénomènes de diffusion en ligne des matrices de spectroscopie de fluorescence en utilisant l'apprentissage profond**

Le prétraitement des données provenant d'un capteur est une étape importante. Cette étape permet l'élimination de différents bruits, ou de réorganiser les données. Dans la spectroscopie de fluorescence, après l'acquisition des matrices d'excitation et d'émission (MEEFs), le prétraitement est l'étape importante avant d'appliquer la CPD car elle permet d'éliminer les phénomènes de diffusion (Raman et Rayleigh).

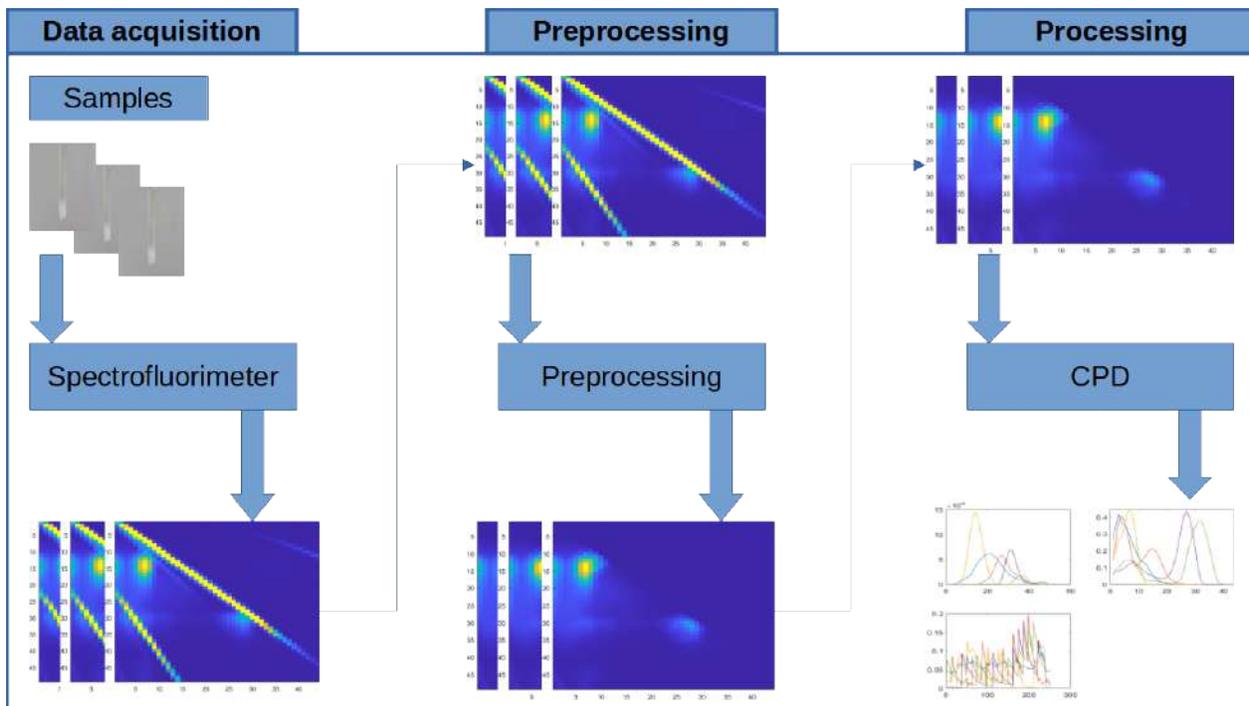


FIGURE B.1 – Les différentes étapes de surveillance d'un environnement aquatique.

## B.1 Introduction

La spectroscopie de fluorescence est un outil de surveillance et de mesure de plus en plus utilisé dans l'étude de l'environnement et de la ressource aquatique naturelle [18, 92, 93, 2].

Les données de la spectroscopie de fluorescence sont des matrices d'acquisition appelées Matrices d'Emission et d'Excitation de fluorescence (MEEFs). Cette matrice résultant de la spectroscopie de fluorescence permet de déterminer les contributions de molécules organiques dans un milieu liquide complexe. Les différentes étapes de surveillance d'un environnement aquatique est résumée dans la Figure B.1. Pour produire la MEEF d'un échantillon, une longueur d'onde d'excitation  $\lambda_{ex}$  est sélectionnée par le premier monochromateur du spectrofluorimètre avant de balayer toutes les longueurs d'onde d'émission  $\lambda_{em}$  à l'aide du second monochromateur (Figure 2.11), ce qui permet d'obtenir un spectre d'émission : l'ensemble des spectres d'émission associés aux différentes longueurs d'onde d'excitation correspond à une MEEF dont un exemple est représenté par la Figure B.2. Un tenseur de fluorescence (Figure 2.13) correspond au regroupement de plusieurs MEEF.

Cette étude est axée sur le prétraitement des MEEFs. Cette étape est très importante car un mauvais prétraitement des matrices de fluorescence peut se produire des erreurs dans la dernière étape (étape de la CPD).

Comme cela a été évoqué précédemment, chaque MEEF contient une superposition de signaux de fluorescence et de pics de diffusion Rayleigh et Raman (Figure B.4). Les réseaux de diffraction qui composent les monochromateurs des spectrofluorimètres produisent des spectres d'ordres plus ou moins élevés. Habituellement, on peut observer sur les MEEFs mesurées les spectres de fluorescence et les pics de diffusion pour l'ordre un mais aussi les spectres et les pics de diffusion pour les ordres plus élevés. La présence inévitable des pics de diffusion engendre une double difficulté de traitement

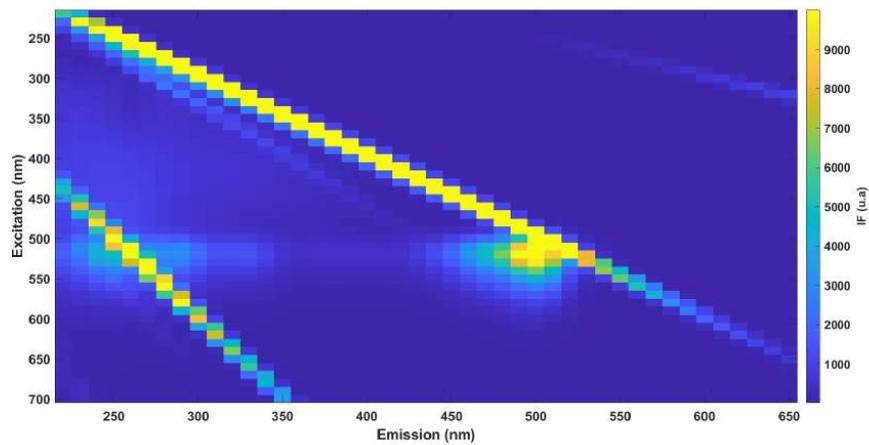


FIGURE B.2 – MEEF avec le phénomène de diffusion.

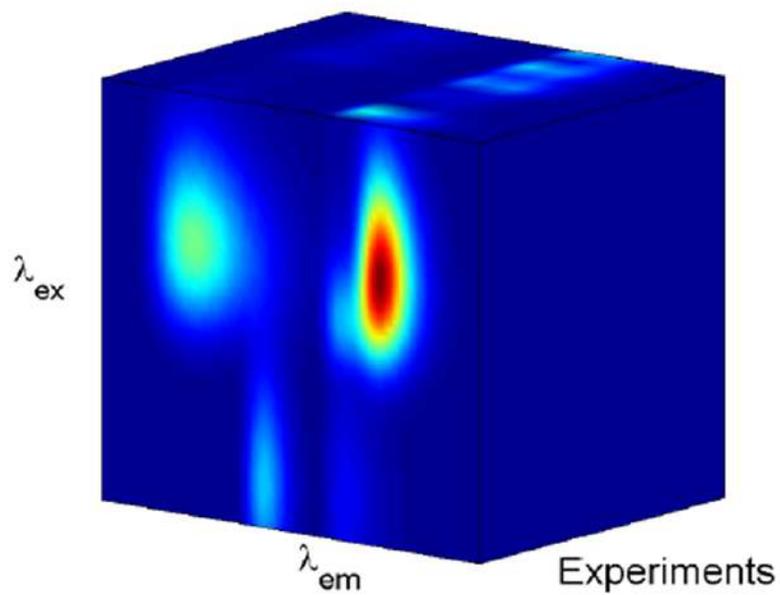
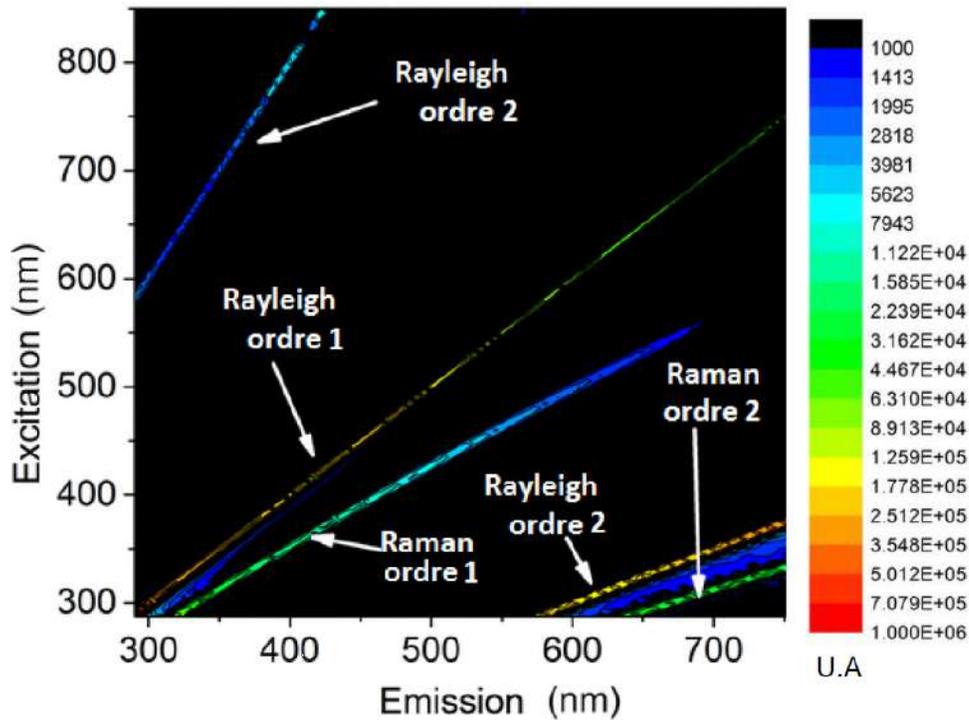


FIGURE B.3 – Exemple de tenseur de troisième ordre. Intensité de fluorescence fonction de 3 variables : longueurs d'onde d'excitation et d'émission et des échantillons.



