



# CP-TT: using TT-SVD to greedily construct a Canonical Polyadic tensor approximation

Virginie Ehrlacher, Maria Fuente-Ruiz, Damiano Lombardi

## ► To cite this version:

Virginie Ehrlacher, Maria Fuente-Ruiz, Damiano Lombardi. CP-TT: using TT-SVD to greedily construct a Canonical Polyadic tensor approximation. 2020. hal-03018646v1

**HAL Id: hal-03018646**

**<https://inria.hal.science/hal-03018646v1>**

Preprint submitted on 24 Nov 2020 (v1), last revised 17 Nov 2021 (v3)

**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.

# CP-TT: USING TT-SVD TO GREEDILY CONSTRUCT A CANONICAL POLYADIC TENSOR APPROXIMATION

VIRGINIE EHRLACHER\*, MARIA FUENTE RUIZ†, AND DAMIANO LOMBARDI ‡

**Abstract.** In the present work, a method is proposed in order to compute a Canonical Polyadic (CP) approximation of a given tensor. It is based on a greedy method and an adaptation of the TT-SVD method. The proposed approach can be straightforwardly extended to compute rank- $k$  updates in a stable way. Some numerical experiments are proposed, in which the proposed method is compared to ALS and ASVD methods and performs particularly well for high-order tensors.

**Key words.** Tensor methods, Canonical Polyadic, Tensor Train.

**AMS subject classifications.** 65F99, 65D15

**1. Introduction.** Machine learning and data mining algorithms are becoming increasingly important in analyzing large volume, multi-relational and multi-modal datasets, which are often conveniently represented as multiway arrays or tensors, ([5], [17], [18]). The main challenge in dealing with such data is the so called *curse of dimensionality*, that refers to the need of using a number of degrees of freedom exponentially increasing with the dimension (the reader is referred to [21]).

A tensor is said to be in a full format when it is represented as an original multi-dimensional array. However, distributed storage and processing of high-order tensors in their full format is unfeasible due to the curse of dimensionality.

This problem can be alleviated through various distributed and compressed tensor network formats, achieved by low-rank tensor network approximations, as described for instance in [16]. Some results on low rank decomposition and its uniqueness could be found in [4], [9], [13]. It is important to note that, except for very special data structures, a tensor cannot be compressed without incurring in some compression error, since a low-rank tensor representation is only an approximation of the original tensor.

The concept of compression of multidimensional large-scale data by tensor network decompositions can be intuitively explained as follows. Consider the approximation of a  $d$ -variate function  $F(x) = F(x_1, x_2, \dots, x_d)$  by a finite sum of products of individual functions, each depending on only one or a very few variables. In the simplest scenario, the function  $F(x)$  can be (approximately) represented in the following separable form

$$F(x_1, x_2, \dots, x_d) \approx u^{(1)}(x_1)u^{(2)}(x_2) \cdots u^{(d)}(x_d)$$

In practice, when a  $d$ -variate function  $F(x)$  is discretized into a  $d$ th- order array, or a tensor, the approximation above then corresponds to the representation by rank-1 tensors, also called elementary tensors. Denoting by  $\mathcal{N}_n$ ,  $n = 1, 2, \dots, d$ , the size of the discretization grid associated to the  $n^{th}$  variate, and by  $\mathcal{N} = \max_n(\mathcal{N}_n)$ , the memory requirement to store such a full discretized tensor is equal to  $\prod_{n=1}^d \mathcal{N}_n \leq \mathcal{N}^d$  and grows exponentially with the number of variates  $d$ . On the other hand, the separable representation of a function is completely defined by its factors,  $f^{(n)}(x_n)$ ,

---

\*Ecole des Ponts ParisTech & INRIA (Materials project-team) ([virginie.ehrlacher@enpc.fr](mailto:virginie.ehrlacher@enpc.fr)).

†COMMEDIA, INRIA Paris, France & LJLL, Sorbonne Université ([maria.fuente-ruiz@inria.fr](mailto:maria.fuente-ruiz@inria.fr)).

‡COMMEDIA, INRIA Paris, France & LJLL, Sorbonne Université ([damiano.lombardi@inria.fr](mailto:damiano.lombardi@inria.fr)).

( $n = 1, 2, \dots, d$ ), and requires only  $\sum_{n=1}^d \mathcal{N}_n \ll \mathcal{N}^d$  storage units.

The separation of variables principle can be achieved through different tensor formats. In the present work we will make use of two of them: the Canonical Polyadic and the Tensor Train decompositions.

The **Canonical Polyadic (CP)** decomposition (see [16]):

Let  $r \in \mathbb{N}^*$  be the CP rank, see [10]. The approximation of  $F$  in CP format reads:

$$F(x_1, x_2, \dots, x_d) \approx \sum_{i=1}^r u_i^{(1)}(x_1) u_i^{(2)}(x_2) \cdots u_i^{(d)}(x_d)$$

The **Tensor Train (TT)** decomposition (see [20]): Let  $r_1, \dots, r_{d-1} \in \mathbb{N}^*$  be the TT ranks. The approximation reads:

$$F(x_1, x_2, \dots, x_d) \approx \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} u_{i_1}^{(1)}(x_1) u_{i_1, i_2}^{(2)}(x_2) u_{i_2, i_3}^{(3)}(x_3) \cdots u_{i_{d-2}, i_{d-1}}^{(d-1)}(x_{d-1}) u_{i_{d-1}}^{(d)}(x_d)$$

The main advantages of the CP decomposition are its intuitive expression and its storage scaling. Indeed, instead of the original cost of  $\mathcal{O}(\mathcal{N}^d)$ , the number of entries to store a CP representation reduces to  $\mathcal{O}(d\mathcal{N}r)$ , which scales linearly in the tensor order  $d$  and size  $\mathcal{N}$ . However, the tensor approximation in CP format may be ill-posed [7] and leads to numerical instabilities. The compression of tensors using the CP is usually computed by means of the ALS (alternating least squares) method, which sometimes has some performance issues [2] especially for high-order tensors. Some modifications have been proposed in order to make the compression more efficient, as in [29], [25].

The Tensor Train format is probably one of the most used tensor formats in realistic applications ([3, 32, 26]), due to a good trade off between optimality and numerical stability. The TT format combines two advantages to take into consideration: on the one hand, it is stable from an algorithmic point of view; on the other, it is computationally affordable provided that the TT ranks of the tensors stay reasonably small. The number of entries in the TT format is  $\mathcal{O}(rd\mathcal{N} + r^3(d-2))$ . Even if the number of entries could be larger than in CP, the main advantage of the TT format is its ability to provide stable quasi-optimal rank reduction, obtained, for instance, by truncated singular value decompositions.

In the literature, hybrid formats combining CP with other methods have been proposed in [23], and described in [14]. Also, CP has been combined with TT in [19], where we can see from a different perspective, that both methods combined have potential improvements.

The main contribution of the present work is a method that constructs a CP tensor approximation by exploiting the TT-SVD algorithm. We will refer to it as CP-TT. The proposed method is based on a greedy algorithm in which we reduce the approximation to a sequence of rank-1 (or rank- $k$  for any  $k \in \mathbb{N}^*$ ) approximations. The method, although relying on the TT-SVD iteration does not require to fix *a priori* the order of the variables.

The work is structured as follows: in Section 2 the generic greedy strategy to define a CP decomposition is presented along with two alternatives to the proposed method, namely ALS and ASVD (respectively in Sections 2.2 and 2.3). The formulation of the present approach is presented in Section 2.4. The properties of the method, the

computational cost and the extension to compute stable rank- $k$  updates are presented in Section 3. Some numerical experiments and results are presented in Section 4.

**2. The method.** Before detailing the method formulation and its properties, we introduce hereafter the notation and the problem setting. Then, different methods are recalled, which were proposed in the literature in order to construct a CP tensor approximation of a given tensor. Among them, in the present work, we focus on ALS and on ASVD. The Section ends with the presentation of the CP-TT method.

**2.1. Notation.** The notation is presented. The method proposed in the present work was motivated by high-dimensional function approximation. We therefore make the choice to deal with continuous tensors. The extension to discrete tensors is straightforward.

Let  $d \in \mathbb{N}^*$  be the tensor order. Let  $i = 1, \dots, d$ , and  $d_i \in \mathbb{N}^*$ . We introduce  $\Omega^{(i)} \in \mathbb{R}^{d_i}$ , which are open bounded sets. Let the domain be denoted by:  $\Omega = \Omega^{(1)} \times \dots \times \Omega^{(d)}$ : the variables are  $x \in \Omega$ ,  $x = (x_1, \dots, x_d)$ .

A real valued tensor  $F$  is a function defined as:

$$(2.1) \quad F : \begin{cases} \Omega \rightarrow \mathbb{R} \\ x \mapsto F(x_1, \dots, x_d). \end{cases}$$

Let  $k \in \mathbb{N}^*$ , and  $1 \leq p \leq \infty$ . In the present work we restrict to the problem of approximating a function  $F \in L^2(\Omega)$ . Observe that  $L^2(\Omega) = L^2(\Omega^{(1)}) \otimes \dots \otimes L^2(\Omega^{(d)})$ . From now on, we denote the  $L^2$  scalar product of two nonzero functions  $u \in L^2(\Omega)$  and  $v \in L^2(\Omega)$ :

$$\langle u, v \rangle = \int_{\Omega} uv \, dx,$$

and the  $L^2$  norm as  $\|u\|_{L^2(\Omega)}^2 = \langle u, u \rangle$ .

