Numerical Strategies for the Galerkin-Proper Generalized Decomposition Method

A. Falcó[a], L. Hilario[a], N. Montés[b], M. C. Mora[c]

[a]*Departamento de Ciencias, Físicas, Matemáticas y de la Computación, Universidad CEU Cardenal Herrera San Bartolomé 55 46115 Alfara del Patriarca (Valencia), Spain.*
[b]*Departamento de Ingeniería de la Edificación y Producción Industrial, Universidad CEU Cardenal Herrera San Bartolomé 55 46115 Alfara del Patriarca (Valencia), Spain.*
[c]*Departamento de Ingeniería Mecánica y Construcción, Universitat Jaume I, Avd. Vicent Sos Baynat s/n 12071 Castellón, Spain*

Abstract

The Proper Generalized Decomposition or, in short, PGD is a tensor decomposition based technique to solve PDE problems. It reduces calculation and storage cost drastically and presents some similarities with the Proper Orthogonal Decomposition, in short POD. In this work, we propose an efficient implementation to improve the convergence of the PGD, towards the numerical solution of a discretized PDE problem, when the associated matrix is Laplacian-like.

Keywords: Numerical tensor calculus, Tensor Banach space, Proper Generalized Decomposition, Laplacian-like matrix.

1. Introduction

The Proper Generalized Decomposition PGD appears as a natural extension of the Proper Orthogonal Decomposition POD. It is well-known that POD framework is based in the SVD methodology. On the other hand, the SVD is closely related, in finite dimension, with the Eckart-Young Theorem [12] and in infinite dimension with the classical result of Schmidt [15]. Both results contain one of the main ingredients of PGD framework: the existence of a best approximation by using a tensor decomposition. It is important to point out, that both results are true only for tensors of order two, because the milestone of the proof is the existence of a best rank-n approximation for this class of tensor. Unfortunately, in [11], it has been proved that tensors of order 3 or higher can fail to have best rank-n approximation, that is, it is an ill-posed problem. In consequence, as shown in [6] only rank-one approximations are available.

Recently, Falcó and Hackbusch [5] have proved the existence of a best approximation for tensor representations based in subspaces (see also [9]). This fact allows to extend the PGD to other type of tensor decompositions.

There exists several approaches to the numerical analysis of PGD. Both combine the existence of a best approximation and a greedy algorithm. The idea of using greedy algorithms was introduced by Ammar, Mokdad, Chinesta and Keunings [2], in the context of high-dimensional PDEs. It can be applied to several problems including, among others, the stationary FokkerPlanck equation of the FENE bead-spring chain model [13]. In Ammar, Mokdad, Chinesta and Keunings [3], the approach is extended to time-dependent problems. One should note, however, the method in [2] can be interpreted as a greedy algorithm for solving elliptic problems associated with self-adjoint operators, because it is equivalent to a minimization problem related with an energy functional (see Le Bris, Lelièvre and Maday [4] for elliptic problems associated with self-adjoint operators and for general elliptic ones, see Falcó and Nouy [6],). The extension of the method to more general problems, like the non-stationary FokkerPlanck equation, is done merely by analogy. In particular, its convergence, in general, is not guaranteed (a recent result in this approach is given in the paper of Figueroa and Süli [7]). A modification of the low-rank separation method, introduced in Ammar, Chinesta and Falcó [1], where one minimizes on each iteration an L^2 norm of the residual rather than a norm of the error. Such a method can be linked with the greedy algorithms of DeVore and Temlyakov [10] for essentially any PDE.

The main goal of this paper is to study the non-convex minimization problem included in the PGD algorithm. In particular we propose an efficient implementation for a class of discreticed elliptic problems. The paper is organized as follows. In the next section we introduce some preliminary definitions and notation in order to explain the Galerkin-PGD method in a Hilbert tensor space framework. Section 3 is devoted to the numerical strategies to solve the PGD minimization problem for full-rank linear systems. Finally, in 4 we study an optimal implementation for the class of Laplacian-like matrices. Moreover, some numerical examples are given.

2. The Galerkin-PGD method

Concerning the definition of the algebraic tensor space $_a\bigotimes_{j=1}^d V_j$ generated from vector spaces V_j $(1 \le j \le d)$, we refer to Greub [8]. As the underlying field we choose $\mathbb{R}$, but the results hold also for $\mathbb{C}$. The suffix 'a' in $_a\bigotimes_{j=1}^d V_j$ refers to the 'algebraic' nature. By definition, all elements of

$$\mathbf{V} := {_a}\bigotimes_{j=1}^d V_j$$

are *finite* linear combinations of elementary tensors $\mathbf{v} = \bigotimes_{j=1}^d v_j$ $(v_j \in V_j)$. Next, we introduce the following class of Hilbert spaces.

