HAL
open science

Study Of Different Strategies For The Canonical Polyadic Decomposition Of Nonnegative Third Order Tensors With Application To The Separation Of Spectra In 3D Fluorescence Spectroscopy

Xuan Thanh Vu †, Caroline Chaux, Sylvain Maire, Nadège Thirion-Moreau

To cite this version:

Xuan Thanh Vu †, Caroline Chaux, Sylvain Maire, Nadège Thirion-Moreau. Study Of Different Strategies For The Canonical Polyadic Decomposition Of Nonnegative Third Order Tensors With Application To The Separation Of Spectra In 3D Fluorescence Spectroscopy. 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2014), Sep 2014, Reims, France. hal-01278483

HAL Id: hal-01278483

https://hal.science/hal-01278483

Submitted on 24 Feb 2016

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

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

STUDY OF DIFFERENT STRATEGIES FOR THE CANONICAL POLYADIC DECOMPOSITION OF NONNEGATIVE THIRD ORDER TENSORS WITH APPLICATION TO THE SEPARATION OF SPECTRA IN 3D FLUORESCENCE SPECTROSCOPY

Xuan Thanh VU[1,2*] *Caroline CHAUX*[2] *Sylvain MAIRE*[1] *Nadège THIRION-MOREAU*[1]

Aix-Marseille Université, CNRS, ENSAM,
LSIS, UMR 7296, F-13397 Marseille
Université de Toulon, CNRS,
LSIS, UMR 7296, F-83957 La Garde
France
{vu, maire, thirion}@univ-tln.fr

Aix-Marseille Université, CNRS, CMI
LATP, UMR 7353, F-13453 Marseille
France

caroline.chaux@latp.univ-mrs.fr

ABSTRACT

In this communication, the problem of blind source separation in chemical analysis and more precisely in the fluorescence spectroscopy framework is addressed. Classically multi-linear Canonical Polyadic (CP or Candecomp/Parafac) decompositions algorithms are used to perform that task. Yet, as the constituent vectors of the loading matrices should be nonnegative since they stand for nonnegative quantities (spectra and concentrations), we focus on NonNegative CP decomposition algorithms (NNCP). In the unconstrained case, two types of trilinear (or triadic) decomposition model have been studied. Here, our aim is to investigate different strategies concerning the choice of models and optimization schemes in the case of a nonnegativity constraint. Computer simulations are performed on synthetic data to illustrate the robustness of the proposed approaches versus overfactoring problems but also the critical importance of the use of regularization terms.

***Index Terms*—** Nonnegative Tensor Factorization (NTF); multi-linear algebra; Candecomp/Parafac; Non Negative Canonical Polyadic decomposition (NNCP); data analysis

1. INTRODUCTION

The problem of blind source separation in chemical analysis is considered, focusing more precisely on the fluorescence spectroscopy framework. Since the works of Stedmon [1, 2], the technique classically used in this field of applications to separate the contributions of the different compounds present in the mixtures consists of coupling three dimensional (3D) fluorescence spectroscopy (or Fluorescence Excitation-Emission Matrices (FEEM)) together with multi-linear Canonical Polyadic (CP or Candecomp/Parafac) decompositions algorithms [3][4]. In fact, a set of several FEEM constitutes a nonnegative third order tensor which can be modeled by multi-linear decompositions and more precisely by Canonical Polyadic Decompositions (CPD), thanks to the Beer-Lambert law [5], for relatively small concentrations of the compounds (*a.k.a. fluorophores*). Moreover, in the fluorescence spectroscopy context, the constituent vectors of the three loading matrices should be nonnegative since they stand for intrinsically nonnegative quantities (emission, excitation spectra and concentrations), that is why we will restrict our attention to NonNegative CP decomposition algorithms (NNCP). The main advantage of the nonnegativity constraint is that the low rank approximation problem becomes well posed [6]. Its counterpart is that its level of difficulty might increase. The way to take into account this nonnegativity constraint has been extensively studied over the past few years, giving rise to numerous works among which are [7][8][9][10][11][12][13]. A very simple way consists of considering iterative alternating (minimization) optimization schemes where at each iteration the non-negativity contraint is imposed by a projection. The well-known NTF-ALS and NTF-HALS algorithms [8] are based on such an idea. In [10], we have developed an alternative approach based on a square change of variables making it possible to rewrite the tensor decomposition problem as an unconstrained minimization problem, whilst ensuring the nonnegativity constraint.