**FIGURE B.4 – MEEF illustrant l’emplacement des diffusions de Raman et Rayleigh dans un échantillon d’eau ultra pure (Image tirée de [2]).**

numérique ultérieur. Premièrement, ils peuvent masquer en partie un spectre de fluorescence d’intérêt et donc rendre plus délicate l’identification d’un fluorophore. Deuxièmement, les pics de diffusion Rayleigh et Raman ne suivent pas le modèle trilinéaire de l’équation 2.29.

## B.2 Travaux connexes

Il est donc nécessaire de soigneusement éliminer les pics de diffusions numériquement avant toute décomposition de tenseur de fluorescence. Il existe plusieurs méthodes possibles pour supprimer les pics de diffusion mais la plus utilisée est la méthode de Zepp [126] qui est basée sur l'identification des régions des différents pics de diffusions et de l'interpolation bilinéaire de chaque MEEF sur ces régions en utilisant la triangulation de Delaunay. Une autre méthode utilise une technique de filtrage des images basée sur la morphologie mathématique [93]. Une décomposition alternative à la CPD prenant en compte ces effets a été récemment proposée [127]. Cependant l'algorithme associé est assez lent et n'a pas été testé sur ce type de donnée. De même on ignore son comportement lorsque le nombre de composés fluorescents est mal estimé.

### B.2.1 Méthode de Zepp

La méthode de Zepp est la plus utilisée dans la spectroscopie de fluorescence. Elle peut se résumer comme ceci :

- La première étape consiste à modéliser l'emplacement et le comportement des raies de diffusion. Les positions de pics Rayleigh suivent une loi du premier degré et que celles de pics Raman sont estimées à partir d'un polynôme du second degré (Figure B.4).
- Dans un deuxième temps, on élimine tous les pixels présents aux emplacements marqués.
- Pour finir, on reconstruit par interpolation (triangulation de Delaunay) les pixels manquants.

Elle donne de bons résultats dans la majorité des cas mais elle peut parfois s'avérer destructrice en allant jusqu'à supprimer un pic de fluorescence si celui-ci se trouve dans la zone des raies de diffusion.

### B.2.2 Méthode de filtrage par morphologie mathématique

Une autre méthode utilise une technique de filtrage des images basée sur la morphologie mathématique. La morphologie mathématique fut introduite dans [179, 180]. Le principe de la morphologie mathématique est d'utiliser une forme géométrique appelée élément structurant qui va servir de support pour analyser l'ensemble qu'on souhaite étudier. Cet élément structurant est défini par un masque binaire et un entre. Dans [93], les auteurs proposent :

- La première étape consiste à utiliser un élément structurant de type ligne, (approximativement) orthogonal aux raies de diffusion, ces dernières seront supprimées, alors que le reste de l'image ne sera pas trop altéré. L'effet positif étant surtout qu'un pic de fluorescence superposé aux raies de diffusion ne sera pas éliminé.
- Un filtre passe-bas (filtre moyenneur) est utilisé sur les images corrigées afin de lisser les résultats.

Le nettoyage des raies de diffusion par morphologie mathématique supprime toutefois un peu moins bien les raies de diffusion dans les zones où le signal est absent. Pour parer à ce problème, les auteurs proposent une deuxième approche qui consiste à combiner l'approche morphologique mathématique et la méthode de Zepp de façon pondérée. Si on appelle  $I_{Zepp}$  l'image corrigée par l'algorithme de Zepp,  $IM_{morphologie}$  l'image corrigée par morphologie mathématique, l'image résultante peut être définie par :

$$IM_{corrige} = \alpha I_{Zepp} + (1 - \alpha) IM_{morphologie} \quad (\text{B.1})$$

où  $\alpha$  est un coefficient compris entre 0 et 1.

### B.3 Suppression des phénomènes de diffusion en utilisant les autoencodeurs

$$E = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p (X_{ij} - X'_{ij}) \quad (\text{B.2})$$

Les autoencodeurs (AE) sont des réseaux de neurones couramment utilisés pour la sélection et l'extraction de caractéristiques. Il est la fusion de deux blocs, la partie encodeur et la partie décodeur. Cependant, lorsqu'il y a plus de nœuds dans la couche cachée qu'il n'y a d'entrées, le réseau risque d'apprendre ce qu'on appelle la "fonction d'identité", également appelée "fonction nulle", ce qui signifie que la sortie est égale à l'entrée, ce qui rend l'autoencodeur inutile. Le principe d'autoencodeur a été généralisé à des tâches plus variées, en particulier dans les tâches de débruitage (on parle de denoising AE en anglais). Après avoir appris comment représenter de manière optimale les entrées, en fournissant à l'autoencodeur des entrées bruitées, il est capable de donner en sortie des données débruitées qui sont très proches de l'entrée originale. Hormis le débruitage, les autoencodeurs, ont aussi très utilisés pour la création de nouvelles données. Dans le cas de la création de nouvelles données, la partie décodeur est utilisée pour créer ces données à partir de vecteurs de l'espace latent. L'intérêt dans ce cas, est l'augmentation des données ou la "data augmentation" en anglais et cela permet d'entraîner d'autres modèles. Les autoencodeurs sont aussi utilisés pour la compression de données, la détection d'anomalie, etc.

Les autoencodeurs de débruitage résolvent ce problème en corrompant volontairement les données d'entrée, en ajoutant du bruit ou en masquant certaines des valeurs d'entrée. Comme l'indique la Figure B.5, en spectroscopie de fluorescence,  $\mathbf{X}$  correspond à une MEEF non traitée et  $\mathbf{X}'$  correspond à la MEEF traitée (équation B.2). L'idée alors est d'utiliser les autoencodeurs de débruitage pour supprimer les phénomènes de diffusions que l'on considère comme des bruits.

#### B.3.1 Architecture ou réseau utilisé

Nous utilisons une architecture (réseau) classique. Dans la littérature on appelle souvent l'architecture, le modèle ou encore réseau de neurones. Elle n'est pas fixe ni unique elle varie en fonction du problème, de l'objectif voulu. Notre architecture est composée d'une première couche de convolution 2D (conv2D) de 128 filtres de taille (3,3) suivie d'une deuxième couche convolution 2D (conv2D) de 64 filtres de taille (3,3). Ensuite nous ajoutons une troisième couche de convolution 2D (conv2D) de 32 filtres de taille (3,3) et afin une dernière couche de convolution 2D (conv2D) de 16 filtres de taille (3,3). Cet ensemble forme notre encodeur et pour le décodeur, nous utilisons une première couche de la transposée de la convolution 2D (Conv2DTr) de 16 filtres de taille (3,3), une deuxième couche de la transposée de convolution 2D (Conv2DTr) de 32 filtres de taille (3,3). Ensuite une troisième couche de la transposée de convolution 2D (Conv2DTr) de 64 filtres de taille (3,3) et afin une dernière couche de la transposée de convolution 2D (Conv2DTr) de 128 filtres de taille (3,3). Avant cette dernière couche nous ajoutons une couche de "Dropout" pour éviter l'attache aux données d'entraînement. L'architecture est décrite sur l'image B.6 et on remarque que le rôle de l'encodeur est de réduire l'image d'entrée pour ne retenir que les caractéristiques importantes qui lui permettent ensuite de reconstruire l'image.

