# An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA.

Dimitri Nion, Lieven De Lathauwer [1]

*ETIS Laboratory, CNRS UMR 8051, 6, avenue du Ponceau, 95014 Cergy-Pontoise, France. Tel/Fax: +33 130 736 610/ +33 130 736 627*
*E-mail: {nion,delathau}@ensea.fr*

**Abstract**

In this paper, we introduce an enhanced Line Search algorithm to accelerate the convergence of the Alternating Least Squares (ALS) algorithm, which is often used to decompose a tensor in a sum of contributions. This scheme can be used for the computation in the complex case of the Parallel Factor model or the more general Block Component Model. We then illustrate the performance of the algorithm in the context of blind separation-equalization of convolutive DS-CDMA mixtures.

*Key words:* Tensor Decompositions, Parallel Factor Model, Block Component Model, Alternating Least Squares, Line Search, Code Division Multiple Access

## 1 Introduction

An increasing number of problems in signal processing, data analysis and scientific computing involves the manipulation of quantities of which the elements are addressed by more than two indices [1]. In the literature, these higher-order analogues of vectors (first order) and matrices (second order) are called higher-order tensors, multidimensional matrices or multiway arrays.

[1] This work is supported in part by the French Délégation Générale pour l'Armement (DGA), in part by the Research Council K.U.Leuven under Grant GOA-AMBioRICS, CoE EF/05/006 Optimization in Engineering, in part by the Flemish Government under F.W.O. Project G.0321.06, F.W.O. research communities ICCoS, ANMMM and MLDM, in part by the Belgian Federal Science Policy Office under IUAP P6/04, and in part by the E.U.: ERNSI.

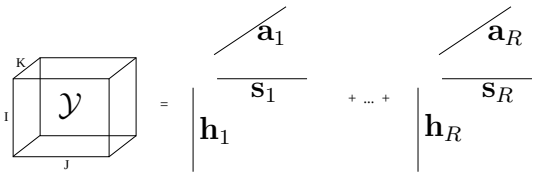



Figure 1. Schematic representation of the PARAFAC model

Key to the development of algorithms is the computation of tensor decompositions. We briefly introduce the decompositions used in the PARAllel FACtor (PARAFAC) model and the more general Block Component Model (BCM). For the definition of PARAFAC, we first need to define the tensor outer product.

**Definition 1.1. (Outer-product)** *The outer product of 3 vectors,* $\mathbf{h} \in \mathbb{C}^{I\times 1}$, $\mathbf{s} \in \mathbb{C}^{J\times 1}$ *and* $\mathbf{a} \in \mathbb{C}^{K\times 1}$, *denoted by* $(\mathbf{h} \circ \mathbf{s} \circ \mathbf{a})$, *is an* $(I \times J \times K)$ *tensor with elements defined by* $(\mathbf{h} \circ \mathbf{s} \circ \mathbf{a})_{ijk} = h_i s_j a_k$.

This definition immediately allows us to define rank-1 tensors.

**Definition 1.2. (Rank-1 tensor)** *A third-order tensor* $\mathcal{Y}$ *has rank 1 if it equals the outer product of 3 vectors.*

We are now in a position to formally define PARAFAC.

**Definition 1.3. (PARAFAC)** *A Canonical or a PARAFAC Decomposition of a third-order tensor* $\mathcal{Y} \in \mathbb{C}^{I\times J\times K}$, *represented in Fig. 1, is a decomposition of* $\mathcal{Y}$ *as a linear combination of a minimal number of rank-1 tensors:*

$$\mathcal{Y} = \sum_{r=1}^{R} \mathbf{h}_r \circ \mathbf{s}_r \circ \mathbf{a}_r, \tag{1}$$

*where* $\mathbf{h}_r$, $\mathbf{s}_r$, $\mathbf{a}_r$ *are the* $r^{th}$ *columns of matrices* $\mathbf{H} \in \mathbb{C}^{I\times R}$, $\mathbf{S} \in \mathbb{C}^{J\times R}$ *and* $\mathbf{A} \in \mathbb{C}^{K\times R}$.

This trilinear model was independently introduced in Psychometrics [2] and Phonetics [3]. More recently, the decomposition found applications in Chemometrics [4] and Independent Component Analysis (ICA) [1, 5]. The authors of [6] were the first to use this multilinear algebra technique in the context of wireless communications. They proposed a blind PARAFAC-based receiver for instantaneous CDMA mixtures impinging on an antenna array.