Furthermore, in the unconstrained case, two types of trilinear (or triadic) decomposition model have been considered. In this communication, our aim is to extend the approach developed in [10] to the second type of model, and then to study different strategies concerning the optimization scheme and the regularization terms that are added to the cost function in order to evaluate their impact on the obtained results. To that aim, computer simulations are performed on synthetic data which are very close to real spectroscopic data. They emphasize the critical importance of l_1-norm regularization terms

*Thanks to Labex Archimède (Aix-Marseille Univerity) for funding.

in order to ensure robustness versus overfactoring problems, but also illustrate the interest of the second "full" model that is considered here. Our approaches are also compared with other "state-of-the-art" methods.

2. PROBLEM STATEMENT

2.1. Some recalls

A N-th order tensor can be represented by a N-way array. Its order corresponds to the number of indices of the associated array (*a.k.a ways* or *modes* [5]). Due to the considered fluorescence spectroscopy application, we focus on real nonnegative 3-way arrays denoted by $\mathbf{T} = (t_{ijk}) \in \mathbb{R}^{+I\times J\times K}$, admitting the following trilinear decomposition, *a.k.a* their triadic decomposition [14]:

$$\mathbf{T} = \sum_{f=1}^{F} \mathbf{a}_f \circledast \mathbf{b}_f \circledast \mathbf{c}_f, \tag{1}$$

where each $\mathbf{a}_f \circledast \mathbf{b}_f \circledast \mathbf{c}_f$ is a rank-1 third order array. The three following matrices $\mathbf{A} = (a_{if}) = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_F] \in \mathbb{R}^{I\times F}$, $\mathbf{B} = (b_{jf}) = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_F] \in \mathbb{R}^{J\times F}$, $\mathbf{C} = (c_{kf}) = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_F] \in \mathbb{R}^{K\times F}$ are the so-called *loading matrices*, whose columns are the *loading factors*. In our case the loading factors will have to be nonnegative since they represent nonnegative quantities which are respectively fluorescence emission, spectra, excitation spectra and concentrations. F is a sufficiently large integer corresponding to the number of components involved in the sum and $\circledast$ stands for the outer product. The smallest integer F that can be found such that Eq. (1) holds exactly is called the *tensor rank* [15]. For this value of F, the above decomposition is called the Canonical Polyadic decomposition (CP) of $\mathbf{T}$.

2.2. A modified model

As it can be more convenient to assume that all vectors $\mathbf{a}_f$, $\mathbf{b}_f$ and $\mathbf{c}_f$ have unit length, a modified model has been introduced [4]. Eq. (1) becomes $\mathbf{T} = \sum_{f=1}^{F} \lambda_f\, \mathbf{a}_f \circledast \mathbf{b}_f \circledast \mathbf{c}_f$, where λ_j are scaling factors and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_F]^T$, with $(\cdot)^T$ the matrix transposition operator.
Finally, this model can be written in a compact form using the Khatri-Rao product $\odot$, as

$$\mathbf{T}_{(1)}^{I,JK} = \mathbf{A}\boldsymbol{\Lambda}(\mathbf{C}\odot\mathbf{B})^T, \tag{2}$$

$$\mathbf{T}_{(2)}^{J,KI} = \mathbf{B}\boldsymbol{\Lambda}(\mathbf{C}\odot\mathbf{A})^T, \tag{3}$$

$$\mathbf{T}_{(3)}^{K,JI} = \mathbf{C}\boldsymbol{\Lambda}(\mathbf{B}\odot\mathbf{A})^T, \tag{4}$$

where $\mathbf{T}_{(1)}^{I,JK}$ (resp. $\mathbf{T}_{(2)}^{J,KI}$ and $\mathbf{T}_{(3)}^{K,JI}$) is the matrix of size $I\times JK$ (resp. $J\times KI$ and $K\times JI$) obtained by unfolding the array $\mathbf{T}$ of size $I\times J\times K$ in the first mode (resp. the second mode and the third mode); $\boldsymbol{\Lambda}$ is the $F\times F$ diagonal matrix defined as $\boldsymbol{\Lambda} = \mathsf{diag}\{\boldsymbol{\lambda}\}$ where the operator $\mathsf{diag}\{\cdot\}$ returns a square diagonal matrix which contains in its diagonal the elements of the vector given in argument.In a similar way, we define $\boldsymbol{\lambda} = \mathsf{Diag}\{\boldsymbol{\Lambda}\}$, where the operator $\mathsf{Diag}\{\cdot\}$ returns in a vector the diagonal elements of the the square diagonal matrix which is given in argument.