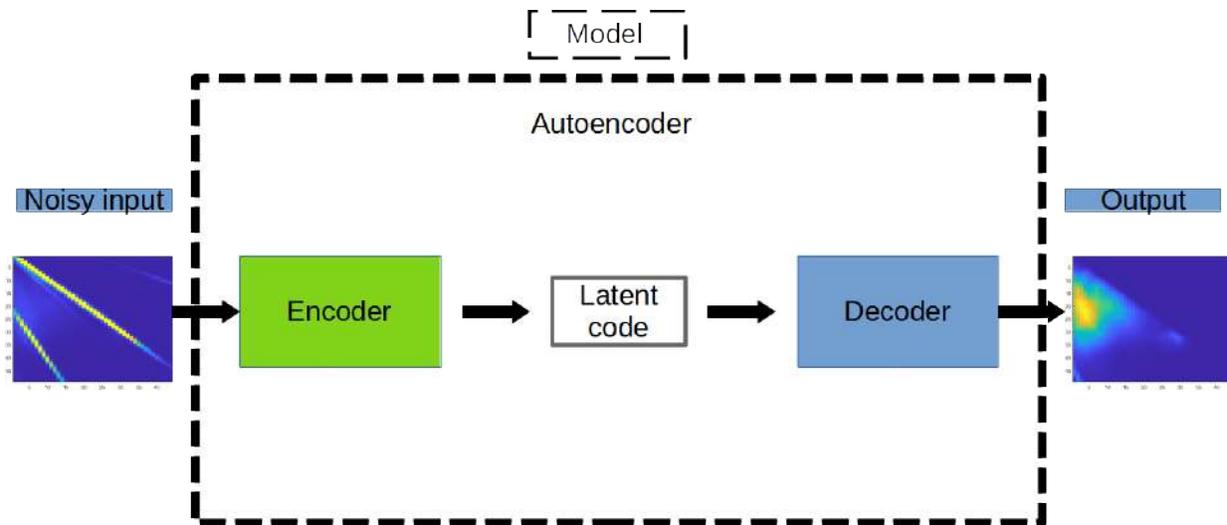


FIGURE B.5 – Illustration d’un autoencodeur avec  $X$  l’entrée et  $X'$  la sortie.

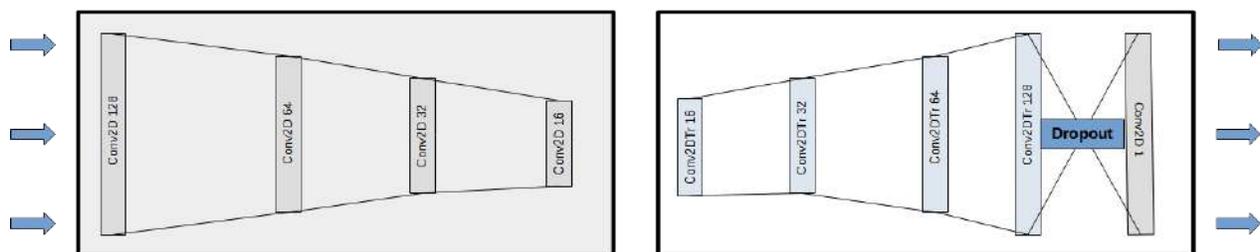


FIGURE B.6 – Illustration de notre architecture utilisé pour l’entraînement des données.

**Conv2D** Une couche de convolution est constituée de plusieurs filtres (ou noyaux) de convolution  $\omega_i$  appliqués à une matrice d'entrée (une image)  $\mathbf{F}$  produisant des images filtrées  $\mathbf{G}_i$  :

$$\mathbf{G}_i(x, y) = \omega_i \otimes \mathbf{F} = \sum_{\delta x=-k_i}^{k_i} \sum_{\delta y=-k_j}^{k_j} \omega(\delta x, \delta y) \mathbf{F}(x - \delta x, y - \delta y) \quad (\text{B.3})$$

**Conv2DTr** La nécessité de la transposée de la convolution résulte généralement du désir d'utiliser une transformation allant dans la direction opposée d'une convolution normale mais reste néanmoins une convolution. On ne fait pas exactement une transposée lors du processus, il s'agit d'un nom d'emprunt. Cette opération revient à faire une convolution normale après avoir rempli une partie de la matrice d'entrée avec des 0. En somme elle peut être vu comme une autre façon de faire une interpolation bilinéaire.

**Dropout** Le dropout est une technique de régularisation pour réduire le surapprentissage. La couche de dropout a été introduite par [181]. Cette couche est utilisée pendant l'apprentissage. Elle permet de désactiver aléatoirement des neurones durant les différentes itérations de l'apprentissage.

### B.3.2 Préparation des Données d'entraînement

Nous avons simulé un tenseur  $\mathcal{T}(80,80,50000)$  contenant de la diffusion sur chaque MEEF et un deuxième tenseur  $\mathcal{Y}(80,80,50000)$  sans diffusion grâce au programme fourni par le laboratoire MIO.  $\mathcal{T}$  représente nos données d'entraînement et  $\mathcal{Y}$  représente le label (étiquette) de  $\mathcal{T}$ . Cela signifie que si nous prenons une MEEF de  $\mathcal{T}$ , cette dernière aura de la diffusion mais cette même MEEF prise dans  $\mathcal{Y}$  sera sans diffusion. Le but est de donner le tenseur  $\mathcal{T}$  comme entrée à l'autoencodeur en lui signifiant que  $\mathcal{Y}$  est la reconstruction c'est-à-dire la vérité terrain (tenseur de référence).

L'entraînement de notre modèle se fait en deux phases comme l'indique la Figure B.7 :

- Training phase ou phase d'entraînement : La phase d'entraînement consiste après préparation des données de procéder à l'entraînement de notre réseau en choisissant les bons paramètres comme le choix de la fonction d'optimisation dans notre cas nous choisissons Adam (Cf Chapitre 3). Une fois l'entraînement des données terminé, nous sauvegardons les poids du modèle.
- Test phase ou phase de test : Nous préparons un troisième tenseur de même taille que le tenseur d'entraînement contenant aussi les phénomènes de diffusions appelé tenseur de test que le réseau ne connaît pas. Nous chargeons les poids sauvegardés et nous prédisons le nouveau tenseur contenant les MEEFs sans diffusions comme l'indique l'équation B.2.

Après la phase de test, le réseau de neurones peut être utilisé pour un tenseur quelconque de fluorescence.

### B.3.3 Résultats et discussions

Pour cette expérimentation nous avons un tenseur  $\mathcal{T}(44,49,252)$  obtenu en laboratoire (Cf chapitre 4) contenant 5 composants. Pour la prédiction, nous utilisons les poids de notre modèle d'entraînement

La Figure B.8 nous donne un exemple d'une MEEF avec la diffusion et les résultats de la méthode de Zepp et ceux obtenus par l'autoencodeur. Nous constatons que notre méthode arrive à supprimer les diffusions correctement. Tandis que les Figures B.9 et B.10 nous montrent les résultats

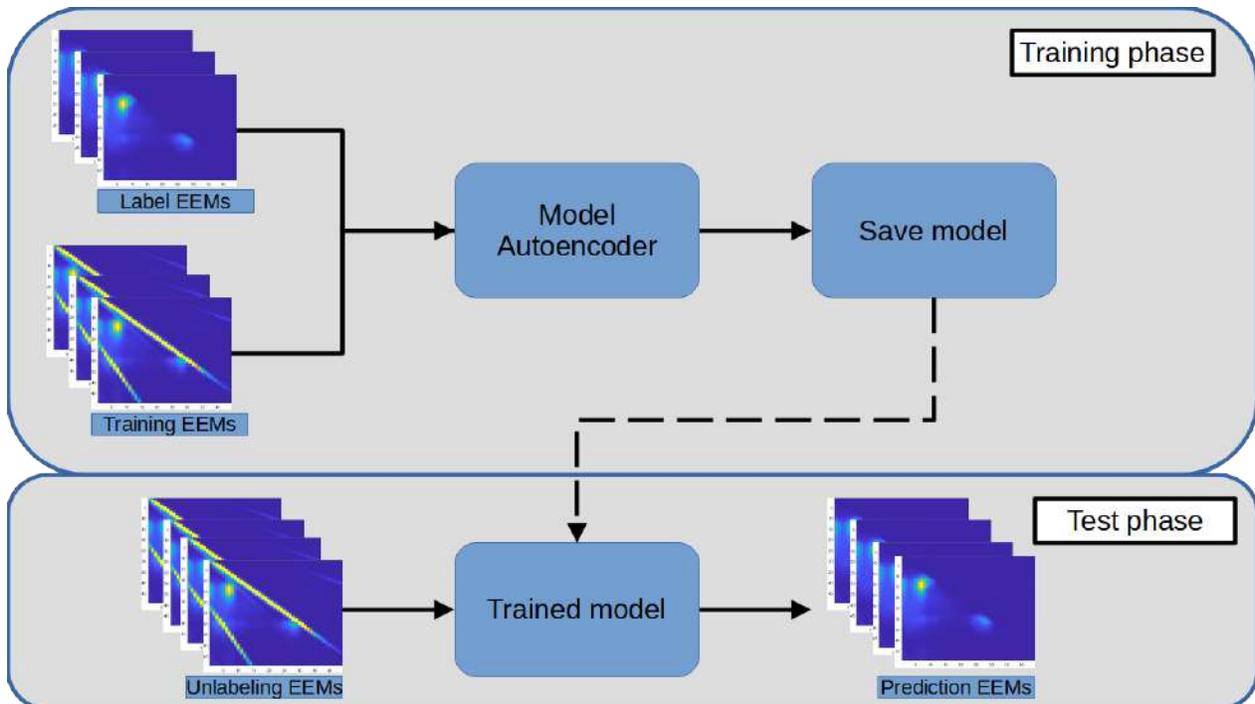


FIGURE B.7 – Les différentes étapes de l'utilisation de notre autoencodeur.