However, in several applications, the inherent algebraic structure of the tensor of observations $\mathcal{Y}$ might result from contributions that are not rank-1 tensors. This more general situation is covered by the BCM, introduced in [7,8].

For the definition of BCM, we first need to define the mode-n product of a tensor and a matrix.

**Definition 1.4. (Mode-n product)** *The mode-2 and mode-3 products of a third-order tensor* $\mathcal{H} \in \mathbb{C}^{I\times L\times P}$ *by the matrices* $\mathbf{S} \in \mathbb{C}^{J\times L}$ *and* $\mathbf{A} \in \mathbb{C}^{K\times P}$, *respectively denoted by* $\mathcal{H} \bullet_2 \mathbf{S}$ *and* $\mathcal{H} \bullet_3 \mathbf{A}$, *result in an* $(I \times J \times P)$*-tensor,*

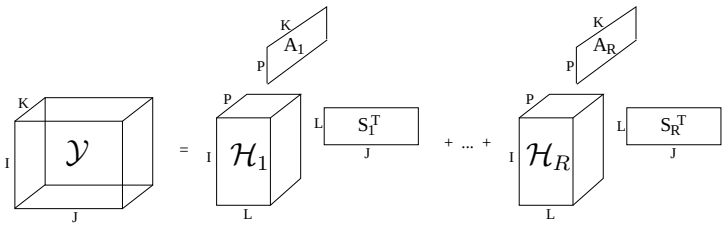



Figure 2. Schematic representation of the BCM

*$(I \times L \times K)$-tensor respectively, with elements defined, for all index values, by*

$$(\mathcal{H} \bullet_2 \mathbf{S})_{ijp} = \sum_{l=1}^{L} h_{ilp} s_{jl} \quad , \quad (\mathcal{H} \bullet_3 \mathbf{A})_{ilk} = \sum_{p=1}^{P} h_{ilp} a_{kp}.$$

We now have the following definition.

**Definition 1.5. (BCM)** *A third-order tensor $\mathcal{Y} \in \mathbb{C}^{I \times J \times K}$ follows a Block Component Model (BCM) if it can be written as follows:*

$$\mathcal{Y} = \sum_{r=1}^{R} \mathcal{H}_r \bullet_2 \mathbf{S}_r \bullet_3 \mathbf{A}_r. \tag{2}$$

*The vectors $\mathbf{h}_r \in \mathbb{C}^{I \times 1}$, $\mathbf{s}_r \in \mathbb{C}^{J \times 1}$ and $\mathbf{a}_r \in \mathbb{C}^{K \times 1}$ of the PARAFAC model are now replaced by a tensor $\mathcal{H}_r \in \mathbb{C}^{I \times L \times P}$ and two matrices $\mathbf{S}_r \in \mathbb{C}^{J \times L}$ and $\mathbf{A}_r \in \mathbb{C}^{K \times P}$, respectively.*

A schematic representation of the BCM is given in Fig. 2. In [10], this generalization of PARAFAC was used to model convolutive CDMA mixtures received by an antenna array. An equivalent but formally different formulation was given in [11]. The standard way to compute the PARAFAC decomposition is the Alternating Least Squares (ALS) algorithm [4]. In [9, 10], this algorithm has been generalized to compute the BCM decomposition. However, it is sensitive to swamps (i.e., many iterations with convergence speed almost null after which convergence resumes) and thus sometimes needs a very large number of iterations to converge. In [3, 12], Line Search was proposed to speed up convergence of ALS for PARAFAC. A remarkable result has been obtained in [13], where the authors have shown that, for *real*-valued tensors that follow the PARAFAC model, the *optimal* step size can be calculated. This method is called "Enhanced Line Search" (ELS).

In this paper, we propose a new Line Search scheme for *both* PARAFAC and BCM decompositions of *complex*-valued tensors. The so-called "Enhanced Line Search with Complex Step" (ELSCS) is performed before each ALS iteration. It consists of looking for the optimal step size in $\mathbb{C}$. A preliminary version of this paper appeared as the conference paper [14].