3. NONNEGATIVE 3-WAY ARRAY FACTORIZATION: NNCP DECOMPOSITION

The problem consists of estimating the three loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ but also $\boldsymbol{\Lambda}$ in the case of the modified model whilst ensuring their nonnegativity. A classical way to solve such a problem consists of coming back to a constrained minimization problem using a suitably designed cost function. But in [10], we used a square change of variables (if a matrix, say $\mathbf{A}'$, possesses only nonnegative terms, all its entries are defined as $a'_{ij} = a_{ij}^2$). Thus, the tensor decomposition problem is rewritten as an unconstrained minimization problem using the following cost function $\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\boldsymbol{\Lambda})$:

$$\begin{aligned}\mathcal{H} &= \|\mathbf{T}_{(1)}^{I,JK} - \mathbf{A}^{\boxdot 2}\boldsymbol{\Lambda}^{\boxdot 2}\left[\mathbf{C}^{\boxdot 2}\odot\mathbf{B}^{\boxdot 2}\right]^T\|_F^2 = \|\boldsymbol{\delta}_{(1)}\|_F^2\\ &= \|\mathbf{T}_{(2)}^{J,KI} - \mathbf{B}^{\boxdot 2}\boldsymbol{\Lambda}^{\boxdot 2}\left[\mathbf{C}^{\boxdot 2}\odot\mathbf{A}^{\boxdot 2}\right]^T\|_F^2 = \|\boldsymbol{\delta}_{(2)}\|_F^2\\ &= \|\mathbf{T}_{(3)}^{K,JI} - \mathbf{C}^{\boxdot 2}\boldsymbol{\Lambda}^{\boxdot 2}\left[\mathbf{B}^{\boxdot 2}\odot\mathbf{A}^{\boxdot 2}\right]^T\|_F^2 = \|\boldsymbol{\delta}_{(3)}\|_F^2,\end{aligned}$$

where $\mathbf{A}^{\boxdot 2} = \mathbf{A}\boxdot\mathbf{A}$, $\boxdot$ is the Hadamard product and $\|\cdot\|_F$ is the Frobenius norm. In the NNCP decomposition problem (like in most Blind Source Separation problems), the number of mixed compounds *i.e.* the rank F of the tensor is unknown. It can be either overestimated or estimated by different methods among which are Split Half Analysis [16], COre CONsistency DIAgnostic (CORDONDIA) [17], LTMC [18], Threshold-CORCONDIA [19], DIFFIT [20], SORTE [21], AROFAC2 [22] to mention only the most known. Yet, those estimation methods can be subject to forecast errors. That is why, it remains important to develop NNCP methods that are robust versus a possible overestimation of the tensor rank (*i.e.* overfactoring problems).

3.1. Gradient matrices

Using the same methodology as the one developed in [10], the four gradient components $\nabla_{\mathbf{A}}\mathcal{H}$, $\nabla_{\mathbf{B}}\mathcal{H}$, $\nabla_{\mathbf{C}}\mathcal{H}$ and $\nabla_{\boldsymbol{\Lambda}}\mathcal{H}$ can be derived:

$$\begin{aligned}\nabla_{\mathbf{A}}\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\boldsymbol{\Lambda}) &= \frac{\partial\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\boldsymbol{\Lambda})}{\partial\mathbf{A}}\\ &= 4\mathbf{A}\boxdot\left(-\boldsymbol{\delta}_{(1)}\left[\mathbf{C}^{\boxdot 2}\odot\mathbf{B}^{\boxdot 2}\right]\boldsymbol{\Lambda}^{\boxdot 2}\right),\\ \nabla_{\mathbf{B}}\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\boldsymbol{\Lambda}) &= \frac{\partial\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\boldsymbol{\Lambda})}{\partial\mathbf{B}}\\ &= 4\mathbf{B}\boxdot\left(-\boldsymbol{\delta}_{(2)}[\mathbf{C}^{\boxdot 2}\odot\mathbf{A}^{\boxdot 2}]\boldsymbol{\Lambda}^{\boxdot 2}\right),\end{aligned}$$

$$\nabla_{\mathbf{C}}\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{\Lambda}) = \frac{\partial\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{\Lambda})}{\partial\mathbf{C}} = 4\mathbf{C}\boxdot\left(-\boldsymbol{\delta}_{(3)}[\mathbf{B}^{\boxdot 2}\odot\mathbf{A}^{\boxdot 2}]\mathbf{\Lambda}^{\boxdot 2}\right)$$

$$\nabla_{\mathbf{\Lambda}}\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{\Lambda}) = \frac{\partial\mathcal{H}(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{\Lambda})}{\partial\mathbf{\Lambda}} = 4\mathbf{\Lambda}\boxdot\left(\mathbf{A}^{\boxdot 2}\right)^T\left(-\boldsymbol{\delta}_{(1)}\left[\mathbf{C}^{\boxdot 2}\odot\mathbf{B}^{\boxdot 2}\right]\right).$$