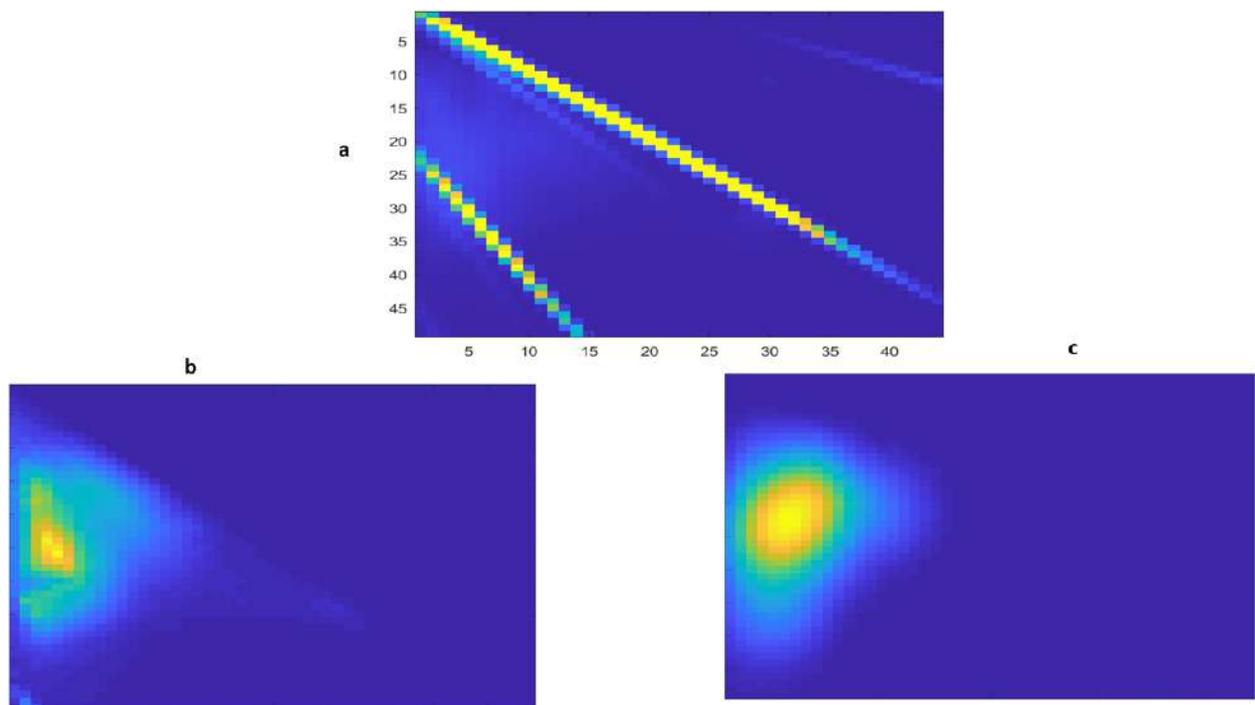
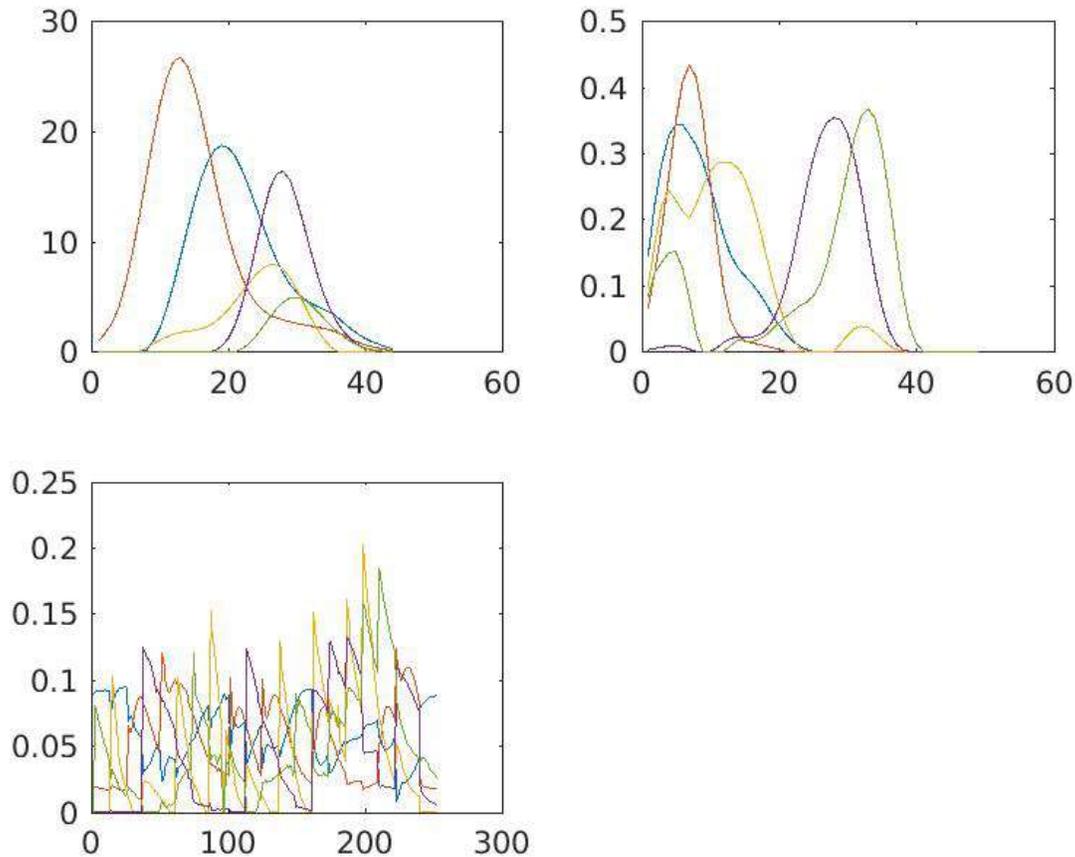


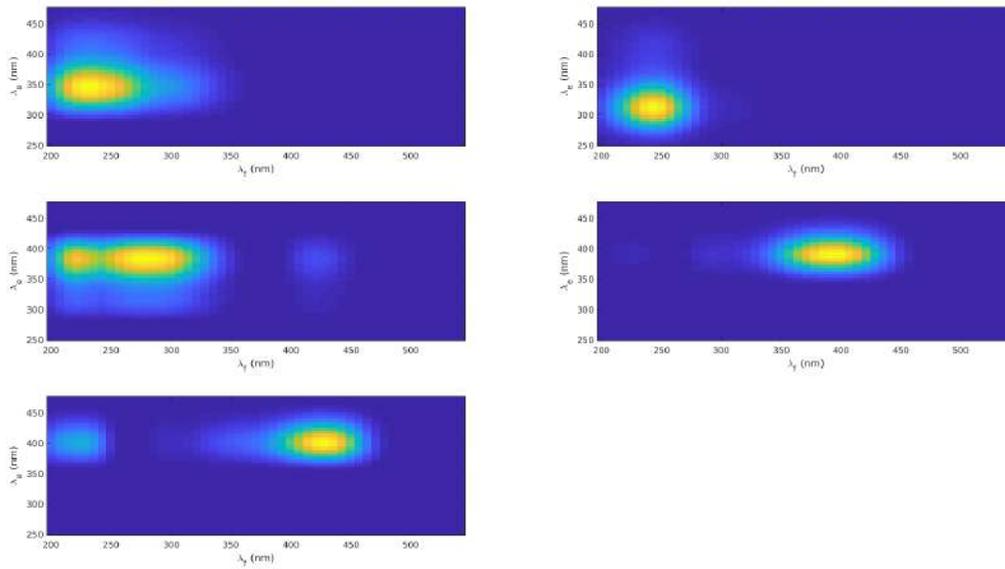
FIGURE B.8 – Suppression de la diffusion sur une MEEF. (a) MEEF bruitée (avec la diffusion). (b) MEEF Corrigée avec la méthode de Zepp. (c) MEEF Corrigée avec la méthode de l'autoencodeur.

de la décomposition obtenus après un prétraitement des MEEFs par autoencodeur. Nous utilisons l'algorithme de Bro avec un rang 5 pour la décomposition. L'algorithme dure 1.7s avant de converger avec un pourcentage de reconstruction de 99,3%. Après la phase d'apprentissage, le prétraitement des MEEFs ne prend que quelques secondes en fonction de la taille du tenseur. Dans ce cas cela dure 1s.