Definition 2.1. *We say that* $\mathbf{V}_{\|\cdot\|}$ *is a* Hilbert tensor space *if there exists an algebraic tensor space* $\mathbf{V}$ *and a norm* $\|\cdot\|$ *on* $\mathbf{V}$ *such that* $\mathbf{V}_{\|\cdot\|}$ *is the completion of* $\mathbf{V}$ *with respect a given norm* $\|\cdot\|$*, i.e.,*

$$\mathbf{V}_{\|\cdot\|} := {_{\|\cdot\|}}\bigotimes_{j=1}^d V_j = \overline{{_a}\bigotimes_{j=1}^d V_j}^{\|\cdot\|}.$$

The following notation and definitions, introduced in [5], will be useful. Let $\mathcal{I} := \{1, \ldots, d\}$ be the index set of the 'spatial directions'. In the sequel, the index sets $\mathcal{I}\backslash\{j\}$ will appear. Here, we use the abbreviations

$$\mathbf{V}_{[j]} := {}_a\bigotimes_{k \neq j} V_k \, , \qquad \text{where} \quad \bigotimes_{k \neq j} \text{ means } \bigotimes_{k \in \mathcal{I}\backslash\{j\}} \, , \tag{1}$$

Similarly, elementary tensors $\bigotimes_{k \neq j} v^{(j)}$ are denoted by $\mathbf{v}_{[j]}$.

Let $\mathbf{V}_{\|\cdot\|}$ be a tensor Hilbert space, $a : \mathbf{V}_{\|\cdot\|} \times \mathbf{V}_{\|\cdot\|} \to \mathbb{R}$ be a bilinear form, and $\boldsymbol{\varphi} \in \mathbf{V}_{\|\cdot\|}^*$. We consider the problem

$$\mathbf{u} \in \mathbf{V}_{\|\cdot\|}, \, a(\mathbf{u}, \mathbf{v}) = \boldsymbol{\varphi}(\mathbf{v}) \text{ for all } \mathbf{v} \in \mathbf{V}_{\|\cdot\|}. \tag{2}$$

We assume that $a(\cdot, \cdot)$ is bounded,

$$|a(\mathbf{u}, \mathbf{v})| \leq M \|\mathbf{u}\| \|\mathbf{v}\| \text{ for all } \mathbf{u}, \mathbf{v} \in \mathbf{V}_{\|\cdot\|}, \tag{3}$$

and $\mathbf{V}_{\|\cdot\|}$-elliptic,

$$a(\mathbf{v}, \mathbf{v}) \geq c_0 \|\mathbf{v}\|^2 \text{ for all } \mathbf{v} \in \mathbf{V}_{\|\cdot\|}, \tag{4}$$

for some positive constants M and c_0. Classical Galerkin Method consists of constructing an approximate solution in an N-dimensional subspace $\mathbf{V}_N$ of the tensor Hilbert space $\mathbf{V}_{\|\cdot\|}$. Then we project problem (2) onto $\mathbf{V}_N$:

$$\mathbf{u} \in \mathbf{V}_N, \, a(\mathbf{u}, \mathbf{v}) = \boldsymbol{\varphi}(\mathbf{v}) \text{ for all } \mathbf{v} \in \mathbf{V}_N. \tag{5}$$

We introduce the operator $A : \mathbf{V}_{\|\cdot\|} \longrightarrow \mathbf{V}_{\|\cdot\|}$ associated with $a(\cdot, \cdot)$, and defined by

$$a(\mathbf{u}, \mathbf{v}) = \langle A\mathbf{u}, \mathbf{v}\rangle \tag{6}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbf{V}_{\|\cdot\|}$. We also introduce the element $\mathbf{f} \in \mathbf{V}_{\|\cdot\|}$ associated with ℓ and defined by

$$\ell(\mathbf{v}) = \langle \mathbf{f}, \mathbf{v}\rangle \tag{7}$$

for all $\mathbf{v} \in \mathbf{V}_{\|\cdot\|}$. The existence of A and $\mathbf{f}$ is ensured by the Riesz representation theorem [14]. Problem (2) can be rewritten in an operator form:

$$A\mathbf{u} = \mathbf{f} \tag{8}$$

We further assume the following property on A: for all $\mathbf{v} \in \mathbf{V}_{\|\cdot\|}$,

$$\exists c > 0 \quad \text{such that} \quad \|A\mathbf{v}\| \geq c\|\mathbf{v}\| \tag{9}$$

From properties of A and its adjoin A^*, A^*A is a self-adjoint continuous and V-elliptic operator. It then defines an inner product on $\mathbf{V}_{\|\cdot\|}$, denoted $\langle\cdot, \cdot\rangle_{A^*A} := \langle A\cdot, A\cdot\rangle$, with associated norm $\|\cdot\|_{A^*A}$ which is equivalent to the norm $\|\cdot\|$. Formulation (5) is equivalent to the following minimal residual formulation:

$$\mathbf{u}_N = \arg\min_{\mathbf{v}\in\mathbf{V}_N} \|\mathbf{f} - A\mathbf{v}\|^2 = \arg\min_{\mathbf{v}\in\mathbf{V}_N} \|A^{-1}\mathbf{f} - \mathbf{v}\|^2_{A^*A} \tag{10}$$