And we can build either the following $(I+J+K+F)\times F$ matrices $\mathbf{G}^{(k)}$ and $\mathbf{X}^{(k)}$ (standing resp. for gradient matrices and unknown parameters):

$$\mathbf{G}^{(k)} = \begin{pmatrix}\nabla_{\mathbf{A}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\\ \nabla_{\mathbf{B}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\\ \nabla_{\mathbf{C}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\\ \nabla_{\mathbf{\Lambda}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\end{pmatrix}, \mathbf{X}^{(k)} = \begin{pmatrix}\mathbf{A}^{(k)}\\ \mathbf{B}^{(k)}\\ \mathbf{C}^{(k)}\\ \mathbf{\Lambda}^{(k)}\end{pmatrix} \tag{5}$$

or more efficiently, the following $(I+J+K+1)F\times 1$ vectors:

$$\mathbf{g}^{(k)} = \begin{pmatrix}\mathsf{vec}\{\nabla_{\mathbf{A}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\}\\ \mathsf{vec}\{\nabla_{\mathbf{B}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\}\\ \mathsf{vec}\{\nabla_{\mathbf{C}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\}\\ \mathsf{Diag}\{\nabla_{\mathbf{\Lambda}}\mathcal{H}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)},\mathbf{\Lambda}^{(k)})\}\end{pmatrix} \tag{6}$$

$$\mathbf{x}^{(k)} = \begin{pmatrix}\mathsf{vec}\{\mathbf{A}^{(k)}\}\\ \mathsf{vec}\{\mathbf{B}^{(k)}\}\\ \mathsf{vec}\{\mathbf{C}^{(k)}\}\\ \boldsymbol{\lambda}^{(k)}\end{pmatrix} \tag{7}$$

3.2. Optimization algorithm: the non linear conjugate gradient algorithm

As described in [23] when applied to non quadratic problems, the non linear conjugate gradient (NLCG) method is more efficient than Newton methods especially for large systems of equations since it does not require to compute the Hessian matrix. The NLCG algorithm can take several forms, yet, we focus on line search methods. The complete NLCG algorithm (using restarts) thus reads:

Step 1. Given $\mathbf{x}^{(0)}$ (Eq. (7)), compute $\mathbf{g}^{(0)}$ (Eq. (6)) and set $\mathbf{d}^{(0)} = -\mathbf{g}^{(0)}$

Step 2. For $k = 0, 1, \ldots, n-1$

$$\begin{cases}\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mu^{(k)}\mathbf{d}^{(k)}\\ \mathbf{d}^{(k+1)} &= -\mathbf{g}^{(k+1)} + \beta^{(k)}\mathbf{d}^{(k)}\end{cases} \tag{8}$$

Step 3. If $k = n-1$ replace $\mathbf{x}^{(0)}$ by $\mathbf{x}^{(n)}$ and go back to **Step 1** (restart).

When the optimal step size is used, $\mu^{(k)}$ minimizes the polynomial $\mathcal{H}(\mathbf{P_1}\boxdot\mathbf{P_1},\mathbf{P_2}\boxdot\mathbf{P_2},\mathbf{P_3}\boxdot\mathbf{P_3},\mathbf{P_4}\boxdot\mathbf{P_4})$ where $\mathbf{P_1} = \mathbf{A}^{(k)} + \mu\mathbf{D}_{\mathbf{A}}^{(k)}$, $\mathbf{P_2} = \mathbf{B}^{(k)} + \mu\mathbf{D}_{\mathbf{B}}^{(k)}$, $\mathbf{P_3} = \mathbf{C}^{(k)} + \mu\mathbf{D}_{\mathbf{C}}^{(k)}$ and $\mathbf{P_4} = \mathbf{\Lambda}^{(k)} + \mu\mathbf{D}_{\mathbf{\Lambda}}^{(k)}$ with

$$\mathbf{D}^{(k)} = \begin{pmatrix}\mathbf{D}_{\mathbf{A}}^{(k)}\\ \mathbf{D}_{\mathbf{B}}^{(k)}\\ \mathbf{D}_{\mathbf{C}}^{(k)}\\ \mathbf{D}_{\mathbf{\Lambda}}^{(k)}\end{pmatrix}, \quad \mathbf{d}^{(k)} = \begin{pmatrix}\mathsf{vec}\{\mathbf{D}_{\mathbf{A}}^{(k)}\}\\ \mathsf{vec}\{\mathbf{D}_{\mathbf{B}}^{(k)}\}\\ \mathsf{vec}\{\mathbf{D}_{\mathbf{C}}^{(k)}\}\\ \mathsf{vec}\{\mathbf{D}_{\mathbf{\Lambda}}^{(k)}\}\end{pmatrix} = \begin{pmatrix}\mathbf{d}_{\mathbf{A}}^{(k)}\\ \mathbf{d}_{\mathbf{B}}^{(k)}\\ \mathbf{d}_{\mathbf{C}}^{(k)}\\ \mathbf{d}_{\mathbf{\Lambda}}^{(k)}\end{pmatrix} \tag{9}$$