## 2 Enhanced Line Search in the complex case

Given only $\mathcal{Y}$, the computation of the BCM decomposition consists in the estimation of $\mathcal{H}_r$, $\mathbf{S}_r$ and $\mathbf{A}_r$, $r = 1 \dots R$. We first formulate the computation as the minimization of a quadratic cost function. Denote by $\mathbf{A}$ and $\mathbf{S}$ the $K \times RP$ and $J \times RL$ matrices that result from the concatenation of the $R$ matrices $\mathbf{A}_r$ and $\mathbf{S}_r$, respectively, and by $\mathbf{H}$ the $I \times RLP$ matrix in which the entries of the tensors $\mathcal{H}_r$ are stacked as follows: $[\mathbf{H}]_{i,(r-1)LP+(l-1)P+p} = \mathcal{H}_r(i,l,p)$.

Let $\mathbf{Y}^{(JK\times I)}$ be the $JK \times I$ matrix representation of $\mathcal{Y}$, with elements defined as follows: $[\mathbf{Y}^{(JK\times I)}]_{(j-1)K+k,i} = y_{ijk}$. Let $\otimes$ denote the Kronecker product, $\|\cdot\|_F$ the Frobenius norm and $\hat{\mathcal{Y}}$ an estimate of $\mathcal{Y}$, built from the estimated factors $\hat{\mathbf{A}}$, $\hat{\mathbf{S}}$ and $\hat{\mathbf{H}}$. The calculation of the BCM decomposition now consists of the minimization of the following cost function

$$\phi = \left\|\mathcal{Y} - \hat{\mathcal{Y}}\right\|_F^2 = \left\|\mathbf{Y}^{(JK\times I)} - (\hat{\mathbf{S}} \odot_R \hat{\mathbf{A}}) \cdot \hat{\mathbf{H}}^T\right\|_F^2, \tag{3}$$

where the partition-wise Kronecker product $\odot_R$ of the matrices $\hat{\mathbf{S}} \in \mathbb{C}^{J\times RL}$ and $\hat{\mathbf{A}} \in \mathbb{C}^{K\times RP}$, results in a $JK \times RLP$ matrix defined by $\hat{\mathbf{S}} \odot_R \hat{\mathbf{A}} = [\hat{\mathbf{S}}_1 \otimes \hat{\mathbf{A}}_1 | \dots | \hat{\mathbf{S}}_R \otimes \hat{\mathbf{A}}_R]$. For the PARAFAC decomposition, $L = P = 1$ so the estimation of the matrices $\mathbf{H} \in \mathbb{C}^{I\times R}$, $\mathbf{S} \in \mathbb{C}^{J\times R}$ and $\mathbf{A} \in \mathbb{C}^{K\times R}$ is done by the minimization of the same cost function except that $\odot_R$ is replaced by $\odot$, which is the Khatri-Rao product, or column-wise Kronecker product. Hence, the ELSCS scheme proposed in the following works both for PARAFAC and BCM. Note that $\mathcal{Y}$ is multi-linear in $\mathbf{S}$, $\mathbf{A}$, $\mathbf{H}$. The ALS algorithm exploits the multilinearity of PARAFAC/BCM by minimizing $\phi$ alternately w.r.t. the unknowns $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{H}$ in each iteration. Explicit formulation for the ALS algorithm is given in [4, 12] for PARAFAC and in [9, 10] for BCM.

For PARAFAC, it was noticed through simulations that, when the convergence of the ALS algorithm is slow, $\hat{\mathbf{A}}$, $\hat{\mathbf{S}}$ and $\hat{\mathbf{H}}$ are gradually incremented along fixed directions. Consequently, Line Search was proposed to speed up the convergence in [3,12]. The procedure consists of the linear interpolation of the unknown factors from their previous estimates:

$$\begin{cases} \hat{\mathbf{A}}^{(new)} = \hat{\mathbf{A}}^{(n-2)} + \rho\left(\hat{\mathbf{A}}^{(n-1)} - \hat{\mathbf{A}}^{(n-2)}\right) \\ \hat{\mathbf{S}}^{(new)} = \hat{\mathbf{S}}^{(n-2)} + \rho\left(\hat{\mathbf{S}}^{(n-1)} - \hat{\mathbf{S}}^{(n-2)}\right) \\ \hat{\mathbf{H}}^{(new)} = \hat{\mathbf{H}}^{(n-2)} + \rho\left(\hat{\mathbf{H}}^{(n-1)} - \hat{\mathbf{H}}^{(n-2)}\right) \end{cases}, \tag{4}$$

where $\hat{\mathbf{A}}^{(n-1)}$, $\hat{\mathbf{S}}^{(n-1)}$ and $\hat{\mathbf{H}}^{(n-1)}$ are the estimates of $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{H}$, respectively, obtained from the $(n-1)^{th}$ ALS iteration. The known matrices $\mathbf{G}_A^{(n)} =$