The PGD-Galerkin Method is based in the fact that in a tensor Hilbert space a typical representation format is the tensor subspace or Tucker format

$$\mathbf{u} = \sum_{\mathbf{i}\in\mathbf{I}} \mathbf{a_i} \bigotimes_{j=1}^{d} b_{i_j}^{(j)}, \tag{11}$$

where $\mathbf{I} = I_1 \times \ldots \times I_d$ is a multi-index set with $I_j = \{1, \ldots, r_j\}$, $r_j \leq \dim(V_j)$, $b_{i_j}^{(j)} \in V_j$ ($i_j \in I_j$) are basis vectors, and $\mathbf{a_i} \in \mathbb{R}$. Here, i_j are the components of $\mathbf{i} = (i_1, \ldots, i_d)$. The data size is determined by the numbers r_j collected in the tuple $\mathbf{r} := (r_1, \ldots, r_d)$. The set of all tensors representable by (11) with fixed $\mathbf{r}$ is

$$\mathcal{T}_{\mathbf{r}}(\mathbf{V}) := \left\{ \mathbf{v} \in \mathbf{V} : \begin{array}{l} \text{there are subspaces } U_j \subset V_j \text{ such that} \\ \dim(U_j) = r_j \text{ and } \mathbf{v} \in \mathbf{U} := {}_a\bigotimes_{j=1}^{d} U_j . \end{array} \right\} \quad (12)$$

Here, it is important that the description (11) with the vectors $b_i^{(j)}$ can be replaced by the generated subspace $U_j = \text{span}\{b_i^{(j)} : i \in I_j\}$. Note that $\mathcal{T}_{\mathbf{r}}$ is neither a subspace of $\mathbf{V}$ nor a convex set. However, $\text{span}\,\mathcal{T}_{\mathbf{r}}(\mathbf{V})$ is dense in $\mathbf{V}$, for all $r_i \geq 1$ where $1 \leq i \leq d$, and hence in $\mathbf{V}_{\|\cdot\|}$.

In [5] some conditions are given in order to show that $\mathcal{T}_{\mathbf{r}}(\mathbf{V})$ is weakly closed in $\mathbf{V}_{\|\cdot\|}$. Under this assumption every element in $\mathbf{V}_{\|\cdot\|}$ has a best approximation in $\mathcal{T}_{\mathbf{r}}(\mathbf{V})$. Then the PGD-Galerkin method appears as follows:

$$\mathbf{u_r} \in \arg\min_{\mathbf{v} \in \mathcal{T}_{\mathbf{r}}(\mathbf{V})} \|\mathbf{f} - A\mathbf{v}\|^2 = \arg\min_{\mathbf{v} \in \mathcal{T}_{\mathbf{r}}(\mathbf{V})} \|A^{-1}\mathbf{f} - \mathbf{v}\|_{A^*A}^2. \quad (13)$$

Then in a similar way that in Theorem 14 in [6] it can be shown the following result.

Theorem 2.2. *Let $\mathbf{V}_{\|\cdot\|}$ be a tensor Hilbert space, $a : \mathbf{V}_{\|\cdot\|} \times \mathbf{V}_{\|\cdot\|} \to \mathbb{R}$ be a bilinear form bounded and $\mathbf{V}_{\|\cdot\|}$-elliptic, and $\varphi \in \mathbf{V}_{\|\cdot\|}^*$. Assume that $A : \mathbf{V}_{\|\cdot\|} \longrightarrow \mathbf{V}_{\|\cdot\|}$ defined by (6) satisfies (9) and $\mathcal{T}_{\mathbf{r}}(\mathbf{V})$ is weakly closed in $\mathbf{V}_{\|\cdot\|}$. Let $\mathbf{u_r}^{(0)} = \mathbf{0}$, and for each $n \geq 1$ take*

$$\mathbf{e}_{n-1} = \mathbf{f} - A\mathbf{u_r}^{(n-1)}, \quad (14)$$

$$\textit{compute } \mathbf{w}^{(n)} \in \arg\min_{\mathbf{v} \in \mathcal{T}_{\mathbf{r}}(\mathbf{V})} \|\mathbf{e}_{n-1} - A\mathbf{v}\|, \quad (15)$$

$$\mathbf{u_r}^{(n)} = \mathbf{u_r}^{(n-1)} + \mathbf{w}^{(n)}, \quad (16)$$

$$\textit{update } n. \quad (17)$$

Then

$$\lim_{n \to \infty} \mathbf{u_r}^{(n)} = \mathbf{u} = \arg\min_{\mathbf{v} \in \mathbf{V}_{\|\cdot\|}} \|\mathbf{f} - A\mathbf{v}\|.$$

3. Strategies to solve the minimization problem (15)

From now on, we will concentrate our efforts to solve numerically the non-convex minimization problem (15). To this end we introduce the following notation. For each $\mathbf{m} = (m_1, \ldots, m_d) \in \mathbb{N}^d$ we define $\ell(\mathbf{m}) := \prod_{j=1}^{d} m_j$ and $|\mathbf{m}| := \sum_{j=1}^{d} m_j$.