In exact line searches methods, two expressions for β are classically used: the Fletcher-Reeves (β_{FR}) and the Polak-Ribière (β_{PR}) formula [24]:

$$\beta_{\mathsf{FR}}^{(k+1)} = \frac{\mathbf{g}^{(k+1)^T}\mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)^T}\mathbf{g}^{(k)}} \tag{10}$$

$$\beta_{\mathsf{PR}}^{(k+1)} = \frac{(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})^T\mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)^T}\mathbf{g}^{(k)}}. \tag{11}$$

Finally, as noticed in [23][24], the restarting aspect of the algorithm is fundamental for the global convergence analysis since in general one cannot guarantee that the directions $\mathbf{d}^{(i)}$ are effectively descent directions. But since a pure steepest descent step is taken at each restart by setting $\mathbf{d}^{(i)} = -\mathbf{g}^{(i)}$ the global convergence is assured. In our case, this "restart" is normally performed every $n-1 = (I+J+K+1)F$ iterations or each time $|\mathbf{g}^{(i)}| > \rho$ where ρ is a constant chosen equal to $10e+3$.

Finally, at each iteration, the columns of matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are rescaled to have unit l_1-norm, while the diagonal of the matrix $\mathbf{\Lambda}$ is multiplied by a certain scalar in order to maintain the cost function $\mathcal{H}$ unchanged.

3.3. Addition of regularization terms

The previous objective function $\mathcal{H}$, can be modified by adding regularization terms in order to ensure boundedness on the solution and/or to enforce another specific property on the solution such as smoothness, sparsity or uncorrelatedness. The modified objective function finally reads $\mathcal{G}(.) = \mathcal{H}(.) + \alpha_A\|\mathbf{A}^{\boxdot 2}\|_\bullet + \alpha_B\|\mathbf{B}^{\boxdot 2}\|_\bullet + \alpha_C\|\mathbf{C}^{\boxdot 2}\|_\bullet + \alpha_{\mathbf{\Lambda}}\|\mathbf{\Lambda}^{\boxdot 2}\|_\bullet$ where α_A, α_B, α_C and $\alpha_{\mathbf{\Lambda}}$ are nonnegative regularization parameters: they represent the weight of the regularized part. In order to avoid a bound in the performance, we suggest here to decrease these parameters $\alpha_\bullet$ every a certain number of iterations (we opted for each restart).

Different kinds of norms $\|\cdot\|_\bullet$ can be considered. The standard Tikhonov l_2-norm regularization ($\|\cdot\|_\bullet = \|\cdot\|_F^2$) is meant to enforce smoothness of the solution while the l_1-norm regularization ($\|\mathbf{C}\|_\bullet = \|\mathbf{C}\|_1 = \sum_{i,j}|c_{ij}|$) is meant to enforce sparsity of the solution. The gradient components given in (6)

or (5) are replaced by:

$$\nabla_{\mathbf{A}}\mathcal{G}(\cdot) = \nabla_{\mathbf{A}}\mathcal{H}(\cdot) + \alpha_A \star \tag{12}$$

$$\nabla_{\mathbf{B}}\mathcal{G}(\cdot) = \nabla_{\mathbf{B}}\mathcal{H}(\cdot) + \alpha_B \star\star \tag{13}$$

$$\nabla_{\mathbf{C}}\mathcal{G}(\cdot) = \nabla_{\mathbf{C}}\mathcal{H}(\cdot) + \alpha_C \star\star\star \tag{14}$$

$$\nabla_{\mathbf{\Lambda}}\mathcal{G}(\cdot) = \nabla_{\mathbf{\Lambda}}\mathcal{H}(\cdot) + \alpha_{\mathbf{\Lambda}} \star\star\star\star \tag{15}$$

where $\star = 2\mathbf{A}$ (resp. $2\mathbf{B}$, $2\mathbf{C}$ or $2\mathbf{\Lambda}$) is added in the case of a l_1-norm regularization, while $\star = 4\mathbf{A}^{\boxdot 3} = 4\mathbf{A} \boxdot \mathbf{A} \boxdot \mathbf{A}$ (resp. $4\mathbf{B}^{\boxdot 3}$, $4\mathbf{C}^{\boxdot 3}$ or $4\mathbf{\Lambda}^{\boxdot 3}$) is added in the case of a l_2-norm regularization.