A pure tensor product is a function defined as follows:

$$(2.2) \quad u^{(1)} \otimes \dots \otimes u^{(d)} : \begin{cases} \Omega^{(1)} \times \dots \times \Omega^{(d)} \rightarrow \mathbb{R} \\ (x_1, \dots, x_d) \mapsto u^{(1)}(x_1) \dots u^{(d)}(x_d). \end{cases}$$

Let  $n \in \mathbb{N}^*$ ,  $c_i \in \mathbb{R}$ ,  $i = 1, \dots, n$  be a sequence of real numbers, a tensor  $T$  in CP format is written as follows:

$$T = \sum_{i=1}^n c_i u_i^{(1)} \otimes \dots \otimes u_i^{(d)}.$$

The number of terms  $n$  is the CP-rank of the tensor. The set of the rank- $n$  tensors in CP format is denoted as:

$$(2.3) \quad \mathcal{T}^{(n)} = \left\{ T = \sum_{i=1}^n c_i u_i^{(1)} \otimes \dots \otimes u_i^{(d)}, \, c \in \mathbb{R}^n, \, u_i^{(j)} \in L^2(\Omega_j) \, \forall i = 1, \dots, n, j = 1, \dots, d \right\}$$

The objective is to construct a tensor approximation of a given function. The problem reads as follows:

$$(2.4) \quad T_*^{(n)} = \arg \inf_{T \in \mathcal{T}^{(n)}} \|F - T\|_{L^2(\Omega)}^2.$$

A solution  $T_*^{(n)}$  to (2.4) is called a best rank- $n$  approximation of  $F$ . This problem is known to be ill-posed, in the sense that there may not exist any solutions to (2.4). This translates into numerical instabilities for most algorithms aiming at computing CP approxiamtions of tensors [8].

One way to approach the solution of the problem is to adopt a greedy strategy, that consists in replacing the best rank- $n$  approximation by a sequence of rank-1 approximations, which are well defined. The general strategy reads as follows:

$$\begin{aligned} T^{(0)} &= 0, \\ t_*^{(k+1)} &\in \arg \inf_{t \in \mathcal{T}^{(1)}} \|F - T^{(k)} - t\|_{L^2(\Omega)}^2, \\ T^{(k+1)} &= T^{(k)} + t_*^{(k+1)}. \end{aligned}$$

There are several algorithms in order to compute a rank-1 approximation solution to the above problem, and the methods presented hereafter aim at computing a numerical approximation of a solution  $t_*^{(k+1)}$  of the above minimization problem.

**2.2. Alternating least square (ALS).** The Alternating Least Square (ALS) method is based on fixed point method iterations and is one of the most used method to compress tensors in CP format. It consists in the following iterative scheme to compute an approximation  $t_{\text{ALS}}^{(k+1)}$  of  $t_*^{(k+1)}$ , for a given error threshold  $\epsilon > 0$ .

**ALS:**

- **Initialization:** Choose randomly  $u^{(1,0)} \in L^2(\Omega^{(1)})$ , ...,  $u^{(d,0)} \in L^2(\Omega^{(d)})$  and set  $m \geq 1$ .
- **Iteration**  $m \geq 1$ :
  1. For  $i = 1, \dots, d$ , compute  $u^{(i,m)} \in L^2(\Omega^{(i)})$  solution to
 
$$u^{(i,m)} \in \operatorname{argmin}_{u^{(i)} \in L^2(\Omega^{(i)})} \|F - T^{(k)} - u^{(1,m)} \otimes u^{(2,m)} \otimes \dots \otimes u^{(i-1,m)} \otimes u^{(i)} \otimes u^{(i+1,m-1)} \otimes \dots \otimes u^{(d,m-1)}\|_{L^2(\Omega)}^2.$$
  2. Compute  $\eta := \|u^{(1,m)} \otimes \dots \otimes u^{(d,m)} - u^{(1,m-1)} \otimes \dots \otimes u^{(d,m-1)}\|_{L^2(\Omega)}$ .
  3. If  $\eta < \epsilon$ , define  $t_{\text{ALS}}^{(k+1)} := u^{(1,m)} \otimes \dots \otimes u^{(d,m)}$  the approximation of the best rank-one approximation of  $F - T^{(k)}$ . Else, set  $m = 2$  and iterate again.

The convergence properties of the ALS algorithm have been abundantly studied. We refer the reader for more details to the following series of works [28, 27, 11, 31, 22].

**2.3. Alternating Singular Value Decomposition (ASVD).** For the sake of comparison with the ALS method and our proposed procedure, we introduce in this section the so-called Alternating Singular Value Decomposition (ASVD) method, which was proposed in [12].

We denote by  $\mathcal{I} := \{\{i, j\}, 1 \leq i < j \leq d\}$  be the set of all possible pairs of indices between 1 and  $d$ . An ordering of the elements of  $\mathcal{I}$  is chosen so that

$$\mathcal{I} = (I_k)_{1 \leq k \leq K}$$

where  $K = |\mathcal{I}|$ .

The ASVD method then also consists in an iterative scheme to compute an approximation  $t_{\text{ASVD}}^{(k+1)}$  of  $t_*^{(k+1)}$ , for a given error threshold  $\epsilon > 0$ , which reads as follows:

**ASVD:**

- **Initialization:** Choose randomly  $u^{(1,0)} \in L^2(\Omega^{(1)})$ , ...,  $u^{(d,0)} \in L^2(\Omega^{(d)})$  and set  $m \geq 1$ .

- **Iteration**  $m \geq 1$ :

1. Set  $u^{(i,m)} = u^{(i,m-1)}$  for all  $1 \leq i \leq d$ .
2. For  $k = 1, \dots, K$ , let  $I_k = (i_k, j_k)$  and compute  $U^{(k,m)} \in L^2(\Omega^{(i_k)} \times \Omega^{(j_k)})$  solution to

$$U^{(k,m)} \in \underset{U^{(k)} \in L^2(\Omega^{(i_k)} \times \Omega^{(j_k)})}{\operatorname{argmin}} \|F - T^{(k)} - U^{(k)} \otimes \bigotimes_{1 \leq l \leq d; l \neq i_k, j_k} u^{(l,m)}\|_{L^2(\Omega)}^2.$$

Update  $u^{(i_k,m)}$  and  $u^{(j_k,m)}$  so that  $(u^{(i_k,m)}, u^{(j_k,m)}) \in L^2(\Omega^{(i_k)}) \times L^2(\Omega^{(j_k)})$  is solution to

$$(u^{(i_k,m)}, u^{(j_k,m)}) \in \underset{(u^{(i_k)}, u^{(j_k)}) \in L^2(\Omega^{(i_k)}) \times L^2(\Omega^{(j_k)})}{\operatorname{argmin}} \|U^{(k,m)} - u^{(i_k)} \otimes u^{(j_k)}\|_{L^2(\Omega)}^2.$$

3. Compute  $\eta := \|u^{(1,m)} \otimes \dots \otimes u^{(d,m)} - u^{(1,m-1)} \otimes \dots \otimes u^{(d,m-1)}\|_{L^2(\Omega)}$ .
4. If  $\eta < \epsilon$ , define  $t_{\text{ASVD}}^{(k+1)} := u^{(1,m)} \otimes \dots \otimes u^{(d,m)}$  the approximation of the best rank-one approximation of  $F - T^{(k)}$ . Else, set  $m = 2$  and iterate again.

**2.4. CP-TT.** In this section, the CP-TT method is introduced. The idea behind this method of tensor approximation is to combine the CP format and the TT-SVD method, in order to benefit from the simplicity of the CP format and the numerical stability of the TT-SVD.

The principle of the method (which is sketched in Fig.1 in the particular case of a 3rd order tensor) relies itself on a greedy algorithm, which is detailed below. Without loss of generality, for the sake of simplicity, we present the method in full details in the case where  $k = 0$ , i.e. when  $T^{(0)} = 0$  so that  $F - T^{(0)} = F$ .

For all  $1 \leq i \leq d$ , let us define  $\widehat{\Omega}_i := \Omega_1 \times \dots \times \Omega_{i-1} \times \Omega_{i+1} \times \dots \times \Omega_d$  and let us introduce  $M_i : \Omega_i \times \widehat{\Omega}_i \rightarrow \mathbb{R}$  the  $i^{\text{th}}$  unfolding of  $F$  defined by

$$M_i(x_i, y_i) = F(x_1, \dots, x_d)$$

for all  $x_i \in \Omega_i$  and all  $y_i = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d) \in \widehat{\Omega}_i$ .

Let us now consider a POD decomposition of  $M_i$ :

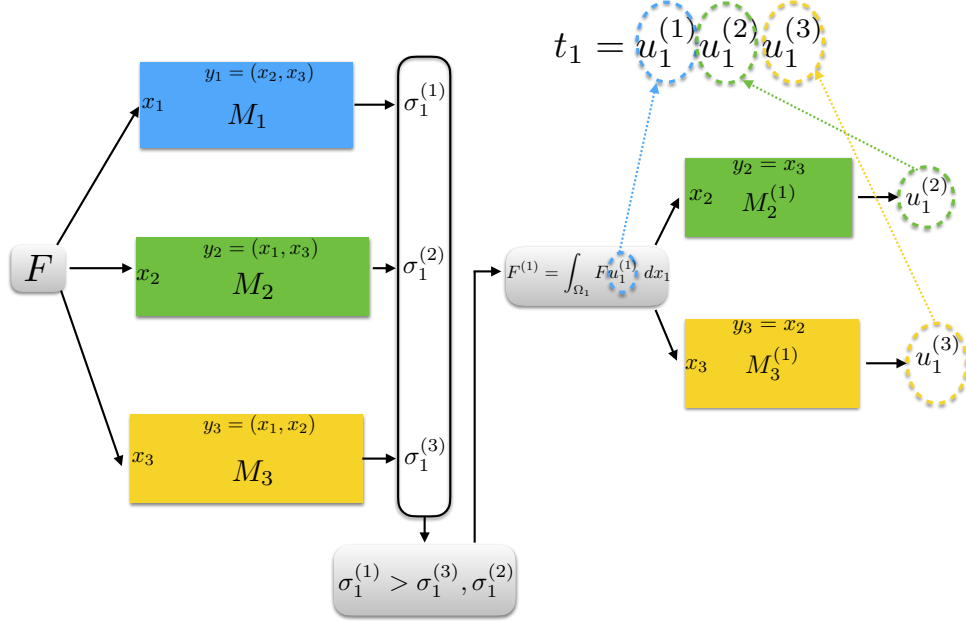
$$(2.5) \quad M_i(x_i, y_i) = \sum_{j>0} \sigma_j^{(i)} u_j^{(i)}(x_i) v_j^{(i)}(y_i),$$

where  $u_j^{(i)}$  is an orthonormal basis of  $L^2(\Omega_i)$ ,  $v_j^{(i)}$  is an orthonormal basis of  $L^2(\widehat{\Omega}_i)$  and  $\sigma_j^{(i)} \in \mathbb{R}^+$  are the singular values, that we assumed to be ranked in decreasing order.