Since in real life problems we usually work in a tensor space ${}_a\bigotimes_{j=1}^{d} \mathbb{R}^{m_j} = \mathbb{R}^{\ell(\mathbf{m})}$, for example by using a Finite Element o Finite Difference method, we identify a tensor $\mathbf{u} \in {}_a\bigotimes_{j=1}^{d} \mathbb{R}^{m_j}$ with a vector $u \in \mathbb{R}^{\ell(\mathbf{m})}$. Then given a full-rank matrix $A \in \mathbb{R}^{\ell(\mathbf{m}) \times \ell(\mathbf{m})}$ and $b \in \mathbb{R}^{\ell(\mathbf{m})}$, we would to compute the solution

4

$A^{-1}b$, of the linear system $Au = b$, by using (14)-(17). The most simple case is to consider $\mathbf{r} = \mathbf{1} := (1, \ldots, 1)$, that is, the set of rank-one tensors

$$\mathcal{T}_1 \left({}_a \bigotimes_{j=1}^{d} \mathbb{R}^{m_j} \right) = \left\{ \bigotimes_{j=1}^{d} v_j : v_j \in \mathbb{R}^{m_j} \right\}$$

where $\bigotimes$ denotes the Kronecker product. Thus, at the end of n-iteration in (14)-(17) we have a vector

$$u_{\mathbf{r}}^{(n)} = \sum_{k=1}^{n} \bigotimes_{j=1}^{d} c_j^{(k)}.$$

From Theorem 2.2 we obtain that $u_{\mathbf{r}}^{(n)} \to A^{-1}b$ as $n \to \infty$.

In order to solve the minimization problem (15) we use (13), and in consequence we consider the map

$$\mathcal{J} : \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_k} \longrightarrow \mathbb{R}_+$$

defined by

$$\mathcal{J}(x_1, \ldots, x_d) := \frac{1}{2} \left(\otimes_{j=1}^{d} x_j^T \right) A^T A \left(\otimes_{j=1}^{d} x_j \right) - \otimes_{j=1}^{d} x_j^T A^T b.$$

We point out that

$$\mathcal{J}(x_1, \ldots, x_d) = \frac{1}{2} \left| A^{-1}b - \otimes_{j=1}^{d} x_j \right|_{A^T A}^2 - \frac{1}{2} |A^{-1}b|_{A^T A}^2,$$

where $|\cdot|_A$ denotes the norm induced by the inner product over $\mathbb{R}^{\ell(\mathbf{m})}$ defined by $\langle u, v \rangle_{A^T A} := u^T A^T A v$.

Remark 3.1. *If A is a full-rank symmetric matrix, that is $A^T = A$, then the we use the map*

$$\mathcal{J}(x_1, \ldots, x_d) := \frac{1}{2} \left(\otimes_{j=1}^{d} x_j^T \right) A \left(\otimes_{j=1}^{d} x_j \right) - \otimes_{j=1}^{d} x_j^T b,$$

and

$$\mathcal{J}(x_1, \ldots, x_d) = \frac{1}{2} \left| A^{-1}b - \otimes_{j=1}^{d} x_j \right|_A^2 - \frac{1}{2} |A^{-1}b|_A^2.$$

A first approach to compute a minimum, perhaps local, for the map $\mathcal{J}$, is the well-known Alternating Least Squares ALS strategy, based in the Block Coordinated Descent Method given in Algorithm 1.

The next statement gives the conditions for the convergence of Algorithm 1 to a stationary point of the map $\mathcal{J}$.

Theorem 3.2. *Let $b \in \mathbb{R}^{\ell(\mathbf{m})}$ and $A \in \mathbb{R}^{\ell(\mathbf{m}) \times \ell(\mathbf{m})}$, be an invertible matrix. Assume that for each $k \in \{1, 2, \ldots, d\}$ the $\ell(\mathbf{m}) \times m_k$-matrix $Z_k = A \left(id_k \otimes \mathbf{x}_{[k]} \right)$ (respectively, the $m_k \times m_k$-matrix*

$$Z_k = \left(id_k \otimes \mathbf{x}_{[k]} \right)^T A \left(id_k \otimes \mathbf{x}_{[k]} \right)$$

Algorithm 1 A Block Coordinated Descent Algorithm

1: Given $\mathcal{J} : \mathbb{R}^{|\mathbf{m}|} \to \mathbb{R}$
2: Initialize $x_i^{(0)} \in \mathbb{R}^{r_i}$ for $i = 1, 2 \ldots, d$.
3: **for** $n = 1, 2, \ldots,$ **do**
4:      **for** $k = 1, 2, \ldots, d$ **do**
5:          $x_k^{(n)} \in \arg\min_{x_k \in \mathbb{R}^{r_k}} \mathcal{J}(x_1^{(n)}, \ldots, x_{k-1}^{(n)}, x_k, x_{k+1}^{(n-1)}, \ldots, x_d^{(n-1)})$
6:      **end for**
7: **end for**