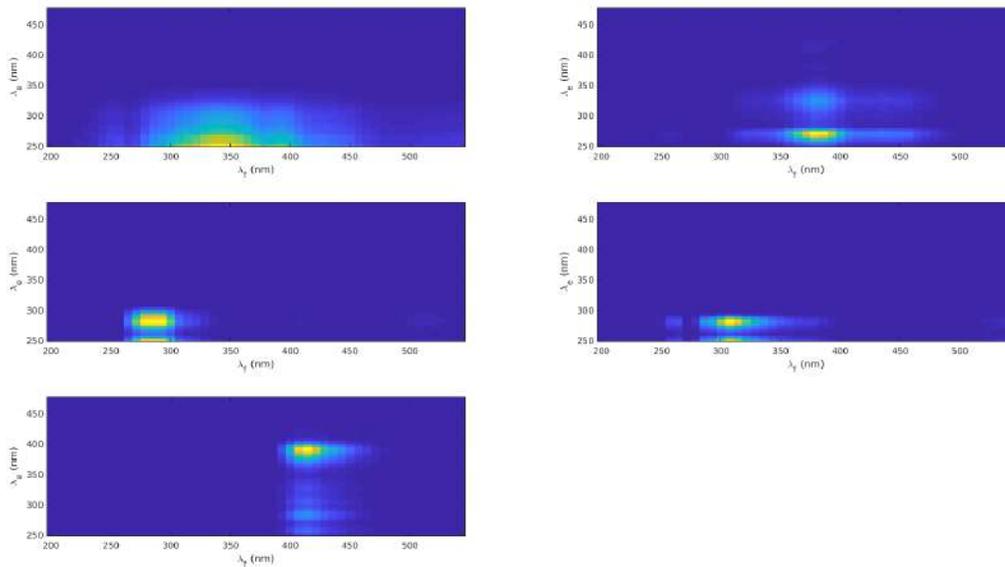


**FIGURE B.9 – Résultats de la CPD obtenus avec un traitement des MEEFs par autoencodeur.**

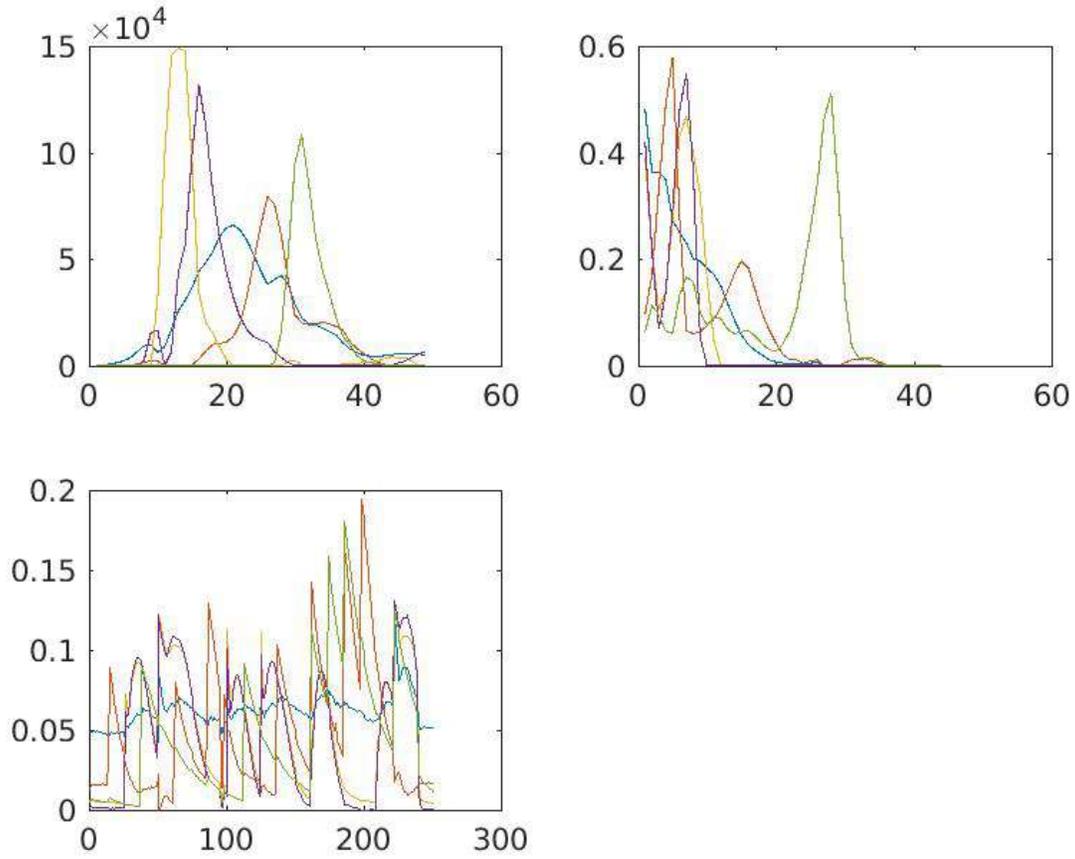
En comparaison, les Figures B.11 et B.12 nous montrent la CPD avec la correction de la méthode de Zepp. Nous remarquons que les concentrations sont assez identiques ainsi que la correction avec l'autoencodeur.



**FIGURE B.10 – Images des composants obtenus avec un traitement des MEEF par autoencodeur après CPD.**



**FIGURE B.11 – Images des composants obtenus avec un traitement des MEEF par la méthode de Zepp.**



**FIGURE B.12 – Résultats de la décomposition obtenus avec un traitement des MEEF par la méthode de Zepp.**

## **B.4 Perspectives et conclusion**

Une nouvelle approche d'architecture profonde est présentée pour la correction des phénomènes de diffusions sur les matrices de fluorescence basée sur les autoencodeurs. La technique est rapide et ne nécessite pas de réglages au préalable. Notre modèle peut évidemment être amélioré notamment en augmentant encore la base d'apprentissage ou ajouter des couches supplémentaires.



# Bibliographie

- [1] D. Maurel. Diagramme de Jablonski.
- [2] Angélique Goffin. *Potentiel d'utilisation de la spectrofluorimétrie 3D pour la caractérisation en ligne de la matière organique dissoute : de la station d'épuration au milieu récepteur*. PhD thesis, 2017.
- [3] Xavier Luciani. *Analyse numérique des spectres de fluorescence 3D issus de mélanges non linéaires*. Theses, Université du Sud Toulon Var, December 2007.
- [4] Wen Chen, Paul Westerhoff, Jerry A. Leenheer, and Karl Booksh. Fluorescence Excitation-Emission Matrix Regional Integration to Quantify Spectra for Dissolved Organic Matter. *Environmental Science and Technology*, 37(24) :5701–5710, 2003.
- [5] Lu Yan, Xianjun Xie, Kang Peng, Ningdi Wang, Yuyao Zhang, Yamin Deng, Yiqun Gan, Qinghua Li, and Yanpeng Zhang. Sources and compositional characterization of chromophoric dissolved organic matter in a Hainan tropical mangrove-estuary. *Journal of Hydrology*, 600(January) :126572, 2021.
- [6] Maria Strack, Yoseph Zuback, Colin McCarter, and Jonathan Price. Changes in dissolved organic carbon quality in soils and discharge 10 years after peatland restoration. *Journal of Hydrology*, 527 :345–354, aug 2015.
- [7] Dianbao Li, Baozhu Pan, Xu Han, Junhua Li, Qingwei Zhu, and Ming Li. Assessing the potential to use CDOM as an indicator of water quality for the sediment-laden Yellow river, China. *Environmental Pollution*, 289(April) :117970, 2021.
- [8] C Guéguen, C Belin, and J Dominik. Organic colloid separation in contrasting aquatic environments with tangential flow filtration. *Water Research*, 36(7) :1677–1684, 2002.
- [9] K. Kalbitz, S. Geyer, and W. Geyer. A comparative characterization of dissolved organic matter by means of original aqueous samples and isolated humic substances. *Chemosphere*, 40(12) :1305–1312, jul 2000.
- [10] Adam Zsolnay, Erik Baigar, Miguel Jimenez, Bernd Steinweg, and Flavia Saccomandi. Differentiating with fluorescence spectroscopy the sources of dissolved organic matter in soils subjected to drying. *Chemosphere*, 38(1) :45–50, jan 1999.
- [11] Antonio Nebbioso and Alessandro Piccolo. Molecular characterization of dissolved organic matter (DOM) : a critical review. *Analytical and bioanalytical chemistry*, 405(1) :109–124, 2013.
- [12] Namguk Her, Gary Amy, Diane McKnight, Jinsik Sohn, and Yeomin Yoon. Characterization of DOM as a function of MW by fluorescence EEM and HPLC-SEC using UVA, DOC, and fluorescence detection. *Water research*, 37(17) :4295–4303, 2003.

- [13] Stéphane Mounier and Roland Redon. The Use of 3-D Fluorescence and Its Decomposition in Environmental Organic Matter Studies. In *Encyclopedia of Analytical Chemistry*, pages 1–16. John Wiley Sons, Ltd, Chichester, UK, jun 2018.
- [14] Anu Matilainen, Egil T Gjessing, Tanja Lahtinen, Leif Hed, Amit Bhatnagar, and Mika Sillanpää. An overview of the methods used in the characterisation of natural organic matter (nom) in relation to drinking water treatment. *Chemosphere*, 83(11) :1431–1442, 2011.
- [15] Elfrida M Carstea, Cristina L Popa, Andy Baker, and John Bridgeman. In situ fluorescence measurements of dissolved organic matter : A review. *Science of The Total Environment*, page 134361, 2019.
- [16] R. A. Harshman. Foundation of PARAFAC procedure : Models and conditions for an ‘explanatory’ multi-mode factor analysis. *UCLA working papers in Phonetics*, (16) :1–84, 1970.
- [17] Aaron S Field and Daniel Graupe. Topographic component (parallel factor) analysis of multichannel evoked potentials : practical issues in trilinear spatiotemporal decomposition. *Brain topography*, 3(4) :407–423, 1991.
- [18] Rasmus Bro. PARAFAC. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2) :149–171, 1997.
- [19] Hongcheng Wang and Narendra Ahuja. Compact representation of multidimensional data using tensor rank-one decomposition. *vectors*, 1 :5, 2004.
- [20] Ivica Kopriva and Andrzej Cichocki. Blind multispectral image decomposition by 3D nonnegative tensor factorization. *Optics letters*, 34(14) :2210–2212, 2009.
- [21] Qiang Zhang, Han Wang, Robert J Plemmons, and V Pau’l Pauca. Tensor methods for hyperspectral data analysis : a space object material identification study. *JOSA A*, 25(12) :3001–3012, 2008.
- [22] Marc Castella, Pascal Bianchi, Antoine Chevreuil, and Jean-Christophe Pesquet. A blind source separation framework for detecting cpm sources mixed by a convolutive mimo filter. *Signal Processing*, 86(8) :1950–1967, 2006.
- [23] Pascal Chevalier, Laurent Albera, Pierre Comon, and Anne Ferréol. Comparative performance analysis of eight blind source separation methods on radiocommunications signals. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 1, pages 273–278. IEEE, 2004.
- [24] Dimitri Nion and Lieven De Lathauwer. An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma. *Signal Processing*, 88(3) :749–755, 2008.
- [25] Muhammad Farooq Saleem Khan, Mona Akbar, Jing Wu, and Zhou Xu. A review on fluorescence spectroscopic analysis of water and wastewater. *Methods and Applications in Fluorescence*, pages 0–23, nov 2021.
- [26] Jonathan Bridgeman, A Baker, D Brown, and JB Boxall. Portable led fluorescence instrumentation for the rapid assessment of potable water quality. *Science of the Total Environment*, 524 :338–346, 2015.
- [27] Charlotte Dupont, Philippe Cousin, and Samuel Dupont. Iot for aquaculture 4.0 smart and easy-to-deploy real-time water monitoring with iot. In *2018 Global Internet of Things Summit (GloTS)*, pages 1–5, 2018.