We define  $1 \leq i_* \leq d$  as the integer such that:

$$(2.6) \quad i_* = \arg \max_{1 \leq i \leq d} \sigma_1^{(i)}.$$

A rank-one approximation of  $M_{i_*}$ , and therefore of  $F$ , is then constructed as  $m_{i_*} = \sigma_1^{(i_*)} u_1^{(i_*)} \otimes v_1^{(i_*)} = u_1^{(i_*)} \otimes w_1^{(i_*)}$ , where  $w_1^{(i_*)} \in L^2(\widehat{\Omega}_{i_*})$  is defined as:  $w_1^{(i_*)} =$

FIG. 1. Schematic representation of the CP-TT iteration for a 3rd order tensor  $F$ .

$$\sigma_1^{(i_*)} v_1^{(i_*)}.$$

Without loss of generality (up to changing the labeling of the variables), let us assume that  $i_* = 1$ . It then holds that:

$$w_1^{(1)}(y_1) = \int_{\Omega_1} M_1(x_1, y_1) u_1^{(1)}(x_1) dx_1, \quad \forall y_1 = (x_2, \dots, x_d) \in \widehat{\Omega}_1$$

We introduce an auxiliary tensor, defined as follows:

$$F^{(1)}(x_2, \dots, x_d) = w_1^{(1)}(y_1) = \int_{\Omega_1} F(x_1, \dots, x_d) u_1^{(1)}(x_1) dx_1.$$

In the next step of the method, we look for an approximation of  $F^{(1)}$ , by proceeding in an analogous way.

For all  $2 \leq i \leq d$ , let us define  $\widehat{\Omega}_i^{(1)} := \Omega_2 \times \dots \times \Omega_{i-1} \times \Omega_{i+1} \times \dots \times \Omega_d$  and let us introduce  $M_i^{(1)} : \Omega_i \times \widehat{\Omega}_i^{(1)} \rightarrow \mathbb{R}$  the  $i^{th}$  unfolding of  $F^{(1)}$  defined by:

$$M_i^{(1)}(x_i, y_i^{(1)}) = F^{(1)}(x_2, \dots, x_d)$$

for all  $x_i \in \Omega_i$  and all  $y_i^{(1)} = (x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_d) \in \widehat{\Omega}_i^{(1)}$ .

A POD decomposition of  $M_i^{(1)}$  is computed, of the form:

$$(2.7) \quad M_i^{(1)}(x_i, y_i^{(1)}) = \sum_{j>0} \sigma_j^{(i,1)} u_j^{(i,1)}(x_i) v_j^{(i,1)}(y_i^{(1)}),$$

where  $u_j^{(i,1)}$  is an orthonormal basis of  $L^2(\Omega_i)$ ,  $v_j^{(i,1)}$  is an orthonormal basis of  $L^2(\widehat{\Omega}_i^{(1)})$  and  $\sigma_j^{(i,1)} \in \mathbb{R}^+$  are the singular values, that we assumed to be ranked in decreasing order.

We define  $2 \leq i_*^{(1)} \leq d$  as the integer such that:

$$(2.8) \quad i_*^{(1)} = \arg \max_{2 \leq i \leq d} \sigma_1^{(i,1)}.$$

Analogously to what we have presented above, up to changing the labeling of the variables, we can assume that  $i_*^{(1)} = 2$ . We define:

$$(2.9) \quad F^{(2)}(x_3, \dots, x_d) = \int_{\Omega_2} F^{(1)}(x_2, \dots, x_d) u_1^{(2,1)}(x_2) dx_2.$$

This process continues until it remains a 2-nd order tensor and the POD decomposition gives us the final terms. Let us suppose, for sake of simplicity in the notation, that the two last variables are  $x_{d-1}, x_d$ . The truncated rank-one approximation of  $F^{(d-2)}$  is given by:

$$F^{(d-2)}(x_{d-1}, x_d) \approx \sigma_1^{(d-1,d-2)} u_1^{(d-1,d-2)}(x_{d-1}) v_1^{(d,d-2)}(x_d).$$

The final rank-one approximation of  $F$  is obtained as:

$$t_{CPTT}^{(1)} = \sigma_1^{(d-1,d-2)} u_1^{(1)} \otimes u_1^{(2,1)} \otimes \dots \otimes u_1^{(d-1,d-2)} \otimes v_1^{(d,d-2)}.$$

*Remark.:* We have considered, for sake of simplicity in the notation, the order  $I_* = \{1, 2, \dots, d\}$ . We stress here that in the proposed method, the order is not fixed *a priori*. Instead, it is the result of the iteration, and, in particular, of the optimization step on  $\sigma_i^{(\cdot)}$ .

The CP-TT method is summarized in the pseudocode 2.1.

**2.5. Optimization of the CP coefficients.** To obtain a more accurate approximation, we introduce an optimization problem in order to find the best linear combination of the rank-1 tensor obtained by one of the three methods presented above. Let us denote by  $t^{(1)}, \dots, t^{(k)}$  the  $k$  pure rank-1 tensors obtained after  $k$  iterations of each of the three methods mentioned above (ALS, ASVD or CPTT).

For each method, at the  $k^{th}$  iteration, we look for coefficients  $c := (c_1, \dots, c_k) \in \mathbb{R}^k$  such that:

$$T^{(k)}(c) = \sum_{j=1}^k c_j t^{(j)}(x_1, \dots, x_d)$$

minimizes the norm of the residual. The optimization problem reads:

$$(c_{*,1}, \dots, c_{*,k}) =: c_* = \arg \min_{c \in \mathbb{R}^k} \left( \frac{1}{2} \|F - T^{(k)}(c)\|_{L^2(\Omega)}^2 \right)$$

We define the functional  $J$ :

$$J : \begin{cases} \mathbb{R}^k & \rightarrow \mathbb{R}^+ \\ c & \mapsto \|F - T^{(k)}(c)\|_{L^2(\Omega)}^2 \end{cases}$$

The Euler Lagrange equations read as follows:

$$\forall 1 \leq l \leq k, \quad \frac{\partial J}{\partial c_l}(c_*) = \int_{\Omega} (F - \sum_{j=1}^k c_{*,j} t^{(j)}) (-t^{(l)}) dx_1 \dots dx_d = 0$$

This implies that

$$\forall 1 \leq l \leq k, \quad \int_{\Omega} F t^{(l)} dx_1 \dots dx_d = \sum_{j=1}^k \int_{\Omega} c_{*,j} t^{(j)} t^{(l)} dx_1 \dots dx_d,$$

which reduces to the following linear system

$$Ac_* = b$$

where  $A = (A_{lj})_{1 \leq l, j \leq k} \in \mathbb{R}^{k \times k}$ ,  $b = (b_l)_{1 \leq l \leq k} \in \mathbb{R}^k$  with

$$\forall 1 \leq l, j \leq k, \quad A_{lj} = \int_{\Omega} t^{(j)} t^{(l)} dx_1 \dots dx_d \text{ and } b_l = \int_{\Omega} F t^{(l)} dx_1 \dots dx_d.$$

For each of the three methods mentioned above (ALS, ASVD, CP-TT), the resulting procedure then reads as follows:

**ALGORITHM 1:**

1. **Initialization:**  $k = 0$ ,  $T^{(0)} = 0$ ;
2. **Iteration  $k$ :** Compute  $t^{(k+1)}$  an approximate solution to the optimization problem

$$\arg \inf_{t \in \mathcal{T}^{(1)}} \|F - T^{(k)} - t\|_{L^2(\Omega)}^2,$$

using either the ALS, ASVD or CP-TT method.

Compute

$$c_*^{(k+1)} := \arg \min_{c \in \mathbb{R}^{k+1}} \left( \frac{1}{2} \|F - T^{(k+1)}(c)\|_{L^2(\Omega)}^2 \right).$$

Define  $T^{(k+1)} := \sum_{l=1}^{k+1} c_{*,l}^{(k+1)} t^{(l)}$  and set  $k := k + 1$ .

**3. Properties of the CP-TT method.** In this section, several properties of the proposed CP-TT method are analyzed.

**3.1. Orthogonality properties.** Let us analyze the inner iteration used in order to compute a rank-1 update. For sake of simplicity, without loss of generality we consider the sequence of  $i_*$  corresponding to the largest singular values is  $1, 2, \dots, d$ . Let us define

$$G^{(1)}(x_1, \dots, x_d) := F - u_1^{(1)}(x_1) F^{(1)}(x_2, \dots, x_d).$$

Then, it holds that

$$\|G^{(1)}\|_{L^2(\Omega)}^2 = \sum_{j>1} \left( \sigma_j^{(1)} \right)^2.$$

By projecting the residual  $F$  onto  $u_1^{(1)}$  we get

**Algorithm 2.1** Scheme of the computation algorithm of the CP-TT term**The CP-TT method:**

We define  $F^{(0)} := F$

$I^{(0)} = \{1, \dots, d\}$

Set  $m = 1$

**while**  $\#I^{(m)} > 2$  **do**

    Compute the unfoldings  $(M_i)_{i \in I^{(m-1)}}$  of  $F^{(m-1)}$

    Compute their POD approximation, and denote by  $(\sigma_j^i)_{j \geq 1}$  the singular values of  $M_i$  for all  $i \in I^{(m-1)}$

    Select the largest singular value:  $i_{*,m} = \arg \max_{i \in I^{(m-1)}} \sigma_1^i$

    Define  $I^{(m)} := I^{(m-1)} \setminus \{i_{*,m}\}$

    Select  $u_1^{i_{*,m}}$  the first singular function associated to the POD decomposition of  $M_{i_{*,m}}$

    Define  $F^{(m+1)} := \int_{\Omega_{i_*}} F^{(m)} u_1^{i_{*,m}} dx_{i_*}$

    Set  $m := m + 1$ .