$(\hat{\mathbf{A}}^{(n-1)} - \hat{\mathbf{A}}^{(n-2)})$, $\mathbf{G}_S^{(n)} = (\hat{\mathbf{S}}^{(n-1)} - \hat{\mathbf{S}}^{(n-2)})$ and $\mathbf{G}_H^{(n)} = (\hat{\mathbf{H}}^{(n-1)} - \hat{\mathbf{H}}^{(n-2)})$ represent the search directions in the $n^{th}$ iteration and $\rho$ is the relaxation factor, i.e., the step size in the search directions. This Line Search step is performed before each ALS iteration and the interpolated matrices $\hat{\mathbf{A}}^{(new)}$, $\hat{\mathbf{S}}^{(new)}$ and $\hat{\mathbf{H}}^{(new)}$ are then used to start the $n^{th}$ iteration of the ALS. The challenge of Line Search is to find a "good" step size in the search directions in order to speed up convergence. In [3], the step size $\rho$ is given a fixed value (between 1.2 and 1.3). In [12] $\rho$ is set to $n^{1/3}$ and the Line Search step is accepted only if the interpolated value of the loss function is less than its current value. For *real*-valued tensors, the ELS technique [13] calculates the *optimal* step size by rooting a polynomial.

However, in several applications [6,10], the data are complex-valued. We therefore propose to generalize the ELS algorithm to the complex case, i.e., we look for the optimal step $\rho$ in $\mathbb{C}$. The new scheme is called Enhanced Line Search with Complex Step (ELSCS).

Combination of (3) and (4) shows that, given the estimates of $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{H}$ at iterations $(n-1)$ and $(n-2)$, the optimal relaxation factor $\rho$ at iteration $n$ is found by minimization of:

$$\begin{aligned}\phi_{ELSCS}^{(n)} &= \left\| \left( \hat{\mathbf{S}}^{(new)} \odot_R \hat{\mathbf{A}}^{(new)} \right) \cdot \hat{\mathbf{H}}^{(new)^T} - \mathbf{Y}^{(JK \times I)} \right\|_F^2 \\ &= \left\| \left( (\hat{\mathbf{S}}^{(n-2)} + \rho \mathbf{G}_S^{(n)}) \odot_R (\hat{\mathbf{A}}^{(n-2)} + \rho \mathbf{G}_A^{(n)}) \right) \cdot \right. \\ &\quad \left. \left( \hat{\mathbf{H}}^{(n-2)} + \rho \mathbf{G}_H^{(n)} \right)^T - \mathbf{Y}^{(JK \times I)} \right\|_F^2 . \end{aligned} \tag{5}$$

It is a matter of technical formula manipulations to show that this equation can also be written as follows:

$$\phi_{ELSCS}^{(n)} = \left\| \rho^3 \mathbf{T}_3 + \rho^2 \mathbf{T}_2 + \rho \mathbf{T}_1 + \mathbf{T}_0 \right\|_F^2 , \tag{6}$$

in which the $JK \times I$ known matrices $\mathbf{T}_3$, $\mathbf{T}_2$, $\mathbf{T}_1$ and $\mathbf{T}_0$ are defined by

$$\begin{cases} \mathbf{T}_3 = & (\mathbf{G}_S \odot_R \mathbf{G}_A)\mathbf{G}_H^T \\ \mathbf{T}_2 = & (\mathbf{S} \odot_R \mathbf{G}_A + \mathbf{G}_S \odot_R \mathbf{A})\mathbf{G}_H^T + (\mathbf{G}_S \odot_R \mathbf{G}_A)\mathbf{H}^T \\ \mathbf{T}_1 = & (\mathbf{S} \odot_R \mathbf{A})\mathbf{G}_H^T + (\mathbf{S} \odot_R \mathbf{G}_A + \mathbf{G}_S \odot_R \mathbf{A})\mathbf{H}^T \\ \mathbf{T}_0 = & (\mathbf{S} \odot_R \mathbf{A})\mathbf{H}^T - \mathbf{Y}^{(JK \times I)} \end{cases} ,$$