---

if $A = A^T$) has linearly independent columns for every $(\mathbf{x}_1, \ldots, \mathbf{x}_d) \in \mathbb{R}^{|\mathbf{m}|}$ satisfying

$$\left| A^{-1}b - \otimes_{j=1}^d x_j \right|_Z \leq \left| A^{-1}b - \otimes_{j=1}^d x_j^{(0)} \right|_Z, \tag{18}$$

where $Z = A^T A$ (respectively, $Z = A$ if $A = A^T$). Then every accumulation point $(x_1^*, \ldots, x_d^*)$ of the sequence

$$\left\{ \left(x_1^{(n)}, \ldots, x_d^{(n)} \right) \right\}_{n=0}^\infty,$$

generated by Algorithm 1 satisfies the equation $\nabla \mathcal{J}(\mathbf{x}_1^*, \ldots, \mathbf{x}_d^*) = \mathbf{0}$. Moreover, assume that

$$\{x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d\},$$

are fixed for some $k \in \{1, 2, \ldots, d\}$, then $x_k = (Z_k^T Z_k)^{-1} Z_k^T b$ (respectively, $x_k = Z_k^{-1} b$ if $A = A^T$), is the global minimum of the directional minimization problem

$$\min_{x \in \mathbb{R}^{m_k}} \mathcal{J}(x_1, \ldots, x_{k-1}, x, , x_{k+1}, \ldots, x_d). \tag{19}$$

Proof. We prove the theorem for a general full-rank matrix A, the proof for a symmetric matrix is quite similar. The first part of the theorem follows in a similar way as the first part of Theorem 2 in [1]. To prove the second one, observe that the map from $\mathbb{R}^{m_k}$ to $\mathbb{R}$ given by $x \to \mathcal{J}(x_1, \ldots, x_{k-1}, x, , x_{k+1}, \ldots, x_d)$ is convex. Thus, to conclude the proof we need to show that

$$D_{x_k} \mathcal{J}(x_1, \ldots, x_{k-1}, (Z_k^T Z_k)^{-1} Z_k^T b, x_{k+1}, \ldots, x_d) = 0.$$

To this end, we claim that for all $(y_1, \ldots, y_d) \in \mathbb{R}^{|\mathbf{r}|}$,

$$\frac{d}{dt} \mathcal{J}((x_1 + t y_1, \ldots, x_d + t y_d)) \bigg|_{t=0} = \sum_{k=1}^d y_k^T D_{x_k} \mathcal{J}(\mathbf{x})^T, \tag{20}$$

where $\mathbf{x} = \otimes_{j=1}^d x_j$, is equal to

$$\sum_{k=1}^d y_k^T \left((id_k \otimes \mathbf{x}_{[k]})^T A^T b - (id_k \otimes \mathbf{x}_{[k]})^T A^T A \otimes_{j=1}^d x_j \right). \tag{21}$$

Recall that $D\mathcal{J}(\mathbf{x}) = [D_{x_1} \mathcal{J}(\mathbf{x}) \cdots D_{x_d} \mathcal{J}(\mathbf{x})] \in \mathbb{R}^{1 \times |\mathbf{m}|}$ and $\nabla \mathcal{J}(\mathbf{x}) = D\mathcal{J}(\mathbf{x})^T$. To prove the claim we write

$$\mathcal{J}((x_1 + t y_1, \ldots, x_d + t y_d)) = \frac{1}{2} b^T b - (\mathbf{x} + t\mathbf{y})^T A^T b + \frac{1}{2}(\mathbf{x} + t\mathbf{y})^T A^T A(\mathbf{x} + t\mathbf{y}),$$

where $\mathbf{x} + t\mathbf{y} := \otimes_{j=1}^d (x_j + ty_j)$. Then

$$\frac{d\mathcal{J}}{dt} = -\sum_{k=1}^d \left(y_k \otimes (\mathbf{x}_{[k]} + t\mathbf{y}_{[k]}) \right)^T A^T b + \sum_{k=1}^d \left(y_k \otimes (\mathbf{x}_{[k]} + t\mathbf{y}_{[k]}) \right)^T A^T A (\mathbf{x} + t\mathbf{y}),$$

here $\mathbf{x}_{[k]} + t\mathbf{y}_{[k]} := \bigotimes_{j \neq k} (x_j + ty_j)$. Now, by taking $t = 0$, we have that (20) is equivalent to

$$\sum_{k=1}^d \left(y_k \otimes \mathbf{x}_{[k]} \right)^T A^T b - \sum_{k=1}^d \left(y_k \otimes \mathbf{x}_{[k]} \right)^T A^T A \otimes_{j=1}^d x_j.$$

Since, $\left(y_k \otimes \mathbf{x}_{[k]} \right) = \left(id_k \otimes \mathbf{x}_{[k]} \right) y_k$ for $k = 1, \ldots, d$ the claim follows. Thus

$$D_{x_k} \mathcal{J}(\mathbf{x})^T = \left(id_k \otimes \mathbf{x}_{[k]} \right)^T A^T b - \left(id_k \otimes \mathbf{x}_{[k]} \right)^T A^T A \left(id_k \otimes \mathbf{x}_{[k]} \right) x_k.$$

for $1 \leq k \leq d$, and the second part of the theorem follows. ∎

Remark 3.3. *In practice we compute $id_k \otimes \mathbf{x}_{[k]}$ as*

$$x_1 \otimes \cdots \otimes x_{k-1} \otimes I_{m_k} \otimes x_{k+1} \cdots \otimes x_d$$

where $\otimes$ denotes the Kronecker product and I_{m_k} the identity matrix in $\mathbb{R}^{m_k \times m_k}$.