- [28] Wolfgang Ludwig, Jean-Luc Probst, and Stefan Kempe. Predicting the oceanic input of organic carbon by continental erosion. *Global Biogeochemical Cycles*, 10(1) :23–41, 1996.
- [29] Susan Cumberland, John Bridgeman, Andy Baker, Mark Sterling, and David Ward. Fluorescence spectroscopy as a tool for determining microbial quality in potable water applications. *Environmental Technology*, 33(6) :687–693, mar 2012.
- [30] Ledyard R Tucker. Implications of factor analysis of three-way matrices for measurement of change. *Problems in measuring change*, 15(122-137) :3, 1963.
- [31] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3) :279–311, 1966.
- [32] Xavier Luciani, Stéphane Mounier, Roland Redon, and André Bois. A simple correction method of inner filter effects affecting feem and its application to the PARAFAC decomposition. *Chemometrics and Intelligent Laboratory Systems*, 96(2) :227–238, 2009.
- [33] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3) :455–500, 2009.
- [34] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Pappalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13) :3551–3582, 2017.
- [35] Jean-Philip Royer. *Identification aveugle de mélanges et décomposition canonique de tenseurs : application à l'analyse de l'eau*. Theses, Université Nice Sophia Antipolis, October 2013.
- [36] Rémi André. *Algorithmes de diagonalisation conjointe par similitude pour la décomposition canonique polyadique de tenseurs : applications en séparation de sources*. Theses, Université de Toulon, September 2018.
- [37] R. A. Horn. The hadamard product. 40 :87–169, 5 1990.
- [38] J. W. Brewer. Kronecker products and matrix calculus in system theory. 25 :772 – 781, 10 1978.
- [39] C. G. Khatri and C. R. Rao. Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhya : Indian J. Statistics, Series A*, 30 :167–180, 1968.
- [40] Peter McCullagh. *Tensor methods in statistics : Monographs on statistics and applied probability*. Chapman and Hall/CRC, 2018.
- [41] Rasmus Bro. Multi-way analysis in the food industry. *Models, Algorithms, and Applications. Academish proefschrift. Dinamarca*, 1998.
- [42] Ledyard R Tucker et al. The extension of factor analysis to three-dimensional matrices. *Contributions to mathematical psychology*, 110119, 1964.
- [43] Yangyang Xu. On the convergence of higher-order orthogonal iteration. *Linear and Multilinear Algebra*, 66(11) :2247–2265, 2018.
- [44] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications : From two-way to multiway component analysis. *IEEE signal processing magazine*, 32(2) :145–163, 2015.
- [45] Abraham Traoré, Maxime Berar, and Alain Rakotomamonjy. Online multimodal dictionary learning. *Neurocomputing*, 368 :163–179, 2019.

- [46] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4) :164–189, 1927.
- [47] Frank L Hitchcock. Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7(1-4) :39–79, 1928.
- [48] Raymond B Cattell. “parallel proportional profiles” and other principles for determining the choice of factors by rotation. *Psychometrika*, 9(4) :267–283, 1944.
- [49] Raymond B Cattell. The three basic factor-analytic research designs—their interrelations and derivatives. *Psychological bulletin*, 49(5) :499, 1952.
- [50] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear image analysis for facial recognition. In *Object recognition supported by user interaction for service robots*, volume 2, pages 511–514. IEEE, 2002.
- [51] Ravdeep Pasricha, Ekta Gujral, and Evangelos E Papalexakis. Identifying and alleviating concept drift in streaming tensor decomposition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 327–343. Springer, 2018.
- [52] Rasmus Bro and Henk AL Kiers. A new efficient method for determining the number of components in PARAFAC models. *Journal of Chemometrics*, 17(5) :274–286, 2003.
- [53] Colin A Stedmon and Rasmus Bro. Characterizing dissolved organic matter fluorescence with parallel factor analysis : a tutorial. *Limnology and Oceanography : Methods*, 6(11) :572–579, 2008.
- [54] Evangelos E Papalexakis. Automatic unsupervised tensor mining with quality assessment. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 711–719. SIAM, 2016.
- [55] João Paulo Carvalho Lustosa da Costa, Florian Roemer, Martin Haardt, and Rafael Timóteo de Sousa. Multi-dimensional model order selection. *EURASIP Journal on Advances in Signal Processing*, 2011(1) :1–13, 2011.
- [56] Zhaoshui He, Andrzej Cichocki, Shengli Xie, and Kyuwan Choi. Detecting the number of clusters in n-way probabilistic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11) :2006–2021, 2010.
- [57] Canyi Lu, Changbo Zhu, Chunyan Xu, Shuicheng Yan, and Zhouchen Lin. Generalized singular value thresholding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.
- [58] Saeed Pouryazdian, Soosan Beheshti, and Sridhar Krishnan. Candecomp/parafac model order selection based on reconstruction error in the presence of kronecker structured colored noise. *Digital Signal Processing*, 48 :12–26, 2016.
- [59] Morten Mørup and Lars Kai Hansen. Automatic relevance determination for multi-way models. *Journal of Chemometrics : A Journal of the Chemometrics Society*, 23(7-8) :352–363, 2009.
- [60] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6) :716–723, 1974.
- [61] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5) :465–471, 1978.
- [62] Mati Wax and Thomas Kailath. Detection of signals by information theoretic criteria. *IEEE Transactions on acoustics, speech, and signal processing*, 33(2) :387–392, 1985.

- [63] Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences*, 6(3) :1758–1789, 2013.
- [64] Xuan Vu, Caroline Chauv, Nadège Thirion-Moreau, Sylvain Maire, and Elfrida Mihaela Carstea. A new penalized nonnegative third-order tensor decomposition using a block coordinate proximal gradient approach : Application to 3d fluorescence spectroscopy. *Journal of Chemometrics*, 31(4) :2859, 2017.
- [65] Jeremy Emile Cohen and Nicolas Gillis. Dictionary-based tensor canonical polyadic decomposition. *IEEE Transactions on Signal Processing*, 66(7) :1876–1889, Apr 2018.
- [66] Emilie Chouzenoux, Jean-Christophe Pesquet, and Audrey Repetti. A block coordinate variable metric forward–backward algorithm. *Journal of Global Optimization*, 66(3) :457–485, 2016.
- [67] Rémi André, Xavier Luciani, Laurent Albera, and Eric Moreau. A two-step algorithm for joint eigenvalue decomposition-application to canonical polyadic decomposition of fluorescence spectra. *Chemometrics and Intelligent Laboratory Systems*, 206 :104065, 2020.
- [68] Joseph B Kruskal. Three-way arrays : rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2) :95–138, 1977.
- [69] Richard A Harshman and Margaret E Lundy. The parafac model for three-way factor analysis and multidimensional scaling. *Research methods for multimode data analysis*, 46 :122–215, 1984.
- [70] Nicholas D Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of chemometrics*, 14(3) :229–239, 2000.
- [71] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6) :391–407, 1990.
- [72] Florian Roemer and Martin Haardt. A semi-algebraic framework for approximate cp decompositions via simultaneous matrix diagonalizations (secsi). *Signal Processing*, 93(9) :2722–2738, 2013.
- [73] Florian Roemer and Martin Haardt. A closed-form solution for multilinear parafac decompositions. In *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 487–491. IEEE, 2008.
- [74] Xavier Luciani and Laurent Albera. Canonical polyadic decomposition based on joint eigenvalue decomposition. *Chemometrics and Intelligent Laboratory Systems*, 132 :152–167, 2014.
- [75] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. *Psychometrika*, 35(3) :283–319, 1970.
- [76] Athanasios P Liavas and Nicholas D Sidiropoulos. Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 63(20) :5450–5463, 2015.
- [77] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. Optimization-based algorithms for tensor decompositions : Canonical polyadic decomposition, decomposition in rank-( $l_r, l_r, 1$ ) terms, and a new generalization. *SIAM Journal on Optimization*, 23(2) :695–720, 2013.