where the superscripts $n$ and $n-2$ have been omitted for convenience of notation. We repeat that the goal is the computation of the optimal $\rho$ from

the minimization of (6). Denote by $Vec$ the operator that writes a matrix $\mathbf{A} \in \mathbb{C}^{I \times J}$ in vector format by concatenation of the columns such that $\mathbf{A}(i,j) = [Vec(\mathbf{A})]_{i+(j-1)I}$. Eq. (6) is then equivalent to:

$$\phi^{(n)}_{ELSCS} = \|\mathbf{T} \cdot \mathbf{u}\|^2_F = \mathbf{u}^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{u}, \tag{7}$$

where $\mathbf{T} = [Vec(\mathbf{T}_3)|\, Vec(\mathbf{T}_2)|\, Vec(\mathbf{T}_1)|\, Vec(\mathbf{T}_0)]$ is an $IJK \times 4$ matrix, $\mathbf{u} = [\rho^3, \rho^2, \rho, 1]^T$ and $.^H$ denotes the Hermitian transpose. The $(4 \times 4)$ matrix $\Delta = \mathbf{T}^H \cdot \mathbf{T}$ has complex elements defined by $[\Delta]_{m,n} = \alpha_{m,n} + j\beta_{m,n}$. Since $\Delta$ is Hermitian, $\alpha_{m,n} = \alpha_{n,m}$, $\beta_{m,n} = -\beta_{n,m}$ and $\beta_{m,m} = 0$. For real-valued data, the cost function (7) is equivalent to $\phi^{(n)}_{ELSCS} = \mathbf{u}^T \cdot \mathbf{T}^T \cdot \mathbf{T} \cdot \mathbf{u}$. This is a polynomial of degree six in the real variable $\rho$ and can thus easily be minimized [13].

The case of complex-valued data is more difficult. We write the relaxation factor as $\rho = r.e^{i\theta}$, where $r$ is the modulus of $\rho$ and $\theta$ its argument, and propose an iterative scheme that minimizes $\phi^{(n)}_{ELSCS}$ by alternating between updates of $r$ and $\theta$. The complexity of the latter iteration is fairly low compared to the ALS iteration, since updating $r$ and $\theta$ consists of rooting two polynomials of degree five and six, respectively.

The partial derivative of $\phi^{(n)}_{ELSCS}$ w.r.t. $r$ can be expressed as:

$$\frac{\delta \phi^{(n)}_{ELSCS}(r)}{\delta r} = \sum_{p=0}^{5} c_p r^p, \tag{8}$$

where the real coefficients $c_p$ are given in Appendix. Given the last update of $\theta$, the update of $r$ thus consists of finding the real roots of a polynomial of degree five and selecting the root that minimizes $\phi^{(n)}_{ELSCS}(r)$.

After a change of variable, $t = tan(\frac{\theta}{2})$, the partial derivative of $\phi^{(n)}_{ELSCS}$ w.r.t. $t$ can be expressed as:

$$\frac{\delta \phi^{(n)}_{ELSCS}(t)}{\delta t} = \frac{\sum_{p=0}^{6} d_p t^p}{(1+t^2)^3}, \tag{9}$$

where the real coefficients $d_p$ are given in Appendix. Given the last update of $r$, the update of $\theta$ consists of finding the real roots of a polynomial of degree six and selecting the root that minimizes $\phi^{(n)}_{ELSCS}(t)$. The ELSCS scheme is then inserted in the standard ALS algorithm.