4. On the implementation of (15) for Laplacian-like matrices

In this section we now assume that $A \in \mathbb{R}^{\ell(\mathbf{m}) \times \ell(\mathbf{m})}$, where $\mathbf{m} \in \mathbb{N}^d$, is a Laplacian-like matrix. Thus, we introduce the following definition.

Definition 4.1. *We say that a full-rank matrix $A \in \mathbb{R}^{\ell(\mathbf{m}) \times \ell(\mathbf{m})}$, is Laplacian-like, if there exist $r \geq 1$ and $A_j^{(i)} \in \mathbb{R}^{m_j \times m_j}$ for $1 \leq j \leq d$ and $1 \leq i \leq r$ such that*

$$A = \sum_{i=1}^r \bigotimes_{j=1}^d A_j^{(i)}.$$

A first property is the following.

Lemma 4.2. *If $A \in \mathbb{R}^{\ell(\mathbf{m}) \times \ell(\mathbf{m})}$ is a Laplacian-like matrix, then $A^T A$ is also Laplacian-like.*

Proof. Since

$$A^T A = \sum_{l=1}^r \sum_{i=1}^r \bigotimes_{j=1}^d (A_j^{(l)})^T A_j^{(i)},$$

the lemma follows. ∎

From Theorem 2.2 to compute an approximate solution of (15) we need to solve iteratively a linear system

$$Z_k^T Z_k x_k = (id_k \otimes \mathbf{x}_{[k]})^T b, \tag{22}$$

where $Z_k = A(id_k \otimes \mathbf{x}_{[k]})$ and $1 \leq k \leq d$. A first approach introduced in Corollary 1 of [1], proposes to compute previously the matrix Z_k as

$$\sum_{i=1}^{r} \left(A_1^{(i)} x_1 \otimes \cdots \otimes A_{k-1}^{(i)} x_{k-1} \otimes I_{m_k} \otimes A_{k+1}^{(i)} x_{k+1} \cdots \otimes A_d^{(i)} x_d \right), \qquad (23)$$

and then solve (22). It is not difficult to see that if $m_i = m$ for $1 \leq i \leq d$ then the number of elementary operations to construct Z_k by using (23) is $\mathcal{O}(m^d)$.

We remark that if A is a Laplacian-like matrix, from Lemma 4.2, we may assume, without loss of generality, that A is a symmetric definite positive matrix (replacing A by $A^T A$). In consequence, the sequence constructed with the PGD algorithm (14)-(17), where in (15) we replace $|b - Av|_2$ by $|A^{-1}b - v|_A$, converges in the $|\cdot|_A$-norm to $A^{-1}b$. Thus we can use Algorithm 2 replacing in 7: $A^T A$ by A and $A^T b$ by b. In this case we need to solve iteratively the linear system

$$(id_k \otimes \mathbf{x}_{[k]})^T A(id_k \otimes \mathbf{x}_{[k]}) x_k = (id_k \otimes \mathbf{x}_{[k]})^T b. \qquad (24)$$

By using the elementary rules of the Kronecker product, the matrix of the left side of (24) can be written as

$$(id_k \otimes \mathbf{x}_{[k]})^T \left(\sum_{i=1}^{r} \bigotimes_{j=1}^{d} A_j^{(i)} \right) (id_k \otimes \mathbf{x}_{[k]}) = \sum_{i=1}^{r} \left(\bigotimes_{j \neq k} x_j^T A_j^{(i)} x_j \right) \otimes A_k^{(i)}.$$

Since $x_j^T A_j^{(i)} x_j \in \mathbb{R}$, this allows to compute directly the left side of (24) as

$$\sum_{i=1}^{r} \alpha_k^{(i)} A_k^{(i)}, \qquad (25)$$

where

$$\alpha_k^{(i)} := \prod_{\substack{j=1 \\ j \neq k}}^{d} x_j^T A_j^{(i)} x_j \text{ for } 1 \leq k \leq d, \ 1 \leq i \leq r, \qquad (26)$$

and then solve the linear system (24). If $m_k = m$ for $1 \leq k \leq d$ this strategy uses $\mathcal{O}(m^2)$ elementary numerical operations. Thus, under this assumptions and for $d \geq 3$, it seems better the use of this second strategy.

Theorem 3.2 jointly all said above allows to propose the following procedure, given in Algorithm 2, to compute a minimum of $\mathcal{J}$ for Laplacian-like matrices. We point out that to check if the minimum is reached, it is better the use of the `distance` function given by $\max_{1 \leq i \leq d} |x_i^{(0)} - x_i^{(1)}|_2$, against the standard one given by $|\mathcal{J}(x_1^{(0)}, \ldots, x_d^{(0)}) - \mathcal{J}(x_1^{(1)}, \ldots, x_d^{(1)})|$, because the number of elementary operations is smaller.