**end while**

Let  $I^{(d-2)} = \{i_{*,d-1}, i_{*,d}\}$ . Compute the POD decomposition of  $F^{(d-2)}$

Let  $u_1^{i_{*,d-1}}$  and  $u_1^{i_{*,d}}$  be the first singular functions of  $F^{(d-2)}$

Define  $t_{\text{CPTT}}^{(1)} = \bigotimes_{j=1}^d u_1^{i_{*,j}}$ .

(3.1)

$$\int_{\Omega_1} u_1^{(1)}(x_1) F(x_1, \dots, x_d) dx_1 = \|u_1^{(1)}\|_{L^2(\Omega_1)}^2 F^{(1)}(x_2, \dots, x_d) + \int_{\Omega_1} u_1^{(1)}(x_1) G^{(1)}(x_1, \dots, x_d) dx_1,$$

which implies that

$$\int_{\Omega_1} u_1^{(1)}(x_1) G^{(1)}(x_1, \dots, x_d) dx_1 = 0.$$

We proceed then to the next step of the inner iteration in the method, in which we take  $F^{(1)}$  as the tensor to be approximated. We have suppose that the largest singular value corresponds again to the first of the remaining unfoldings and hence:

$$F^{(2)}(x_3, \dots, x_d) = \int_{\Omega_2} F^{(1)}(x_2, \dots, x_d) u_1^{(2,1)}(x_2) dx_2,$$

$$F^{(1)} = u_1^{(2,1)}(x_2) F^{(2)}(x_3, \dots, x_d) + G^{(2)}(x_2, \dots, x_d).$$

From which we can deduce:

$$\int_{\Omega_2} u_1^{(2,1)}(x_2) G^{(2)}(x_2, \dots, x_d) dx_2 = 0.$$

This results could be extended to all the substeps of the inner iteration, what means that for all  $n \in \mathbb{N}$  from 1 to  $d$ :

$$(3.2) \quad F^{(n)} = F^{(n+1)} u_1^{(n+1,1)} + G^{(n+1)}.$$

And, for all  $n \in \mathbb{N}$  from 1 to  $d$ :

$$(3.3) \quad \int_{\Omega_n} G^{(n)} u_1^{(n,n-1)} dx_n = 0$$

Let us consider the first iteration for which the residual  $R^{(0)} = F - F^{(0)} = F$  and we can write:

$$R^{(1)} = F - c_1 u_1^{(1,1)} \otimes u_1^{(2,1)} \otimes \dots \otimes u_1^{(d-1,d-2)} \otimes u_1^{(d,d-1)},$$

where, in the following, to continue with the established notation:  $u_1^{(1,1)} = u_1^{(1)}$ .

Using the above obtained relations this can be rewritten as:

$$R^{(1)} = u_1^{(1,1)} u_1^{(2,1)} \dots u_1^{(d-3,d-4)} u_1^{(d-2,d-3)} G^{(d-1)} + u_1^{(1,1)} \dots u_1^{(d-3,d-4)} G^{(d-2)} + \dots + u_1^{(1,1)} G^{(2)} + G^{(1)}.$$

The squared norm of the residual reads:

$$\begin{aligned} \|R^{(1)}\|_{L^2(\Omega)}^2 &= \int_{\Omega} (u_1^{(1,1)^2} u_1^{(2,1)^2} \dots u_1^{(d-3,1)^2} u_1^{(d-2,1)^2} G^{(d-1)^2} + u_1^{(1,1)^2} u_1^{(2,1)^2} \dots u_1^{(d-3,1)^2} u_1^{(d-2,1)^2} G^{(d-1)} G^{(d-2)} + \dots \\ &\quad \dots + u_1^{(1,1)^2} u_1^{(2,1)^2} \dots u_1^{(d-3,1)^2} G^{(d-2)^2} + \dots + u_1^{(1,1)^2} G^{(2)^2} + G^{(1)^2}) dx_1 \dots dx_d \end{aligned}$$

Separating the integrals, applying the orthogonality relation obtained in Eq.(3.3) and using the fact that the modes are unitary, the crossed terms in  $G$  vanish and this simplifies as:

$$\begin{aligned} \|R^{(1)}\|_{L^2(\Omega)}^2 &= \int_{\Omega} (u_1^{(1,1)^2} u_1^{(2,1)^2} \dots u_1^{(d-3,d-4)^2} u_1^{(d-2,d-3)^2} G^{(d-1)^2} + u_1^{(1,1)^2} u_1^{(2,1)^2} \dots u_1^{(d-3,d-4)^2} G^{(d-2)^2} + \dots \\ &\quad \dots + u_1^{(1,1)^2} G^{(2)^2} + G^{(1)^2}) dx_1 \dots dx_d \end{aligned}$$

Replacing the norms by the squared singular values due to the properties of  $G$ :

$$(3.4) \quad \int_{\Omega_k \times \dots \times \Omega_d} G^{(k)} dx_k \dots dx_{d-1} dx_d = \sum_{j>1} \sigma_j^{(k,)^2}$$

using this result in the expression of the residual is possible to get the analogue expression for the total residual as the sum of the reminders on each term.

$$\|R^{(1)}\|_{L^2(\Omega)}^2 = \|G^{(1)}\|_{L^2(\Omega)}^2 + \|G^{(2)}\|_{L^2(\Omega_2 \times \dots \times \Omega_d)}^2 + \dots + \|G^{(d-1)}\|_{L^2(\Omega_{d-1} \times \Omega_d)}^2$$

*Example: Properties in the 4-th dimensional case.* In the following we illustrate these steps on a 4-th order tensor. After the first iteration, the residual reads:

$$R^{(1)} = F - c_1 u_1^{(1,1)} \otimes u_1^{(2,1)} \otimes u_1^{(3,2)} \otimes u_1^{(4,3)} = u_1^{(1,1)} u_1^{(2,1)} G^{(3,2)} + u_1^{(1,1)} G^{(2)} + G^{(1)}$$

Where we have taken into account the fact that in the last iteration  $F^{(3)} = c_1 u_1^{(4,1)}$  because of its orthogonality in the POD decomposition.

The squared norm of the residual reads:

$$\begin{aligned} \|R^{(1)}\|_{L^2(\Omega)}^2 &= \int_{\Omega} (u_1^{(1,1)} u_1^{(2,1)} G^{(3,2)} + 2u_1^{(1,1)} u_1^{(2,1)} G^{(3)} G^{(2)} + 2u_1^{(1,1)} G^{(2)} G^{(1)} + u_1^{(1,1)} G^{(2)^2} + \dots \\ &\quad \dots + 2u_1^{(1,1)} G^{(2)} G^{(1)} + G^{(1)^2}) dx_1 \dots dx_4 \end{aligned}$$

Using the orthogonality relations seen in Eq.(3.3) and the orthonormality of the modes:

$$\|R^{(1)}\|_{L^2(\Omega)}^2 = \int_{\Omega_3 \times \Omega_4} G^{(3,2)^2} dx_3 dx_4 + \int_{\Omega_2 \times \Omega_3 \times \Omega_4} G^{(2)^2} dx_2 dx_3 dx_4 + \int_{\Omega} G^{(1)^2} dx_1 dx_2 dx_3 dx_4$$

Adding the proprieties of  $G$  from Eq.(3.4) it leads to:

$$\|R^{(1)}\|_{L^2(\Omega)}^2 = \sigma_1^{(3,2)^2} - \sigma_1^{(4,3)^2} + \sigma_1^{(2,1)^2} - \sigma_1^{(2,1)^2} + \|F\|_{L^2(\Omega)}^2 - \sigma_1^{(3,2)^2} = \|F\|_{L^2(\Omega)}^2 - c^2$$

where we have called  $c$  to the last of the considered eigenvalues.

What we can see at this point is that  $\|F^{(1)}\|_{L^2(\Omega)}^2 = \sigma_1^{(1)^2}$ .

And

$$\|F\|_{L^2(\Omega)}^2 = \|F^{(1)}\|_{L^2(\Omega)}^2 + \|G^{(1)}\|_{L^2(\Omega)}^2$$

We can also confirm that the squared norm of the total residual in one iteration is the sum of the squared norms of the residuals in each term on the iteration:

$$\|R^{(1)}\|_{L^2(\Omega)}^2 = \|G^{(1)}\|_{L^2(\Omega)}^2 + \|G^{(2)}\|_{L^2(\Omega_2 \times \Omega_3 \times \Omega_4)}^2 + \|G^{(3)}\|_{L^2(\Omega_3 \times \Omega_4)}^2$$

**3.2. CP-TT is optimal for  $d = 2$ , and it retrieves exactly a rank-1 tensor.** The method proposed, when  $d = 2$ , reduces to compute a POD of a two variables function  $F(x_1, x_2)$ . By the Eckart-Young theorem, the result is optimal. As for other methods, when the number of variables is  $d > 2$ , CP-TT is in general sub-optimal.

When the function to be approximated is a rank-1 tensor, it is exactly recovered by the CP-TT method in  $N_{it} = 1$  iteration.

Let  $F = f^{(1)} \otimes \dots \otimes f^{(d)}$ . Without loss of generality, let the order of the chosen best unfolding be  $I = \{1, \dots, d\}$ . The best approximation for the first unfolding gives us the term  $u_1^{(1)} = \frac{f^{(1)}}{\|f^{(1)}\|_{L^2(\Omega^{(1)})}}$ . Proceeding with the CP-TT iteration, and

considering the unfolding of  $F^{(1)}$  relative to  $x_2$ , the result is  $u^{(2,1)} = \frac{f^{(2)}}{\|f^{(2)}\|_{L^2(\Omega^{(2)})}}$ . This can be iterated, leading to the following outcome of the first CP-TT iteration:

$$\begin{aligned}\tilde{F} &= cu^{(1)} \otimes \dots \otimes u^{(d)}, \\ c &= \|f^{(1)}\|_{L^2(\Omega^{(1)})} \cdot \dots \cdot \|f^{(d)}\|_{L^2(\Omega^{(d)})}, \\ u^{(1)} &= \frac{f^{(1)}}{\|f^{(1)}\|_{L^2(\Omega^{(1)})}}, \\ &\quad \dots \\ u^{(d)} &= \frac{f^{(d)}}{\|f^{(d)}\|_{L^2(\Omega^{(d)})}}.\end{aligned}$$

Henceforth,  $\tilde{F} = F$ , and the method stops.