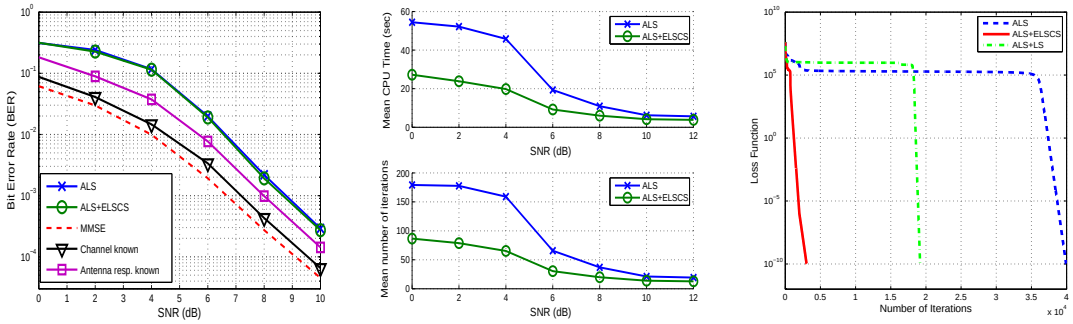
**ALS+ELSCS algorithm**

Initialize $\hat{\mathbf{H}}^{(0)}$, $\hat{\mathbf{H}}^{(1)}$, $\hat{\mathbf{S}}^{(0)}$, $\hat{\mathbf{S}}^{(1)}$, $\hat{\mathbf{A}}^{(0)}$, $\hat{\mathbf{A}}^{(1)}$, set $n = 1$;
**while** $\|\hat{\mathcal{Y}}^{(n)} - \hat{\mathcal{Y}}^{(n-1)}\|_F > \epsilon_1$ *(e.g.* $\epsilon_1 = 10^{-6}$*)* **do**
  - $n \leftarrow n + 1$;
  —— *Start ELSCS scheme*——
  - Set $p = 1$;
  **while** $|\phi^{(p)}_{ELSCS} - \phi^{(p-1)}_{ELSCS}| > \epsilon_2$ *(e.g.* $\epsilon_2 = 10^{-4}$*)* **do**
    - update $r$ from (8) with $\theta$ fixed;
    - update $\theta$ from (9) with $r$ fixed;
    - $p \leftarrow p + 1$;
  **end**
  - Build $\hat{\mathbf{A}}^{(new)}$, $\hat{\mathbf{S}}^{(new)}$ and $\hat{\mathbf{H}}^{(new)}$ from (4);
  —— *Start ALS updates*——
  - Find $\hat{\mathbf{S}}^{(n)}$ from $\hat{\mathbf{H}}^{(new)}$ and $\hat{\mathbf{A}}^{(new)}$;
  - Find $\hat{\mathbf{H}}^{(n)}$ from $\hat{\mathbf{A}}^{(new)}$ and $\hat{\mathbf{S}}^{(n)}$;
  - Find $\hat{\mathbf{A}}^{(n)}$ from $\hat{\mathbf{S}}^{(n)}$ and $\hat{\mathbf{H}}^{(n)}$;
  - Build $\hat{\mathcal{Y}}^{(n)}$ from $\hat{\mathbf{S}}^{(n)}$, $\hat{\mathbf{H}}^{(n)}$ and $\hat{\mathbf{A}}^{(n)}$;
**end**

## 3 Simulations Results

In [10] we used the BCM to solve the problem of *blind separation-equalization* of *convolutive* DS-CDMA mixtures received by an antenna array after multipath propagation. We assume that the signal of the $r^{th}$ user is subject to Inter-Symbol-Interference (ISI) over $L$ consecutive symbols and that this signal arrives at the antenna array via $P$ specular paths. For user $r$, $r = 1 \dots R$, the $I \times L$ frontal slice $\mathcal{H}_r(:,:,p)$ of $\mathcal{H}_r$ then collects samples of the convolved spreading waveform associated to the $p^{th}$ path, $p = 1 \dots P$. The $J \times L$ matrix $\mathbf{S}_r$ holds the $J$ transmitted symbols and has a Toeplitz structure. The $K \times P$ matrix $\mathbf{A}_r$ collects the response of the $K$ antennas according to the angles of arrival of the $P$ paths.

In this section, we illustrate the improvement of performance allowed by the ELSCS scheme, compared to the simple ALS algorithm. We consider $R = 4$ users, pseudo random spreading codes of length $I = 8$, a short frame of $J = 50$ QPSK symbols, $K = 4$ antennas, $L = 2$ interfering symbols and $P = 2$ paths per user. In Figs. 3(a) and 3(b), we give the results of 1000 Monte Carlo trials. The Signal to Noise Ratio at the input of the BCM receiver is defined by $SNR = 10\log_{10}(\|\mathcal{Y}\|_F^2/\|\mathcal{N}\|_F^2)$, where $\mathcal{Y}$ is the complex-valued noise-free tensor of observations and the tensor $\mathcal{N}$ holds zero-mean white (in all dimensions) Gaussian Noise. For each Monte-Carlo trial, the algorithms are initialized with 10 different random starting points and the performance is evaluated after selection of the best initialization (the one that leads to minimal value of $\phi$). Fig. 3(a) shows the average Bit Error Rate (BER) over all