Now, we give some numerical examples of the (14)- (17) algorithm, with $\mathbf{r} = (1, \ldots, 1)$, and performed by using Octave in a laptop with an Intel Atom processor with 1 Gb RAM and running under a Linux Kernel 2.6.38. We have previously fixed $d = r = 3$. Then, given $m \geq 1$, the linear systems are constructed as follows. We generate randomly, for each ω in some finite sample space Ω, a symmetric positive definite matrix $A_k^j(\omega) \in \mathbb{R}^{m \times m}$, for $1 \leq k \leq 3$ and $1 \leq j \leq 3$, and a vector $b(\omega) \in \mathbb{R}^{m^3}$. Then we consider

$$A(\omega) = \sum_{j=1}^{3} (A_1^j \otimes A_2^j \otimes A_3^j)(\omega) \in \mathbb{R}^{m^3 \times m^3}. \qquad (27)$$

Algorithm 2 An Alternated Block Coordinate Algorithm for Laplacian-like matrices
***

1: Given $A = \sum_{i=1}^{r} \bigotimes_{j=1}^{d} A_j^{(i)} \in \mathbb{R}^{\ell(\mathbf{m}) \times \ell(\mathbf{m})}$ and $b \in \mathbb{R}^{\ell(\mathbf{m})}$.

2: Initialize $x_i^{(0)} \in \mathbb{R}^{r_i}$ for $i = 1, 2 \ldots, d$.

3: Introduce $\varepsilon > 0$ and `itermax`, `iter` $= 1$.

4: **while** distance $> \varepsilon$ and `iter` $<$ `itermax` **do**

5:     **for** $k = 1, 2, \ldots, d$ **do**

6:         $x_k^{(1)} = x_k^{(0)}$

7:         **for** $i = 1, 2, \ldots, r$ **do**

8:             $\alpha_k^{(i)} = \left(\prod_{j=1}^{k-1} (x_j^{(0)})^T A_j^{(i)} x_j^{(0)} \right) \left(\prod_{j=k+1}^{d} (x_j^{(1)})^T A_j^{(i)} x_j^{(1)} \right)$

9:         **end for**

10:        $x_k^{(0)}$ solves $\left(\sum_{i=1}^{r} \alpha_k^{(i)} A_k^{(i)} \right) x_k = (id_k \otimes \mathbf{x}_{[k]}^{(0)})^T b$ $\triangleright \mathbf{x}_{[k]}^{(0)} = \otimes_{j \neq k} x_j^{(0)}$.

11:     **end for**

12:     `iter` $=$ `iter` $+ 1$.

13:     distance $= \max_{1 \leq i \leq d} |x_i^{(0)} - x_i^{(1)}|_2$.

14: **end while**
***

We study, in a sample space $\Omega = \{\omega_1, \ldots, \omega_N\}$, the residual behavior of the (14)- (17) steps by means the map $\Phi_N : \mathbb{N} \to \mathbb{R}$ given by

$$\Phi_N(n) := \log \left(\frac{1}{N} \sum_{i=1}^{N} |b(\omega_i) - A(\omega_i) u_{n-1}(\omega_i)|_2 \right),$$

where $u_n(\omega_i)$ solves (15), that is, in this case

$$\min_{(v_1, \ldots, v_d) \in \mathbb{R}^m \times \cdots \times \mathbb{R}^m} |b(\omega_i) - A(\omega_i)(u_{n-1}(\omega_i) + \otimes_{j=1}^{d} v_j)|_2.$$

To show how both strategies can affect the convergence of the PGD, we fix $m = 4$ and consider two samples $N_1 = N_2 = 500$. In both cases we run (14)-(17), by using the Algorithm 2 with `itermax` $= 50$ and $\varepsilon = 2.2204 \times 10^{-16}$. In the first sample, we use (23) to solve (15), in Figure 1 we plot $\Phi_{N_1}(n)$ for $n = 1, \ldots, 100$. On the other hand, for the second sample we consider (25) to solve (15). In this second experiment we also plot $\Phi_{N_2}(n)$ for $n = 1, \ldots, 100$ as we show in Figure 2. In this performance of the PGD algorithm (14)-(17) for Laplacian-like matrices, we obtain better numerical results by using (25) against the use of (23) proposed in [1].

Now, we would to test the dependence of the accuracy with respect to m. To this end we consider $(N, m) \in \{(1, 4), (1, 7), (1, 10), (1, 15)\}$ and compute for each case u_{100} as an approximation of $A^{-1}b$. Also we perform Algorithm 2 with `itermax` $= 50$ and $\varepsilon = 2.2204 \times 10^{-16}$. In Table 4 we show the evolution of the relative error and the CPU time as a function of m. We remark that for $m = 20$, that is $A \in \mathbb{R}^{5832 \times 5832}$, when we try to compute A to use Octave's standard solver, it returns the message: `error:  memory exhausted or requested size too large for range of Octave's index`. However, using (14)-(17) with (25, since we do not need to compute explicitly the matrix A we can solve, for $m = 20$, the linear system in a CPU time of 4070.55 seconds. The residual behavior can be seen in Figure 3.
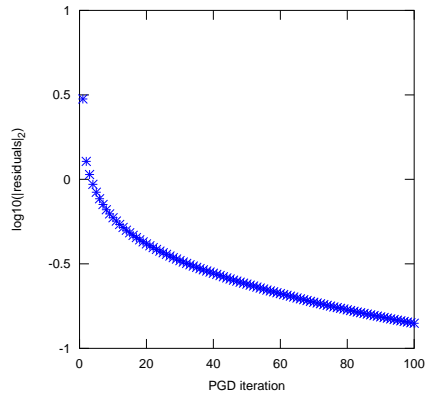
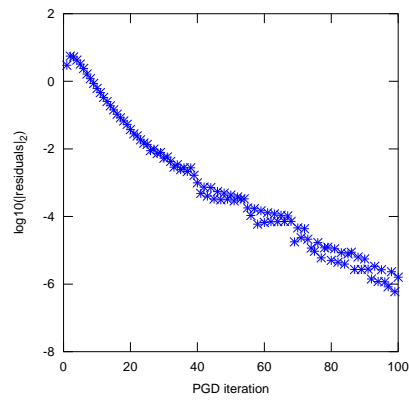Figure 1: The map Φ_{N_1} for $n = 1, \ldots, 100$.