- [78] Michiel Vandecappelle, Nico Vervliet, and Lieven De Lathauwer. Nonlinear least squares updating of the canonical polyadic decomposition. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 663–667. IEEE, 2017.
- [79] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.
- [80] Jeremy Cohen, Rodrigo Cabral Farias, and Pierre Comon. Fast decomposition of large nonnegative tensors. *IEEE Signal Processing Letters*, 22(7) :862–866, 2014.
- [81] William S Rayens and Benjamin C Mitchell. Two-factor degeneracies and a stabilization of parafac. *Chemometrics and Intelligent Laboratory Systems*, 38(2) :173–181, 1997.
- [82] Henk AL Kiers. A three–step algorithm for candecomp/parafac analysis of large data sets with multicollinearity. *Journal of Chemometrics*, 12(3) :155–171, 1998.
- [83] Myriam Rajih, Pierre Comon, and Richard A Harshman. Enhanced line search : A novel method to accelerate PARAFAC. *SIAM journal on matrix analysis and applications*, 30(3) :1128–1147, 2008.
- [84] Giorgio Tomasi and Rasmus Bro. A comparison of algorithms for fitting the parafac model. *Computational Statistics & Data Analysis*, 50(7) :1700–1734, 2006.
- [85] Pierre Comon, Xavier Luciani, and André LF De Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics : A Journal of the Chemometrics Society*, 23(7-8) :393–405, 2009.
- [86] Souquières Laura, Cyril Prissette, Sylvain Maire, and Nadège Thirion-Moreau. A parallel strategy for an evolutionary stochastic algorithm : application to the cp decomposition of nonnegative n-th order tensors. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1956–1960. IEEE, 2021.
- [87] Nico Vervliet and Lieven De Lathauwer. A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors. *IEEE Journal of Selected Topics in Signal Processing*, 10(2) :284–295, 2015.
- [88] Nikos D Sidiropoulos, Evangelos E Papalexakis, and Christos Faloutsos. A parallel algorithm for big tensor decomposition using randomly compressed cubes (paracomp). In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2014.
- [89] P Comon. *Mathematics in signal processing : V*, chapter tensor decompositions, 2002.
- [90] Shirin Hassanzadeh and Azam Karami. Compression and noise reduction of hyperspectral images using non-negative tensor decomposition and compressed sensing. *European Journal of Remote Sensing*, 49(1) :587–598, 2016.
- [91] Miguel A Veganzones, Jérémy E Cohen, Rodrigo Cabral Farias, Jocelyn Chanussot, and Pierre Comon. Nonnegative tensor cp decomposition of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(5) :2577–2588, 2015.
- [92] Kathleen R Murphy, Colin A Stedmon, Daniel Graeber, and Rasmus Bro. Fluorescence spectroscopy and multi-way techniques.PARAFAC. *Analytical Methods*, 5(23) :6557–6566, 2013.
- [93] Jean-Philip Royer, Nadège Thirion-Moreau, Pierre Comon, Roland Redon, and Stéphane Mounier. A regularized nonnegative canonical polyadic decomposition algorithm with

- preprocessing for 3d fluorescence spectroscopy. *Journal of Chemometrics*, 29(4) :253–265, 2015.
- [94] K. Huang, N. D. Sidiropoulos, and A. P. Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19) :5052–5065, 2016.
- [95] Mohamad Jouni. *Image Analysis Based on Tensor Representations*. Theses, Université Grenoble Alpes [2020-....], January 2021.
- [96] Mohamad Jouni, Mauro Dalla Mura, and Pierre Comon. Some issues in computing the cp decomposition of nonnegative tensors. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 57–66. Springer, 2018.
- [97] Shuai Zhang and Jack Xin. Minimization of transformed  $L_1$  penalty : theory, difference of convex function algorithm, and robust application in compressed sensing. *Mathematical Programming*, 169(1) :307–336, 2018.
- [98] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808. Citeseer, 2007.
- [99] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3) :471–501, 2010.
- [100] Ming Yuan and Cun-Hui Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4) :1031–1068, 2016.
- [101] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.
- [102] Xu Han, Laurent Albera, Amar Kachenoura, Lotfi Senhadji, and Huazhong Shu. Low rank canonical polyadic decomposition of tensors based on group sparsity. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 668–672. IEEE, 2017.
- [103] Xianbiao Shu, Fatih Porikli, and Narendra Ahuja. Robust orthonormal subspace learning : Efficient recovery of corrupted low-rank matrices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3874–3881, 2014.
- [104] Christian Steffens, Marius Pesavento, and Marc E Pfetsch. A compact formulation for the  $l_{2,1}$  mixed-norm minimization problem. *IEEE Transactions on Signal Processing*, 66(6) :1483–1497, 2018.
- [105] Alexandre Gramfort, Matthieu Kowalski, and Matti Hämmäläinen. Mixed-norm estimates for the m/eeg inverse problem using accelerated gradient methods. *Physics in Medicine & Biology*, 57(7) :1937, 2012.
- [106] Yong Chen, Wei He, Naoto Yokoya, and Ting-Zhu Huang. Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition. *IEEE transactions on cybernetics*, 50(8) :3556–3570, 2019.
- [107] Jose Henrique de Moraes Goulart, Pedro Marinho Ramos de Oliveira, Rodrigo Cabral Farias, Vicente Zarzoso, and Pierre Comon. Alternating group lasso for block-term tensor decomposition and application to ecg source separation. *IEEE Transactions on Signal Processing*, 68 :2682–2696, 2020.

- [108] Guillaume Olikier, P-A Absil, and Lieven De Lathauwer. Variable projection applied to block term decomposition of higher-order tensors. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 139–148. Springer, 2018.
- [109] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs : dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383, 2006.
- [110] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S Yu, and Christos Faloutsos. Incremental tensor analysis : Theory and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(3) :1–37, 2008.
- [111] Ekta Gujral, Ravdeep Pasricha, Tianxiong Yang, and Evangelos E Papalexakis. Octen : Online compression-based tensor decomposition. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 455–459. IEEE, 2019.
- [112] Isaac Wilfried Sanou, Roland Redon, Xavier Luciani, and Stéphane Mounier. Online nonnegative canonical polyadic decomposition : Algorithms and application. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 1805–1809. IEEE, 2021.
- [113] Andrzej Cichocki. Era of big data processing : A new approach via tensor networks and tensor decompositions. *arXiv preprint arXiv :1403.2048*, 2014.
- [114] Ming Shi, Dan Li, and Jian Qiu Zhang. An alternating bayesian approach to PARAFAC decomposition of tensors. *IEEE Access*, 6 :36487–36499, 2018.
- [115] Sukhan Lee and Youngchul Kay. A kalman filter approach for accurate 3-d motion estimation from a sequence of stereo images. *CVGIP : Image Understanding*, 54(2) :244–258, 1991.
- [116] Andrews Sobral, Sajid Javed, Soon Ki Jung, Thierry Bouwmans, and El-hadi Zahzah. Online stochastic tensor decomposition for background subtraction in multispectral video sequences. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 106–113, 2015.
- [117] Dimitri Nion and Nicholas D Sidiropoulos. Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor. *IEEE Transactions on Signal Processing*, 57(6) :2299–2310, 2009.
- [118] Anh Huy Phan and Andrzej Cichocki. PARAFAC algorithms for large-scale problems. *Neurocomputing*, 74(11) :1970–1984, 2011.
- [119] Shuo Zhou, X. V. Nguyen, J. Bailey, Yunzhe Jia, and I. Davidson. Accelerating online CP decompositions for higher order tensors. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1375–1384. ACM, 2016.
- [120] Ekta Gujral, Ravdeep Pasricha, and Evangelos E Papalexakis. Sambaten : Sampling-based batch incremental tensor decomposition. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 387–395. SIAM, 2018.
- [121] Xiang Ma, Dan Schonfeld, and Ashfaq Khokhar. Dynamic updating and dwn dating matrix svd and tensor hosvd for adaptive indexing and retrieval of motion trajectories. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1129–1132. IEEE, 2009.