**3.3. Computing  $k$  terms: stability.** The CP-TT iteration can be suitably modified to compute more than a rank-one update. Let  $k > 1$  be the number of pure tensor terms to be computed. We introduce the following modification of the above described method. When computing the unfoldings POD for  $M_i$ , instead of selecting the index of the variable corresponding to the largest first singular value, solve the following problem:

$$(3.5) \quad i_* = \arg \max_{1 \leq i \leq d} \sum_{j=1}^k \left( \sigma_j^{(i)} \right)^2.$$

Then, consider the set of  $u_j^{(i_*)}$ ,  $j = 1, \dots, k$ . These can be used to compute:

$$F_j^{(1)} = \int_{\Omega_{i_*}} F u_j^{(i_*)} dx_{i_*}.$$

The CP-TT iteration can be carried out independently for each of the  $F_j^{(1)}$ , for the remaining variables, leading to the definition of  $k$  pure tensor terms. The method so defined computes a stable CP decomposition, in the sense that:

$$\left\| \sum_{j=1}^k t_j \right\|_{L^2(\Omega)}^2 < C \Rightarrow \|t_j\|_{L^2(\Omega)}^2 < C_j, \quad \forall j.$$

This follows from the very first step of the modified iteration. Indeed:

$$(3.6) \quad \left\| \sum_{j=1}^k t_j \right\|_{L^2(\Omega)}^2 = \sum_{l=1}^k \sum_{m=1}^k \langle t_l, t_m \rangle = \sum_{l,m=1}^k \langle u_l^{(i_*)}, u_m^{(i_*)} \rangle_{L^2(\Omega_{i_*})} \langle w_l^{(i_*)}, w_m^{(i_*)} \rangle_{L^2(\Omega \setminus \Omega_{i_*})}$$

Since the terms  $u_j^{(i_*)}$  are elements of an orthonormal basis of  $L^2(\Omega_{i_*})$  we have:

$$(3.7) \quad \sum_{l,m=1}^k \langle u_l^{(i_*)}, u_m^{(i_*)} \rangle_{L^2(\Omega_{i_*})} \langle w_l^{(i_*)}, w_m^{(i_*)} \rangle_{L^2(\Omega \setminus \Omega_{i_*})} = \sum_{l=1}^k \|t_l\|_{L^2(\Omega)}^2,$$

and this is enough to conclude.

**3.4. Discretization and computational costs.** In this section, the computational cost of the proposed method is detailed. Let  $(\mathcal{N}_1, \dots, \mathcal{N}_d)$  be the number of degrees of freedom used to discretize the problem in every direction. The storage of a full tensor would require a memory of  $\prod_{j=1}^d \mathcal{N}_j$ . The simplicity of the CP tensor format is also complemented by its appealing storage scaling, which is:  $r \cdot \sum_{j=1}^d \mathcal{N}_j$ . In view of implementing the proposed method, some aspects need to be considered. The most costly operation to be performed is the computation of the POD of the  $d$  unfoldings. In the following, a method is proposed, in the case in which the function to be compressed is given in CP format, to provide a cheap alternative for the POD decomposition of the unfoldings.

In [1] a fast multilinear singular value decomposition is proposed for symmetric Toeplitz and Hankel third order tensors has been proposed. A fast computation of the gradients in ALS method is proposed in [24]. In [30] efficient representations are exploited to speed up large-scale tensor decompositions. In [15] the authors propose parallel methods to accelerate the decomposition of sparse tensors.

**3.4.1. Computing the POD of an unfolding without storing it.** Let the tensor  $F$  be given in CP format:

$$F = \sum_{i=1}^r c_i f_i^{(1)} \otimes \dots \otimes f_i^{(d)},$$

where  $\forall i, f_i^{(j)} \in L^2(\Omega_j)$  and  $c_i \in \mathbb{R}$ . Henceforth,  $\text{rank}(F) = r$ .

Without loss of generality we present the method for  $M^{(1)}(x_1, y_1)$ , the unfolding relative to the first variable.

The first step consists in introducing the correlation operator:

$$(3.8) \quad \forall f(x_1) \in L^2(\Omega_1), \quad K^{(1)}f = \int_{\Omega_1} \kappa^{(1)}(x_1, x'_1) f(x'_1) dx'_1,$$

where the kernel  $\kappa^{(1)}$  is defined as follows:

$$(3.9) \quad \kappa^{(1)}(x_1, x'_1) = \int_{\Omega_2 \times \dots \times \Omega_d} M(x_1, y_1) M(x'_1, y_1) dy_1.$$

The operator  $K^{(1)}$  is compact, non-negative, self-adjoint, and hence, there exists a sequence of orthonormal functions  $u_i^{(1)}(x_1)$ ,  $i \in \mathbb{N}^*$  and a sequence of scalar values  $\lambda_i \geq 0$  such that:

$$(3.10) \quad K^{(1)}u_i^{(1)} = \lambda_i u_i^{(1)}.$$

When the tensor is given in CP format, the expression of the kernel is reduced to:

$$(3.11) \quad \kappa^{(1)}(x_1, x'_1) = \sum_{i,j=1}^r c_i c_j f_i^{(1)}(x_1) f_j^{(1)}(x'_1) \int_{\Omega_2} f_i^{(2)}(x_2) f_j^{(2)}(x_2) dx_2 \dots \int_{\Omega_d} f_i^{(d)}(x_d) f_j^{(d)}(x_d) dx_d.$$

The POD decomposition of the fibers  $f_i^{(1)}$  is computed, providing:

$$(3.12) \quad f_i^{(1)}(x_1) = \sum_{m=1}^r a_{im} z_m^{(1)}(x_1),$$

where  $\langle z_m^{(1)}, z_l^{(1)} \rangle_{L^2(\Omega_1)} = \delta_{lm}$  and  $a_{im} = \sum_{s=1}^r \sigma_s^{(1)} \delta_{is} v_{sm}$ , where  $\sigma_s^{(1)} \geq 0$  and  $v_{sm}$  are the entries of a unitary matrix. This is injected into the correlation operator, leading to:

$$(3.13) \quad \mathcal{K}(x_1, x'_1) = \sum_{l,m=1}^r z_l^{(1)}(x_1) z_m^{(1)}(x'_1) \left[ \sum_{i,j=1}^r a_{il} a_{jm} c_i c_j \prod_{k=2}^d \int_{\Omega_k} f_i^{(k)}(x_k) f_j^{(k)}(x_k) dx_k \right].$$

Let  $A \in \mathbb{R}^{r \times r}$  be the matrix defined as:

$$(3.14) \quad A_{lm} = \left[ \sum_{i,j=1}^r a_{il} a_{jm} c_i c_j \prod_{k=2}^d \int_{\Omega_k} f_i^{(k)}(x_k) f_j^{(k)}(x_k) dx_k \right].$$

This matrix is symmetric and positive semidefinite by construction. Henceforth, there exists a complete orthonormal basis of  $\mathbb{R}^r$  that diagonalises it:  $A = W S W^T$ . The eigenfunctions of the correlation operator, which are also the left POD modes of the first unfolding are defined as:

$$(3.15) \quad u_i^{(1)}(x_1) = \sum_{l=1}^r z_l(x_1) W_{li}.$$

The associated singular value is  $\sigma_i^{(1)} = \sqrt{S_{ii}}$ .

The method is summarised in Algorithm 3.1.

---

**Algorithm 3.1** POD decomposition of the unfolding

---

**Input:**

$M^{(1)}(x_1, y_1)$

**Method:**

Compute POD of the fibers  $f_i^{(1)}$ , see Eq.(3.12)

Assemble the matrix A, see Eq.(3.14)

Diagonalise it:  $A = W S W^T$

Compute the modes  $u_i^{(1)}$  and the associated  $\sigma_i^{(1)}$

---

**3.4.2. Computational cost.** The computational cost of the method is detailed hereafter.

We start by describing the cost of the computation of the POD for one given unfolding, which is the most expensive operation of the CP-TT iteration. There are two distinct cases. If  $\mathcal{N}_i < r$ , then, the discrete counterpart of  $\kappa^{(i)}$  can be evaluated, and its eigenvalue decomposition computed directly. This leads to a cost of the form:

- Correlation assembly:  $\mathcal{O}\left(r^2(\sum_{j \neq i}^d c_I \mathcal{N}_j) + (2r-1)(r\mathcal{N}_i + \mathcal{N}_i^2)\right)$ .
- Eigenvalue decomposition:  $\mathcal{O}(\mathcal{N}_i^3)$ .

If  $\mathcal{N}_i > r$ , then, the method presented in the previous section is used and its cost reads:

- SVD of the fibers  $f^{(i)}$ :  $\mathcal{O}(\mathcal{N}_i r^2)$ .
- Assembly of A:  $\mathcal{O}\left(r^2(\sum_{j \neq i}^d c_I \mathcal{N}_j) + r^2(1 + 2(2r-1)) + r^3\right)$ .
- Eigenvalue decomposition of A:  $\mathcal{O}(r^3)$ .
- Computation of the left modes:  $\mathcal{O}((2r-1)\mathcal{N}_i)$ .

Overall, the cost of this stage is of order  $\mathcal{O}(r^2 d \mathcal{N} + \min\{\mathcal{N}_i^3, r^3\})$ .

When considering the CP-TT iteration, at the first step we compute  $d$  POD decompositions (these could be performed in a parallel way). After this stage, the tensor is tested against the mode corresponding to the chosen direction. The cost is  $\mathcal{O}(r \mathcal{N}_i)$ . Then, the POD of  $d - 1$  unfoldings is computed. Remark that the cost is the same as the one detailed above, with the exception that the operations involving the assembly of the matrices costs less (since we are working at dimension  $d - 1$ ).