(a) BER vs. SNR (b) Mean CPU time and number of iterations vs SNR (c) $\phi$ vs. number of iterations

Figure 3. Performance of standard ALS algorithm vs. ALS+ELSCS algorithm

users versus SNR, for the BCM receiver based either on ALS or ALS+ELSCS. The performance of the (non-blind) MMSE receiver and of two semi-blind receivers assuming either the antenna array response known or the channel known is also given. The ALS and ALS+ELSCS curves coincide, which means that they converge to the same point, on the average. However, the mean number of initializations required (out of 10) to obtain these two curves was 6.6 for ALS and 3.4 for ALS+ELSCS which illustrates the better capacity of the latter algorithm to reach the global minimum.

Fig. 3(b) shows the mean number of iterations and the mean CPU time required by ALS and ALS+ELSCS. The ELSCS scheme allowed to considerably reduce the number of iterations; moreover the extra cost per iteration step was negligible since the time to converge has been reduced in the same proportion as the number of iterations.

Fig. 3(c) shows typical curves for ill-conditioned data. We compare the evolution of the cost function $\phi$ for ALS, for ALS+LS with $\rho = n^{1/3}$ as in [12] and ALS+ELSCS. In this test, the data are noise-free. The matrix $\mathbf{A}$ has been built such that its highest singular value is equal to 100 and the other singular values to 1. We kept the best initialization among 10 different random starting points. The stop criterion is $\phi < 10^{-10}$. We observe that the LS scheme reduces the number of iterations from $4.10^4$ to $2.10^4$. In the same conditions, the ALS+ELSCS algorithm escapes from the swamp quickly since it only requires $3.10^3$ iterations.

## 4 Conclusion

We have presented an Enhanced Line Search algorithm for the decomposition of complex-valued tensors that follows the PARAFAC model or the Block

Component Model. This scheme looks for the optimal step in $\mathbb{C}$, and thus allows to escape quickly from swamps that might occur when the complex data are ill-conditioned. As a result, the ELSCS scheme inherits the advantages of its real-valued counterpart and remarkably improves the convergence speed of the standard ALS algorithm.

## APPENDIX

**Derivation of the coefficients $c_p$ in Eq. (8)**: from Eq. (7), $\phi_{ELSCS}$ can be written as a polynomial of degree six, $\phi_{ELSCS}(r) = \sum_{p=0}^{6} x_p r^p$, where the coefficients $x_p$ only depend on $\theta$ and the coefficients of $\Delta$:

$$\begin{cases} x_6 = & \alpha_{11} \\ x_5 = & 2\alpha_{12}\cos(\theta) + 2\beta_{12}\sin(\theta) \\ x_4 = & \alpha_{22} + 2\alpha_{13}\cos(2\theta) + 2\beta_{13}\sin(2\theta) \\ x_3 = & 2\alpha_{14}\cos(3\theta) + 2\alpha_{23}\cos(\theta) + 2\beta_{14}\sin(3\theta) + 2\beta_{23}\sin(\theta) \\ x_2 = & \alpha_{33} + 2\alpha_{24}\cos(2\theta) + 2\beta_{24}\sin(2\theta) \\ x_1 = & 2\alpha_{34}\cos(\theta) + 2\beta_{34}\sin(\theta) \\ x_0 = & \alpha_{44} \end{cases}$$

The coefficients $c_p$ in (8) are thus given by $c_p = (p+1)x_{p+1}$.

**Derivation of the coefficients $d_p$ in Eq. (9)**: from Eq. (7), $\phi_{ELSCS}$ can also be written under the following form:

$$\phi_{ELSCS}(\theta) = a_1\cos(3\theta) + a_2\cos(2\theta) + a_3\cos(\theta) + a_4 + b_1\sin(3\theta) + b_2\sin(2\theta) + b_3\sin(\theta),$$

where the coefficients $a_i$ and $b_j$, only depend on $r$ and the coefficients of $\Delta$:

$$\begin{cases} a_1 = & 2r^3\alpha_{14} \\ a_2 = & 2r^4\alpha_{13} + 2r^2\alpha_{24} \\ a_3 = & 2r^5\alpha_{12} + 2r^3\alpha_{23} + 2r\alpha_{34} \\ a_4 = & r^6\alpha_{11} + r^4\alpha_{22} + r^2\alpha_{33} + \alpha_{44} \\ b_1 = & 2r^3\beta_{14} \\ b_2 = & 2r^4\beta_{13} + 2r^2\beta_{24} \\ b_3 = & 2r^5\beta_{12} + 2r^3\beta_{23} + 2r\beta_{34} \end{cases}$$