- [122] Weiming Hu, Xi Li, Xiaoqin Zhang, Xinchu Shi, Stephen Maybank, and Zhongfei Zhang. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *International Journal of Computer Vision*, 91(3) :303–327, 2011.
- [123] Athanasios A Rontogiannis, Eleftherios Kofidis, and Paris V Giampouras. Online rank-revealing block-term tensor decomposition. *arXiv preprint arXiv :2106.10755*, 2021.
- [124] Run-Ze Xu, Jia-Shun Cao, Ganyu Feng, Jing-Yang Luo, Qian Feng, Bing-Jie Ni, and Fang Fang. Fast identification of fluorescent components in three-dimensional excitation-emission matrix fluorescence spectra via deep learning. *Chemical Engineering Journal*, 430 :132893, 2022.
- [125] Jan Niesing. *Simultaneous component and factor analysis methods for two or more groups : a comparative study*. DSWO Press Leiden, The Netherlands, 1997.
- [126] Richard G Zepp, Wade M Sheldon, and Mary Ann Moran. Dissolved organic fluorophores in southeastern us coastal waters : correction method for eliminating rayleigh and raman scattering peaks in excitation–emission matrices. *Marine chemistry*, 89(1-4) :15–36, 2004.
- [127] Jeremy Emile Cohen, Pierre Comon, and Xavier Luciani. Correcting inner filter effects, a non multilinear tensor decomposition method. *Chemometrics and Intelligent Laboratory Systems*, 150 :29–40, 2016.
- [128] Greger G Andersson, Brian K Dable, and Karl S Booksh. Weighted parallel factor analysis for calibration of hplc-uv/vis spectrometers in the presence of beer’s law deviations. *Chemometrics and Intelligent Laboratory Systems*, 49(2) :195–213, 1999.
- [129] Colin A Stedmon, Stiig Markager, and Rasmus Bro. Tracing dissolved organic matter in aquatic environments using a new approach to fluorescence spectroscopy. *Marine chemistry*, 82(3-4) :239–254, 2003.
- [130] Claus A Andersson and Rasmus Bro. The n-way toolbox for matlab. *Chemometrics and intelligent laboratory systems*, 52(1) :1–4, 2000.
- [131] TG Kolda, BW BADER, JM SUN, et al. Matlab tensor toolbox. version 2.2, 2007.
- [132] Nico Vervliet, Otto Debals, and Lieven De Lathauwer. Tensorlab 3.0—numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 1733–1738. IEEE, 2016.
- [133] Anh-Huy Phan, P Tichavský, and Andrzej Cichocki. Tensorbox : a matlab package for tensor decomposition. *LABSP, RIKEN, Japan*, 2012.
- [134] Sen Jia, Yuntao Qian, Jiming Li, Yan Li, and Zhong Ming. Hierarchical alternating least squares algorithm with sparsity constraint for hyperspectral unmixing. In *2010 IEEE International Conference on Image Processing*, pages 2305–2308. IEEE, 2010.
- [135] Anh Huy Phan, Petr Tichavský, and Andrzej Cichocki. On fast computation of gradients for CANDECOMP/PARAFAC algorithms. *arXiv preprint arXiv :1204.1586*, 2012.
- [136] Anh Huy Phan and Andrzej Cichocki. Multi-way nonnegative tensor factorization using fast hierarchical alternating least squares algorithm (hals). In *Proc. of The 2008 International Symposium on Nonlinear Theory and its Applications*, 2008.
- [137] Lieven De Lathauwer and Dimitri Nion. Decompositions of a higher-order tensor in block terms—part iii : Alternating least squares algorithms. *SIAM journal on Matrix Analysis and Applications*, 30(3) :1067–1083, 2008.

- [138] Jean-Philip Royer, Nadege Thirion-Moreau, and Pierre Comon. Computing the polyadic decomposition of nonnegative third order tensors. *Signal Processing*, 91(9) :2159–2171, 2011.
- [139] Xuan Vu, Caroline Chaux, Nadège Thirion-Moreau, Sylvain Maire, and Elfrida Mihaela Carstea. A new penalized nonnegative third order tensor decomposition using a block coordinate proximal gradient approach : application to 3D fluorescence spectroscopy. *Journal of Chemometrics*, 31(4) :e2859, April 2017.
- [140] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.
- [141] Tamara G Kolda and David Hong. Stochastic gradients for large-scale tensor decomposition. *SIAM Journal on Mathematics of Data Science*, 2(4) :1066–1095, 2020.
- [142] Ali Anaissi, Basem Suleiman, and Seid Miad Zandavi. Online tensor-based learning model for structural damage detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(6) :1–18, 2021.
- [143] Wenqiang Pu, Shahana Ibrahim, Xiao Fu, and Mingyi Hong. Fiber-sampled stochastic mirror descent for tensor decomposition with  $\beta$ -divergence. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2925–2929. IEEE, 2021.
- [144] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1) :145–151, 1999.
- [145] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence  $O(1/k^2)$ . In *Doklady an ussr*, volume 269, pages 543–547, 1983.
- [146] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- [147] Matthew D Zeiler. Adadelta : an adaptive learning rate method. *arXiv preprint arXiv :1212.5701*, 2012.
- [148] Diederik P Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.
- [149] Timothy Dozat. Incorporating NESTEROV momentum into ADAM. *Proceedings of 4th International Conference on Learning Representations, Workshop Track*, 2016.
- [150] Ersan Yazan and M Fatih Talu. Comparison of the stochastic gradient descent based optimization techniques. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–5. IEEE, 2017.
- [151] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. Methods for non-linear least squares problems. 2004.
- [152] Dennis L Sun and Cedric Fevotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6201–6205. IEEE, 2014.
- [153] Rafal Zdunek and Andrzej Cichocki. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems. *Computational intelligence and neuroscience*, 2008, 2008.

- [154] Ke Huang and Selin Aviyente. Sparse representation for signal classification. *Advances in neural information processing systems*, 19 :609–616, 2006.
- [155] Joseph F Murray and Kenneth Kreutz-Delgado. Sparse image coding using learned overcomplete dictionaries. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing, 2004.*, pages 579–588. IEEE, 2004.
- [156] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6) :2014–2039, 2011.
- [157] Leila Belmerhnia, El-Hadi Djerroune, Cédric Carteret, and David Brie. Simultaneous regularized sparse approximation for wood wastes nir spectra features selection. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 437–440. IEEE, 2015.
- [158] Jingu Kim and Haesun Park. Sparse nonnegative matrix factorization for clustering. Technical report, Georgia Institute of Technology, 2008.
- [159] Duy Khuong Nguyen and Tu Bao Ho. Fast parallel randomized algorithm for nonnegative matrix factorization with kl divergence for large sparse datasets. *arXiv preprint arXiv :1604.04026*, 2016.
- [160] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the  $\beta$ -divergence. *Neural computation*, 23(9) :2421–2456, 2011.
- [161] Michael Elad and Michal Aharon. Image denoising via learned dictionaries and sparse representation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 895–900. IEEE, 2006.
- [162] NUS Ludivine, Sebastian Miron, Benoît Jaillais, Said Moussaoui, and BRIE David. A semi-supervised rank tracking algorithm for on-line unmixing of hyperspectral images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1514–1518. IEEE, 2020.
- [163] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Collaborative sparse regression for hyperspectral unmixing. *IEEE Transactions on geoscience and remote sensing*, 52(1) :341–354, 2013.
- [164] Jérémy Emile Cohen and Nicolas Gillis. Dictionary-based tensor canonical polyadic decomposition. *IEEE Transactions on Signal Processing*, 66(7) :1876–1889, 2017.
- [165] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696, 2009.
- [166] Isaac Wilfried Sanou, Roland Redon, Xavier Luciani, and Stéphane Mounier. Online nonnegative and sparse canonical polyadic decomposition of fluorescence tensors. *Chemometrics and Intelligent Laboratory Systems*, 225 :104550, 2022.
- [167] Audrey White. Effect of ph on fluorescence of tyrosine, tryptophan and related compounds. *Biochemical Journal*, 71(2) :217, 1959.
- [168] Henryk Szmacinski and Joseph R Lakowicz. Optical measurements of ph using fluorescence lifetimes and phase-modulation fluorometry. *Analytical chemistry*, 65(13) :1668–1674, 1993.
- [169] Michael R Osborne, Brett Presnell, and Berwin A Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3) :389–403, 2000.

- [170] Dmitry M Malioutov, Müjdat Cetin, and Alan S Willsky. Homotopy continuation for sparse signal representation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–733. IEEE, 2005.
- [171] Sachin Singh, Rita K Henderson, Andy Baker, Richard M Stuetz, and Stuart J Khan. Online fluorescence monitoring of ro fouling and integrity : analysis of two contrasting recycled water schemes. *Environmental Science : Water Research & Technology*, 1(5) :689–698, 2015.
- [172] Paula G Coble, Jamie Lead, Andy Baker, Darren M Reynolds, and Robert GM Spencer. *Aquatic organic matter fluorescence*. Cambridge University Press, 2014.
- [173] Jerry A Leenheer and Jean-Philippe Croué. Peer reviewed : characterizing aquatic dissolved organic matter. *Environmental science & technology*, 37(1) :18A–26A, 2003.
- [174] SU Khan and M Schnitzer. Permanganate oxidation of humic acids, fulvic acids, and humins extracted from ah horizons of a black chernozem, a black solod, and a black solonetz soil. *Canadian Journal of Soil Science*, 52(1) :43–51, 1972.
- [175] Paula G. Coble. Characterization of marine and terrestrial DOM in seawater using excitation-emission matrix spectroscopy. *Marine Chemistry*, 51, 1996.
- [176] Paula G Coble. Characterization of marine and terrestrial dom in seawater using excitation-emission matrix spectroscopy. *Marine chemistry*, 51(4) :325–346, 1996.
- [177] Edith Parlanti, K Wörz, Laure Geoffroy, and Michel Lamotte. Dissolved organic matter fluorescence spectroscopy as a tool to estimate biological activity in a coastal zone submitted to anthropogenic inputs. *Organic geochemistry*, 31(12) :1765–1781, 2000.
- [178] Maxime Lafont, Samuel Dupont, Philippe Cousin, Ambre Vallauri, and Charlotte Dupont. Back to the future : Iot to improve aquaculture : Real-time monitoring and algorithmic prediction of water parameters for aquaculture needs. In *2019 Global IoT Summit (GIoTS)*, pages 1–6, 2019.
- [179] J Serra. Les structures gigognes : morphologie mathématique et interprétation métallogénique. *Mineralium Deposita*, 3(2) :135–154, 1968.
- [180] Georges Matheron. *Le krigeage universel*, volume 1. École nationale supérieure des mines de Paris Paris, 1969.
- [181] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1) :1929–1958, 2014.