*Remark.* The first stage of the computation is similar, in a way, to the HOSVD method ([6]). However, in the present method, only the first singular triplet is needed and actually used, which leads to a less expensive computation. Moreover, no storage of the core tensor is required.

**4. Numerical Experiments.** Once we have studied the properties of the method theoretically, we will proceed to do some numerical experiments based on function compression. In this section we can get an idea of how the method acts in practice. The proposed tests are made for functions which admit a Fourier decomposition involving a finite number of modes. In the tests we do a comparison between three methods, namely: ALS, ASVD and CPTT.

Let  $(x_1, \dots, x_d) \in \Omega = [0, 1]^d$ . Let  $(k_1, \dots, k_d) \in \mathbb{N}^d$  be the wave numbers. The function to be compressed is assumed to be given in CP format :

$$(4.1) \quad F(x_1, \dots, x_d) = \sum_{k_1=1}^{l_1} \sum_{k_2=1}^{l_2} \dots \sum_{k_d=1}^{l_d} a_{k_1 \dots k_d} \sin(\pi k_1 x_1) \times \dots \times \sin(\pi k_d x_d)$$

Let  $\beta > 0$ . The values of  $(l_i)_{1 \leq i \leq d}$  are chosen to be a family of independent random integers uniformly distributed between 1 and 6. Let  $(\alpha_{k_1 \dots k_d})_{k_1, \dots, k_d}$  be a family of independent random variables uniformly distributed in  $[-1, 1]$ . The amplitudes  $a_{k_1 \dots k_d}$  are defined as:

$$a_{k_1 \dots k_d} = \frac{\alpha_{k_1 \dots k_d}}{(\sqrt{k_1^2 + \dots + k_d^2})^\beta}.$$

For different random samples we obtain different functions  $F$  with the form presented in Eq.(4.1), the amplitude change and we obtain different functions preserving the shape of  $F$ . The value of the parameter  $\beta$  determines the regularity of the functions in the sense that for  $s \in \mathbb{N}$  it holds:

$$(4.2) \quad \beta > \frac{d}{2} + s \Rightarrow F \in H^s(\Omega).$$

We are testing how the three methods behave for the compression of 32 different functions generated by the random procedure described above for values of  $d$  ranging from 4 to 16. ALS and ASVD are both fixed point based methods, the tolerance for the fixed point has been set as  $1.0 \times 10^{-4}$  and the maximum number of iterations of the method  $it_{max} = 100$ . A uniform discretization grid of  $\Omega$  with 25 degrees of freedom per direction is used for the discretization of  $F$ .

**4.1. Results for functions in  $L^2(\Omega)$ .** Firstly, the method is applied to some test cases in which the functions belong to  $L^2(\Omega)$ , namely when the value of the parameter  $\beta$  is chosen to be equal to  $\frac{d}{2} + 0.1$ . On the left hand side of Figure 2 (respectively Figure 3 and Figure 4), the  $L^2$  norm of the difference between the exact

function  $F$  and its approximation computed by one of the three algorithms is plotted as a function of the rank of the approximation, where  $d = 4$  (respectively  $d = 12$  and  $d = 16$ ).

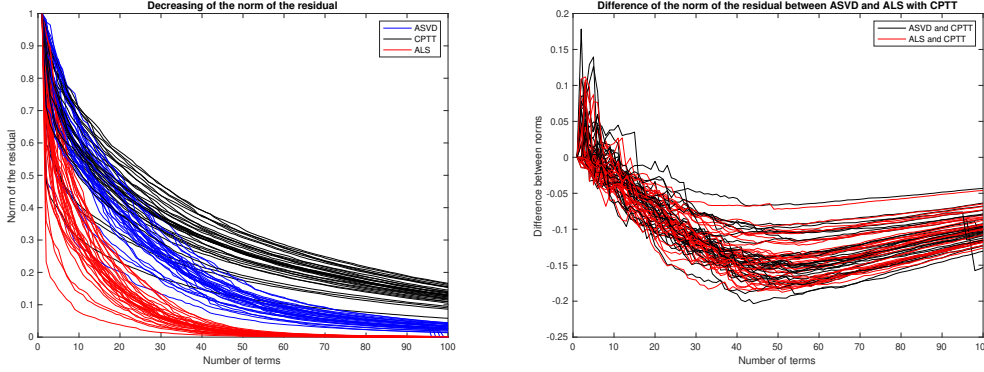


FIG. 2. Plot associated to functions described in Eq.(4.1). On the left, the decreasing of the  $L^2$  norm of the difference between the exact function  $F$  and its approximation in the 4-dimensional tensor of the 32  $L^2$  functions,  $\beta = 2.1$ , using ALS in red, ASVD in blue and the CP-TT method in black. On the right, the difference between both norms the difference between the exact function  $F$  and its approximation  $\|R\|_{ALS} - \|R\|_{CP-TT}$  in red and  $\|R\|_{ASVD} - \|R\|_{CP-TT}$  in black.

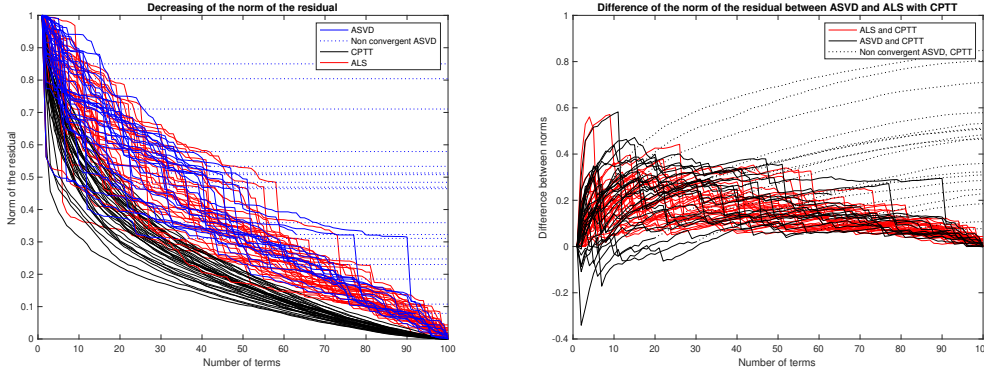


FIG. 3. Plot associated to functions described in Eq.(4.1). On the left, the decreasing of the  $L^2$  norm of the difference between the exact function  $F$  and its approximation in the 12-dimensional tensor of the 32  $L^2$  functions,  $\beta = 6.1$ , using ALS in red, ASVD in blue and the CP-TT method in black. The dotted line shows as a constant the last point that converged on the ASVD method. On the right, the difference between both norms the difference between the exact function  $F$  and its approximation  $\|R\|_{ALS} - \|R\|_{CP-TT}$  in red and  $\|R\|_{ASVD} - \|R\|_{CP-TT}$  in black.

Whereas at dimension 4 (Figure 2) ALS outperforms both ASVD and CP-TT, at dimensions  $d = 12$  and  $d = 16$  (figures 3 and 4 respectively) CP-TT is featured by a better behavior. In particular, the compression rate is better on average and the decrease of the norm of the error with respect to the rank of the approximation is more regular. To better highlight this, on the right-hand side of figures 2, 3 and 4 we

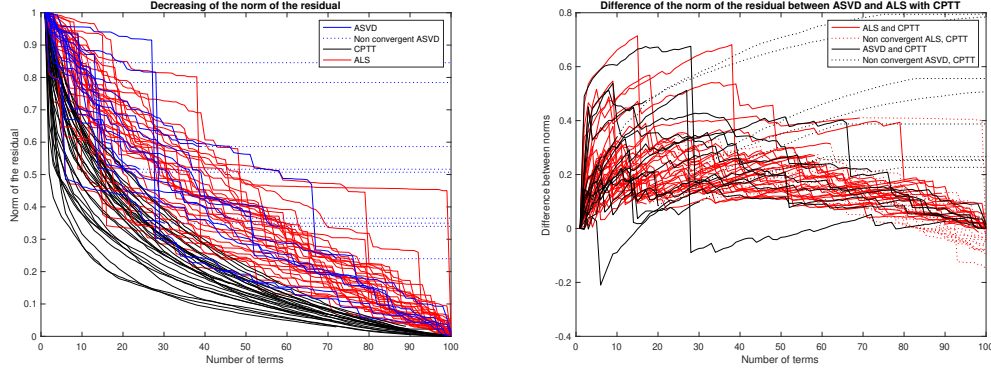


FIG. 4. Plot associated to functions described in Eq.(4.1). On the left, the decreasing of the  $L^2$  norm of the difference between the exact function  $F$  and its approximation in the 16-dimensional tensor of the 32  $L^2$  functions,  $\beta = 8.1$ , using ALS in red, ASVD in blue and the CP-TT method in black. The dotted line shows as a constant the last point that converged on the ASVD and CP-TT methods. On the right, the difference between both norms the difference between the exact function  $F$  and its approximation  $\|R\|_{ALS} - \|R\|_{CP-TT}$  in red and  $\|R\|_{ASVD} - \|R\|_{CP-TT}$  in black.

plot the difference of the error norms of ALS and CP-TT and ASVD and CP-TT as function of the rank of the approximation.

In the previous test a relaxed version of ALS (standard ALS with more flexibility in the scalar products in order to optimize the coefficients) was used that is observed to converge more often than the standard version of ALS. In Figure 5, the relaxed and the non relaxed version of the ALS algorithm in the 12-dimensional case are compared. Both versions were observed to yield similar compression behavior.

The results for  $L^2(\Omega)$  functions, when changing the dimension, are reported in Table 1. In particular, we show the mean and the standard deviation of the error (on the 32 random functions) when the approximation rank is (25, 50, 75) for ALS, ASVD and CP-TT. From the results, we can see that for low-order tensors (*e.g.* when  $d = 4$ ) ALS has better performances, whereas for higher order tensors CP-TT outperforms the other methods both in terms of mean and standard deviation (suggesting that it somehow enjoys a certain stability).