We thus have:

$$\frac{\delta\phi_{ELSCS}(\theta)}{\delta\theta} = -3a_1\sin(3\theta) - 2a_2\sin(2\theta) - a_3\sin(\theta) + 3b_1\cos(3\theta) + 2b_2\cos(2\theta) + b_3\cos(\theta)$$

After the change of variable $t = tan(\frac{\theta}{2})$, and the substitution $\cos(\theta) = \frac{1-t^2}{1+t^2}$

and $\sin(\theta) = \frac{2t}{1+t^2}$ we obtain:

$$\frac{\delta \phi_{ELSCS}(\theta)}{\delta \theta} = \frac{\sum_{p=0}^{6} d_p t^p}{(1+t^2)^3},$$

where the coefficients $d_p$ do not depend on $\theta$:

$$\begin{cases} d_6 = & -3b_1 + 2b_2 - b_3 \\ d_5 = & -18a_1 + 8a_2 - 2a_3 \\ d_4 = & 45b_1 - 10b_2 - b_3 \\ d_3 = & 60a_1 - 4a_3 \\ d_2 = & -45b_1 - 10b_2 + b_3 \\ d_1 = & -18a_1 - 8a_2 - 2a_3 \\ d_0 = & 3b_1 + 2b_2 + b_3 \end{cases}$$

## References


[1] P. Comon, "Tensor Decompositions", in *Mathematics in Signal Proc. V*, J. McWhirter and I. Proudler Eds., Clarendon Press, pp. 1–24, 2002.

[2] J. D. Carroll and J. Chang, "Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of "Eckart-Young" Decomposition", *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[3] R. A. Harshman, "Foundations of the PARAFAC procedure: Model and Conditions for an 'explanatory' Multi-mode Factor Analysis", *UCLA Working Papers in Phonetics*, vol.16, pp. 1–84, 1970.

[4] A. Smilde, R. Bro and P. Geladi, *Multi-way Analysis. Applications in the Chemical Sciences*, Chichester, U.K.: John Wiley and Sons, 2004.

[5] L. De Lathauwer, B. De Moor and J. Vandewalle, "An Introduction to Independent Component Analysis", *J. Chemometrics*, vol. 14, pp. 123–149,2000.

[6] N. D. Sidiropoulos, G. B. Giannakis and R. Bro, "Blind PARAFAC Receivers for DS-CDMA Systems", *IEEE Trans. Signal. Proc.*, vol. 48, pp. 810–823, 2000.

[7] L. De Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms – Part I: Lemmas for Partitioned Matrices", *SIAM J. Matrix Anal. Appl.*, 2007, submitted.

[8] L. De Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms – Part II: Definitions and Uniqueness", *SIAM J. Matrix Anal. Appl.*, 2007, submitted.

[9] L. De Lathauwer and D. Nion, "Decompositions of a Higher-Order Tensor in Block Terms – Part III: Alternating Least Squares Algorithms", *SIAM J. Matrix Anal. Appl.*, 2007, submitted.

[10] D. Nion and L. De Lathauwer, "A Block Factor Analysis Based Receiver for Blind Multi-User Access in Wireless Communications", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 825–828, 2006.

[11] A. L. F. de Almeida, G. Favier, J. C. M. Mota, "PARAFAC-Based Unified Tensor Modeling for Wireless Communication Systems with Application to Blind Multiuser Equalization", *Elsevier Signal Processing*, vol. 87, no. 2, pp. 337–351, 2007.

[12] R. Bro, "Multi-way Analysis in the Food Industry: Models, Algorithms, and Applications", Ph.D. dissertation, University of Amsterdam, 1998.

[13] M. Rajih and P. Comon, "Enhanced Line Search: A Novel Method to Accelerate PARAFAC",*Proc. of Eusipco'05*, Antalya, Turkey, 2005.

[14] D. Nion and L. De Lathauwer, "Line Search computation of the Block Factor Model for Blind Multi-User Access in Wireless Communications", *Proc. of SPAWC'06*, 2006.