4. COMPUTER SIMULATIONS

In this section, we test the NLCG method with different regularization approaches for both models ("full" modified and CP-model) then put in comparison with some reference approaches such as HALS [8] or Parafac [25]. In this example, to better assess the behavior of the different algorithms that are compared, the data have been numerically simulated. With regard to the constituent vectors of the loading matrices $\mathbf{A}$ and $\mathbf{B}$, we have used mixtures of Generalized Gaussian distribution functions, whereas the constituent vectors of the loading matrix $\mathbf{C}$ and the diagonal terms of $\mathbf{\Lambda}$ were simulated according to a uniform distribution. The sizes of the resulting third order nonnegative tensor T are $101 \times 47 \times 10$ and we assumed that $F = 4$ components were mixed. In the overestimated case, we suppose that $F = 5$. All algorithms are initialized thanks to the same arbitrary random initializations and the number of iterations is fixed and is equal to $10e + 5$ in all the tests that are performed.

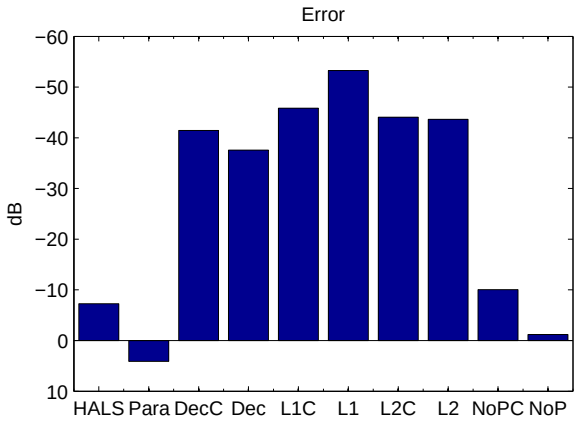

Fig. 1. Error discarding the overestimated part: a comparison of the results obtained with the different approaches: HALS, Parafac(Para), complete model and CP model with Decoupling regularization(DecC, Dec), l_1-norm regularization(L1C, L1) and Frobenius norm regularization(L2C, L2) respectively

By a permutation, we can suppose that the first F columns of the estimates $\widehat{\mathbf{A}}_{1:F}, \widehat{\mathbf{B}}_{1:F}, \widehat{\mathbf{C}}_{1:F}, \widehat{\mathbf{\Lambda}}_{1:F}$ correspond to the initial loading matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Lambda}$. We introduce then two indices of performance: the first one $\mathbf{E}_1$ measuring relative distance between the estimates and the true loading matrices when discarding the overestimated part

$$\mathbf{E}_1 = \frac{\|\widehat{\mathbf{A}}_{1:F} - \mathbf{A}\|_1 + \|\widehat{\mathbf{B}}_{1:F} - \mathbf{B}\|_1 + \|\widehat{\mathbf{C}}_{1:F} - \mathbf{C}\|_1 + \|\widehat{\mathbf{\Lambda}}_{1:F} - \mathbf{\Lambda}\|}{\|\mathbf{A}\|_1 + \|\mathbf{B}\|_1 + \|\mathbf{C}\|_1 + \|\mathbf{\Lambda}\|_1} \tag{16}$$

and the second one $\mathbf{E}_2$ related to the overestimated part

$$\mathbf{E}_2 = \sum_{f>F} \lambda_f. \tag{17}$$

Figure 1 shows better error estimations $\mathbf{E}_1$ of our proposed models and CP model in compare with HALS and Parafac. We also observe that the fact of adding regularisation deduces smaller errors than no-regularisation approaches. The best error estimation ($\mathbf{E}_1 =$) belongs to the l_1 NLCG method in CP model. Furthermore for the overestimate index, the modified model performs better than CP model with respect to each regularisation (Figure 2). We figure out the FEEM matrices (Figure 3) and $\mathbf{\Lambda}$ (Figure 4) for the readers' easy visualization.

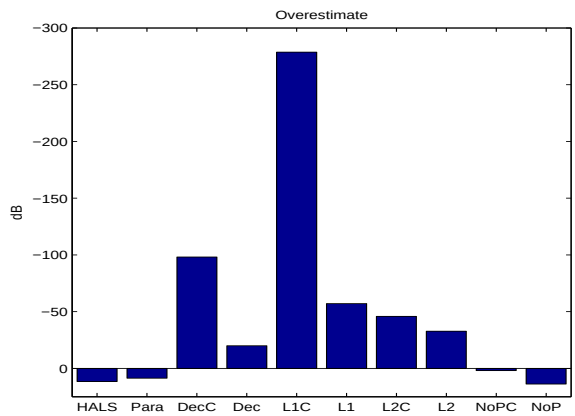

Fig. 2. Error related to the overestimated part: a comparison of the results obtained with the different approaches: HALS, Parafac(Para), complete model and CP model with Decoupling regularisation(DecC, Dec), l_1-norm regularisation(L1C,L1) and Frobenius norm regularisation(L2C, L2) respectively