**4.2. Results for functions in  $H^1(\Omega)$ .** The method is applied to test cases in which the functions belong to  $H^1(\Omega)$ , namely when the value of the parameter  $\beta$  is chosen to be equal to  $\frac{d}{2} + 1.1$ . The Figures 6,7,8 are the counterpart of Figures 2,3,5 introduced and commented in the previous section.

The results obtained on  $H^1(\Omega)$  functions are equivalent to the ones shown for  $L^2(\Omega)$  functions, showing that the decrease in the error norm with the approximation rank is quite regular in CP-TT and behaves in a quite stable way also for higher order tensors.

The Table 2 collects the results for functions in  $H^1(\Omega)$ .

*Remark:.* Despite the fact that one iteration of CP-TT is in general more costly from a computational standpoint with respect to an ALS iteration, at constant target error we computational cost for CP-TT is more or less equivalent (it requires a smaller

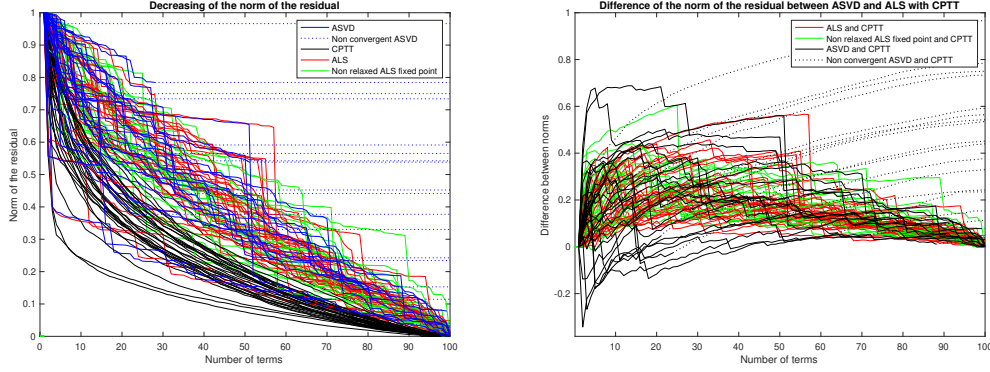


FIG. 5. Plot associated to functions described in Eq.(4.1). On the left, the decreasing of the  $L^2$  norm of the difference between the exact function  $F$  and its approximation in the 12-dimensional tensor of the 32  $L^2$  functions,  $\beta = 6.1$ , using ALS in red, ALS without the relaxed fixed point in green, the ASVD in blue and the CP-TT method in black. The dotted line shows as a constant the last point that converged on the ALS method. On the right, the difference between both norms the difference between the exact function  $F$  and its approximation  $\|R\|_{ALS} - \|R\|_{CP-TT}$  in red and  $\|R\|_{ASVD} - \|R\|_{CP-TT}$  in black.

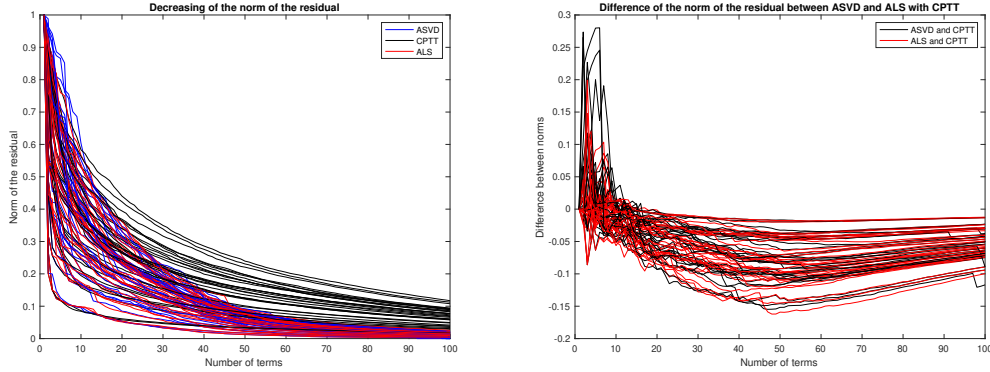


FIG. 6. Plot associated to functions described in Eq.(4.1). On the left, the decreasing of the  $L^2$  norm of the difference between the exact function  $F$  and its approximation in the 4-dimensional tensor of the 32  $H^1$  functions,  $\beta = 3.1$ , using ALS in red, the ASVD method in blue and the CP-TT method in black. On the right, the difference between both norms of the difference between the exact function  $F$  and its approximation  $\|R\|_{ALS} - \|R\|_{CP-TT}$  in red and  $\|R\|_{ASVD} - \|R\|_{CP-TT}$  in black.

rank for higher order tensors and the ALS fix point needs more time to converge). Overall, the numerical experiments results are encouraging, showing that CP-TT is a valuable alternative for high-order tensor approximation.

**5. Conclusions and perspectives.** In the present work, a method has been proposed to compute, given a tensor, its CP approximation. By leveraging the properties of the TT-SVD algorithm, it is possible to compute a CP decomposition in a stable way, also in the case in which we look for a generic rank- $k$  update. Albeit the

Dimension ( $d$ )	Rank ( $r$ )	Mean			Standard deviation		
		ALS	CPTT	ASVD	ALS	CPTT	ASVD
4	25	0.2942	0.3826	0.3118	0.0702	0.0850	0.0843
	50	0.1082	0.2433	0.1257	0.0326	0.0568	0.0664
	75	0.0508	0.1681	0.0689	0.0180	0.0408	0.0666
6	25	0.4479	0.3771	0.4806	0.1099	0.0826	0.1074
	50	0.2705	0.1982	0.2883	0.0752	0.0485	0.0675
	75	0.1232	0.0806	0.1369	0.0325	0.0252	0.0368
8	25	0.5341	0.3707	0.5532	0.1183	0.0592	0.1238
	50	0.3060	0.1909	0.3415	0.0722	0.0341	0.0932
	75	0.1592	0.0682	0.1807	0.0435	0.0160	0.0625
10	25	0.5023	0.3598	0.5451	0.0879	0.0643	0.1055
	50	0.3191	0.1826	0.3797	0.0643	0.0342	0.0774
	75	0.1714	0.0655	0.2792	0.0453	0.0162	0.1265
12	25	0.5170	0.3246	0.5639	0.1117	0.0576	0.1250
	50	0.3249	0.1623	0.4206	0.0824	0.0286	0.1579
	75	0.1543	0.0579	0.3498	0.0369	0.0113	0.2057
14	25	0.4443	0.2336	0.4783	0.1712	0.1064	0.1585
	50	0.2407	0.1004	0.3307	0.0937	0.0588	0.1737
	75	0.1411	0.0321	0.2230	0.0541	0.0235	0.1821
16	25	0.5529	0.3160	0.6150	0.1305	0.0818	0.1656
	50	0.3487	0.1448	0.4424	0.0849	0.0389	0.1942
	75	0.1946	0.0616	0.3678	0.0905	0.0289	0.2354

TABLE 1

Mean and standard deviation of the decreasing of the norm of the residual for the 32 random  $L^2(\Omega)$  functions. The results shown are for different number of terms on each of the dimensions tested.

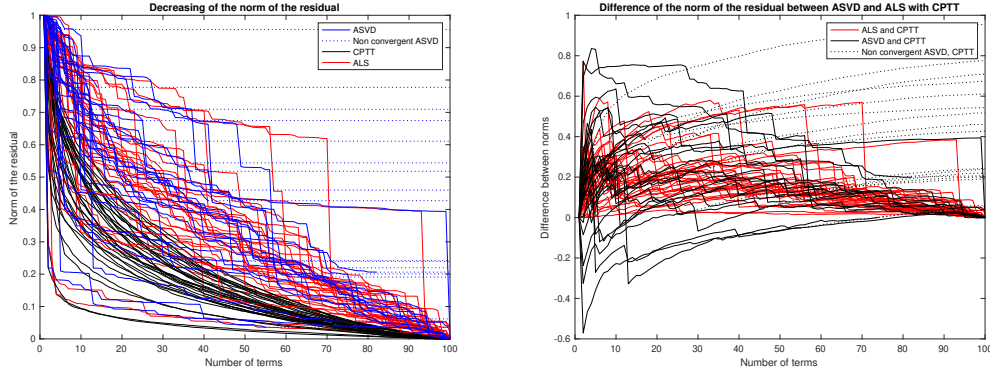


FIG. 7. Plot associated to functions described in Eq.(4.1). On the left, the decreasing of the  $L^2$  norm of the difference between the exact function  $F$  and its approximation in the 12-dimensional tensor of the 32  $H^1$  functions,  $\beta = 7.1$ , using ALS in red, the ASVD method in blue and the CP-TT method in black. On the right, the difference between both norms of the difference between the exact function  $F$  and its approximation  $\|R\|_{ALS} - \|R\|_{CP-TT}$  in red and  $\|R\|_{ASVD} - \|R\|_{CP-TT}$  in black.

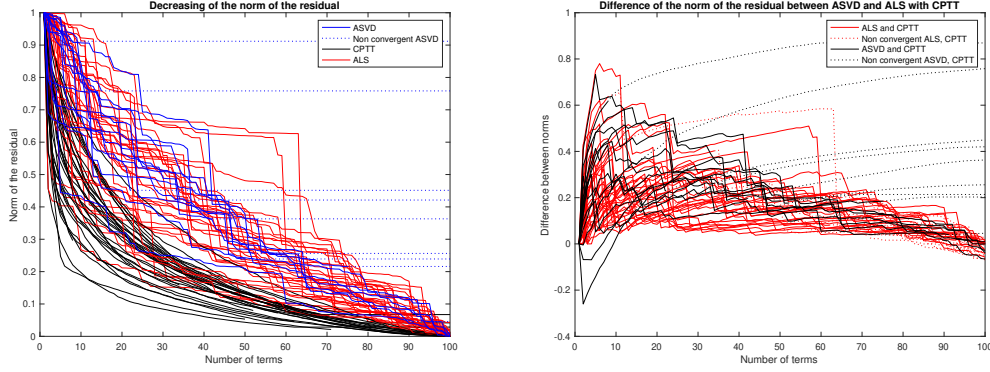


FIG. 8. Plot associated to functions described in Eq.(4.1). On the left, the decreasing of the  $L^2$  norm of the difference between the exact function  $F$  and its approximation in the 16-dimensional tensor of the  $32 H^1$  functions,  $\beta = 9.1$  using ALS in red, the ASVD method in blue and the CP-TT method in black. On the right, the difference between both norms of the difference between the exact function  $F$  and its approximation  $\|R\|_{ALS} - \|R\|_{CP-TT}$  in red and  $\|R\|_{ASVD} - \|R\|_{CP-TT}$  in black. Both tests are made for 100 terms.

fact that the iterations heavily rely on the TT-SVD algorithm, the proposed strategy does not require to fix *a priori* the order of the variables, but it determines it through an optimization step.