Figure 2: The map Φ_{N_2} for $n = 1, \ldots, 100$.

10

m, m^3	CPU time	Relative error
4,64	12.65	9.53e-11
7,343	17.53	0.0016
10,1000	23.94	0.0544
15,3375	52.42	0.1567

Table 1: The CPU time and the relative error obtained by using 100 PGD-steps to compute an approximate solution $\mathbf{u}_{100} \approx A^{-1}b$, for linear system where A is a symmetric matrix given in separated representation form with $r = 3$ and $d = 3$.
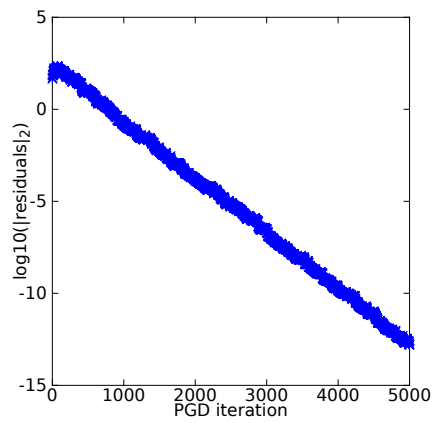

Figure 3: The evolution for the residuals for 5000 steps of PGD by using (25) for a matrix of size 5832×5832. The CPU time was of 4070.55 seconds.

11

[1] A. Ammar, F. Chinesta, and A. Falcó: On the convergence of a greedy rank-one update algorithm for a class of linear systems. *Archives of Computational Methods in Engineering*, 17(4):473–486, 2010.

[2] A. Ammar, B. Mokdad, F. Chinesta and, R. Keunings: A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, *J. Non-Newtonian Fluid Mech.* , **139**, 153-176, 2006.

[3] A. Ammar, B. Mokdad, F. Chinesta and, R. Keunings: A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II: transient simulation using space-time separated representations, *J. Non-Newtonian Fluid Mech.*, **144**, 98-121, 2007.

[4] C. Le Bris, T. Lelièvre, and, Y. Maday: *Constructive Approximation*, 30(3):621–651, 2009.

[5] A. Falcó and, W. Hackbusch: On minimal subspaces in tensor representations. To appear in *Found. Comput. Math.* Available as Preprint 70/2010 at Max Planck Institute for Mathematics in the Sciences, 2010.

[6] A. Falcó and, A. Nouy: A Proper Generalized Decomposition for the solution of elliptic problems in abstract form by using a functional Eckart-Young approach. *J. Math. Anal. Appl.*, 376: 469–480, 2011.

[7] L. Figueroa and, E. Süli: Greedy approximation of high-dimensional Ornstein-Uhlenbeck operators with unbounded drift. Submitted, 2011. Available from arXiv: http://arxiv.org/abs/1103.0726

[8] W. H. Greub: *Linear Algebra.* Graduate Text in Mathematics, Springer-Verlag, fourth edition, 1981.

[9] W. Hackbusch: *Tensor spaces and numerical tensor calculus.* Monograph in preparation.

[10] R. A. DeVore and, V. N. Temlyakov: Some remarks on greedy algorithms. *Adv. Comput. Math.* 5 (2-3), 173187, 1996.

[11] V. de Silva and, L.-H. Lim: Tensor rank and ill-posedness of the best low-rank approximation problem. *SIAM Journal of Matrix Analysis & Appl.*, 30(3):1084–1127, 2008.

[12] C. Eckart and, G. Young: The Approximation Of One Matrix By Another Of Lower Rank. *Psychometrika*, 1(3):211–218, 1936.

[13] A. Lozinski, R. G. Owen, and T. N. Phillips: The Langevin and the Fokker–Plank Equation in Polymer Rheology. In *Numerical Methods for Non-Newtonian Fluids.* Handbook of Numerical Analysis, vol. XVI: 211–303, Elsevier 2011.

[14] R. Meise and D. Vogt: *Introduction to Functional Analysis.* Clarendon Press, Oxford, 1997.

[15] E. Schmidt : Zur Theorie der linearen und nichtlinearen Integralgleichungen I, *Math. Annalen* 63:433–476, 1906.