5. CONCLUSION

In this article, focusing on synthetic examples sticking to the same kind of data as in the 3D fluorescence spectroscopy context, we have presented an alternative approach to handle

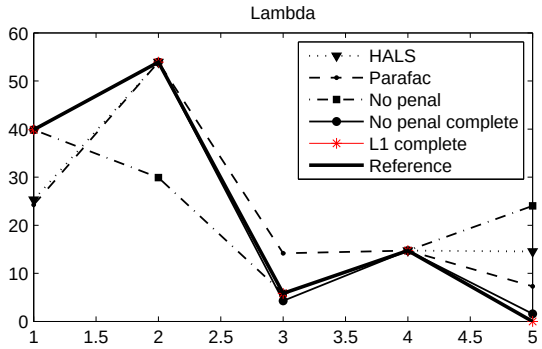

Fig. 3. Estimated values of λ versus the different methods

the NNCP decomposition of the resulting three-way tensor of FEEM. Computer simulations have been provided to emphasize the effectiveness of the suggested approaches. While more CPU time consuming, these latter clearly outperform more classical methods when overfactoring problems are encountered.

6. REFERENCES

[1] C. A. Stedmon and S. Markager. Behaviour of the optical properties of coloured dissolved organic matter under conservative mixing. *Estuarine, Coastal and Shelf Science*, 57(5-6):973–979, 2003.

[2] C. A. Stedmon, S. Markager, and R. Bro. Tracing dissolved organic matter in aquatic environments using a new approach to fluorescence spectroscopy. *Marine Chemistry*, 82:239–254, 2003.

[3] P. Carroll and J. J. Chang. Analysis of individual differences in multi-dimensional scaling via n-way generalization of eckart-young decomposition. *Psychometrika*, 35:283–319, 1970.

[4] R. A. Harshman. Foundation of the Parafac procedure: models and conditions for an explanatory multimodal factor analysis. *UCLA Working papers in phonetics*, 16:1–84, 1970.

[5] A. Smilde, R. Bro, and P. Geladi. *Multi-Way Analysis with applications in the chemical sciences*. Wiley, 2004.

[6] L. H. Lim and P. Comon. Nonnegative approximations of nonnegative tensors. *Jour. Chemometrics*, 23:432–441, August 2009.

[7] R. Bro and S. De Jong. A fast non–negativity–constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1997.

[8] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari. *Non negative matrix and tensor factorizations: Application to exploratory multi-way data analysis and blind separation*. Wiley, 2009.

[9] P. Paatero. A weighted non-negative least squares algorithm for three-way Parafac factor analysis. *Chemometrics Intell. Lab. Systems*, 38(2):223–242, 1997.

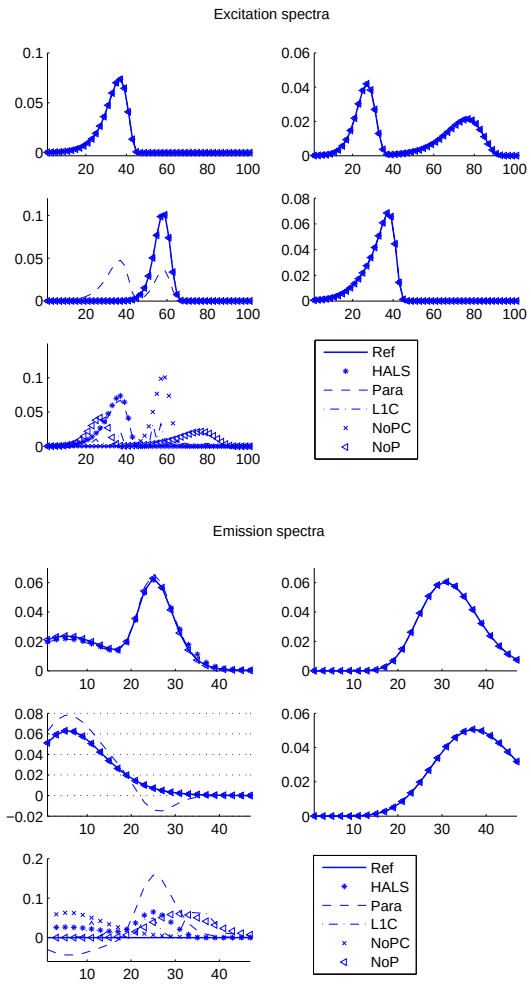

Fig. 4.

[10] J.-P. Royer, N. Thirion-Moreau, and P. Comon. Computing the polyadic decomposition of nonnegative third order tensors. *Eurasip Signal Processing*, 91(9):2159–2171, September 2011.