Several numerical experiments are proposed in order to assess the properties of the method and compare it with ALS and ASVD methods (which are based on similar principles). The experiments suggest that the proposed method can be a valuable tool to compute a CP decomposition in high-dimensional settings, for which the method better behaved with respect to the above mentioned alternatives. The difference in terms of sparsity between CP-TT and ALS and ASVD methods is large, and even larger for more regular functions, presenting all the CP-TT functions much similar behavior. In order to predict the results without computing them, this has to be taken into account.

The main perspectives of the present work consists in using CP-TT in the solution of multi-linear problems and to investigate how some of the defining steps of this method could be exploited in other tensor formats.

**Acknowledgments.** Virginie Ehrlacher acknowledges support from the ANR COMODO project (ANR-19-CE46-0002).

Damiano Lombardi acknowledges support from the ANR ADAPT project (ANR-18-CE46-0001).

This publication is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme – Grant Agreement  $n^\circ$  810367.

## REFERENCES

- [1] R. BADEAU AND R. BOYER, *Fast multilinear singular value decomposition for structured tensors*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 1008–1021.
- [2] G. BEYLKIN AND M. J. MOHLENKAMP, *Algorithms for numerical analysis in high dimensions*, SIAM J. SCI. COMPUT., 26 (2005), pp. 2133–2159.

Dimension ( $d$ )	Rank ( $r$ )	Mean			Standard deviation		
		ALS	CPTT	ASVD	ALS	CPTT	ASVD
4	25	0.1722	0.2261	0.1759	0.0643	0.2261	0.1759
	50	0.0572	0.1382	0.0590	0.0220	0.1382	0.0232
	75	0.0252	0.0948	0.0262	0.0103	0.0948	0.0110
6	25	0.3741	0.2938	0.4171	0.1158	0.0942	0.1341
	50	0.2037	0.1507	0.2281	0.0655	0.0523	0.0791
	75	0.0851	0.0579	0.1045	0.0334	0.0233	0.0493
8	25	0.3676	0.2560	0.3977	0.1361	0.0905	0.1517
	50	0.2136	0.1229	0.2413	0.0807	0.0451	0.1023
	75	0.1046	0.0455	0.1145	0.0437	0.0195	0.0631
10	25	0.4574	0.3737	0.4753	0.1235	0.1548	0.1817
	50	0.2613	0.3483	0.3193	0.0809	0.1825	0.1648
	75	0.1168	0.3332	0.2352	0.0628	0.2034	0.1865
12	25	0.4634	0.2505	0.5182	0.1681	0.0842	0.2116
	50	0.2889	0.1141	0.3922	0.1421	0.0384	0.2170
	75	0.1278	0.0382	0.3144	0.0671	0.0126	0.2502
14	25	0.5943	0.2169	0.4386	0.2043	0.1262	0.2014
	50	0.2841	0.0779	0.3132	0.1277	0.0686	0.1915
	75	0.1422	0.0244	0.2021	0.0814	0.0227	0.2192
16	25	0.4598	0.2460	0.5543	0.1496	0.0726	0.1603
	50	0.2861	0.1108	0.3936	0.1268	0.0348	0.2022
	75	0.1395	0.0438	0.3181	0.0552	0.0153	0.2477

TABLE 2

Mean and standard deviation of the decreasing of the norm of the residual for the 32 random  $H^1(\Omega)$  functions. The results shown are for different number of terms on each of the dimensions tested.

- [3] D. BIGONI, A. P. ENGSIG-KARUP, AND Y. M. MARZOUK, *Spectral tensor-train decomposition*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2405–A2439.
- [4] A. CICHOCKI, N. LEE, I. OSELEDETS, A. P. AMD Q. ZHAONAND, AND D. MANDIC, *Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions*, Now Publishers Inc., 35 (2016), <https://doi.org/10.1561/22000000059>.
- [5] A. CICHOCKI, N. LEE, I. OSELEDETS, A.-H. PHAN, Q. ZHAO, AND D. P. MANDIC, *Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions*, Foundations and Trends® in Machine Learning, 9 (2016), pp. 249–429.
- [6] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM journal on Matrix Analysis and Applications, 21 (2000), pp. 1253–1278.
- [7] V. DE SILVA AND L.-H. LIM, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), p. 1084–1127, <https://doi.org/10.1137/06066518X>.
- [8] V. DE SILVA AND L.-H. LIM, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 1084–1127.
- [9] I. DOMANOV AND L. LATHAUWER, *On uniqueness and computation of the decomposition of a tensor into multilinear rank-(1,  $l_r$ ,  $l_r$ ) terms*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 747–803.
- [10] I. DOMANOV AND L. D. LATHAUWER, *Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition*, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 636–660, <https://doi.org/10.1137/130916084>.
- [11] M. ESPIG, W. HACKBUSCH, AND A. KHACHATRYAN, *On the convergence of alternating least squares optimisation in tensor format representations*, arXiv preprint arXiv:1506.00062, (2015).
- [12] S. FRIEDLAND, V. MEHRMANN, R. PAJAROLA, AND S. K. SUTER, *On best rank one approxima-*

- tion of tensors, Numerical Linear Algebra with Applications, 20 (2013), pp. 942–955.
- [13] S. FRIEDLAND AND G. OTTAVIANI, *The number of singular vector tuples and uniqueness of best rank-one approximation of tensors*, Found Comput Math, 14 (2014), p. 1209–1242, <https://doi.org/10.1007/s10208-014-9194-z>.
  - [14] L. GRASEDYCK, D. KRESSNER, AND C. TOBLER, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen, 36 (2013), pp. 53–78.
  - [15] O. KAYA AND B. UÇAR, *Scalable sparse tensor decompositions in distributed memory systems*, in SC’15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2015, pp. 1–11.
  - [16] T. KOLDA AND B. BADER, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500, <https://doi.org/10.1137/07070111X>.
  - [17] T. G. KOLDA, *Tensor decomposition: A mathematical tool for data analysis.*, tech. report, Sandia National Lab.(SNL-CA), Livermore, CA (United States), 2018.
  - [18] T. G. KOLDA AND J. SUN, *Scalable tensor decompositions for multi-aspect data mining*, in 2008 Eighth IEEE international conference on data mining, IEEE, 2008, pp. 363–372.
  - [19] K. KOUR, S. DOLGOV, M. STOLL, AND P. BENNER, *Efficient structure-preserving support tensor train machine*, arXiv preprint arXiv:2002.05079, (2020).
  - [20] I. OSELEDETS, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.
  - [21] I. OSELEDETS AND E. TYRTYSHNIKOV, *Breaking the curse of dimensionality, or how to use svd in many dimensions*, SIAM Journal on Matrix Analysis and Applications, 31 (2009), p. 1084–1127, <https://doi.org/10.1137/090748330>.
  - [22] I. V. OSELEDETS, M. V. RAKHUBA, AND A. USCHMAJEV, *Alternating least squares as moving subspace correction*, SIAM Journal on Numerical Analysis, 56 (2018), pp. 3459–3479.
  - [23] A.-H. PHAN, K. SOBOLEV, K. SOZYKIN, D. ERMILOV, J. GUSAK, P. TICHAVSKY, V. GLUKHOV, I. OSELEDETS, AND A. CICHOCKI, *Stable low-rank tensor decomposition for compression of convolutional neural network*, ECCV2020, (2020), <https://doi.org/arXiv:2008.05441>.
  - [24] A.-H. PHAN, P. TICHAVSKÝ, AND A. CICHOCKI, *Fast alternating ls algorithms for high order candecomp/parafac tensor factorizations*, IEEE Transactions on Signal Processing, 61 (2013), pp. 4834–4846.
  - [25] M. RAJAH, P. COMON, AND R. HARSMAN, *Enhanced line search: A novel method to accelerate parafac*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), <https://doi.org/10.1137/06065577>.
  - [26] M. RAKHUBA AND I. OSELEDETS, *Calculating vibrational spectra of molecules using tensor train decomposition*, The Journal of Chemical Physics, 145 (2016), p. 124101.
  - [27] T. ROHWEDDER AND A. USCHMAJEV, *On local convergence of alternating schemes for optimization of convex problems in the tensor train format*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 1134–1162.
  - [28] A. USCHMAJEV, *Local convergence of the alternating least squares algorithm for canonical tensor approximation*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 639–652.
  - [29] M. VANDECAPPELLE, N. VERVLIT, AND L. DE LATHAUWER, *Nonlinear least squares updating of the canonical polyadic decomposition*, in 2017 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 663–667.
  - [30] N. VERVLIT, O. DEBALS, AND L. DE LATHAUWER, *Exploiting efficient representations in large-scale tensor decompositions*, SIAM Journal on Scientific Computing, 41 (2019), pp. A789–A815.
  - [31] X. WANG, C. NAVASCA, AND S. KINDERMANN, *On accelerating the regularized alternating least square algorithm for tensors*, arXiv preprint arXiv:1507.04721, (2015).
  - [32] X. WANG, L. T. YANG, Y. WANG, X. LIU, Q. ZHANG, AND M. J. DEEN, *A distributed tensor-train decomposition method for cyber-physical-social services*, ACM Transactions on Cyber-Physical Systems, 3 (2019), pp. 1–15.