[11] J.-P. Royer, P. Comon, and N. Thirion-Moreau. Computing the nonnegative 3-way tensor factorization using tikhonov regularization. In *International Conference on Acoustic Speech and Signal Processing (ICASSP'2011)*, pages 2732–2735, Prague, Czech Republic, May 2009.

[12] D. Bunker, L. Han, and S. Zhang. A proximal anls algorithm for nonnegative tensor factorization with a periodic enhanced line search. *Application of Mathematics*, 58(5):493–509, October 2013.

[13] J. Kim and H. Park. Fast nonnegative tensor factorization with an active-set-like method. *High-Performance Scientific Computing: Algorithms and Applications, Springer*, pages 311–326, 2012.

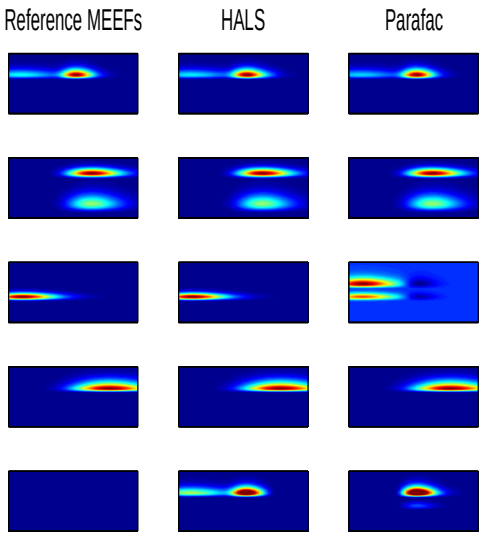

Fig. 5.

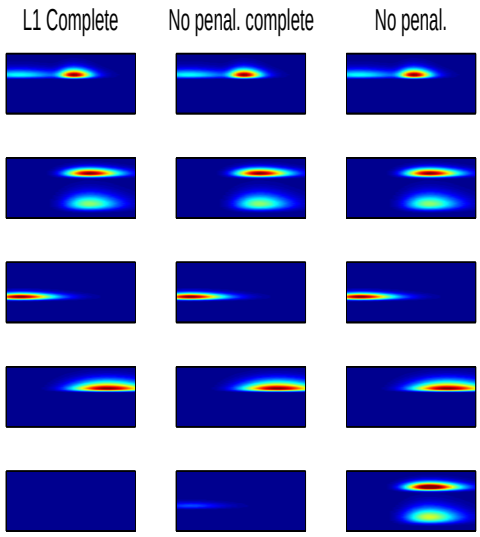

Fig. 6.

[14] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *J. Math. and Phys.*, 6:165–189, 1927.

[15] T. Lickteig. Typical tensorial rank. *Linear Algebra Appl.*, 69:95–120, 1985.

[16] R. A. Harshman and W. S. De Sarbo. An application of parafac to a small sample problem, demonstrating preprocessing, orthogonality constraints, and split-half diagnostic techniquess. *Statistique et Analyse des Donnes*, 13:14–32, 1984.

[17] R. Bro and H. A. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of chemometrics*, 17:274–286, 2003.

[18] L.-Q. Hu, H.-L. Wu, J.-H. Jiang, Q.-J. Han, A.-L. Xia, and R.-Q. Yu. Estimating the chemical rank of three-way data arrays by a simple linear transform incorporating monte carlo simulation. *Talanta*, 71:373–380, 2007.

[19] J. Da Costa, M. Haardt, and F. Roemer. Robust methods based on the hosvd for estimating the model order in parafac models. In IEEE, editor, *Sensor array and multichannel signal processing workshop (SAM)*, pages 510–514, 2008.

[20] M. E. Timmerman and H. A. Kiers. Three-mode principal components analysis: choosing the number of components and sensitivity to local optima. *British journal of mathematical and statistical psychology*, 53(1):1–16, 2000.

[21] Z. He, A. Cichocki, S; Xie, and K. Choi. Detecting the number of clusters in n-way probabilistic clustering. *IEEE transaaction on pattern analysis and machine intelligence*, 32(11):2006–2021, 2010.

[22] F. Kirly and A. Ziehe. Approximate rank-detecting factorization of low-rank tensors. In *in International Conference on Acoustic Speech and Signal Processing (ICASSP'2013)*, Vancouver, Canada, 2013.

[23] D. G. Luenberger and Y. Ye. *Linear and non linear programming*. Wiley, third edition, 2008.

[24] E. Polak. *Optimization algorithms and consistent approximations*. Springer, 1997.

[25] R. Bro and C. A. Andersson. *N-Way Toolbox for Matlab*. http://www.models.life.ku.dk/nwaytoolbox.