

TENSOR-BASED METHODS AND PROPER GENERALIZED DECOMPOSITIONS FOR THE NUMERICAL SOLUTION OF HIGH DIMENSIONAL PROBLEMS: ALTERNATIVE DEFINITIONS

G. BONITHON* AND A. NOUY†

Abstract. Tensor-based methods are receiving a growing interest in computational science and engineering for the numerical solution of problems defined in high dimensional tensor spaces. A family of methods called Proper Generalized Decomposition (PGD) methods have been recently proposed. They introduce alternative definitions of tensor approximations, not based on natural best approximation problems, for the approximation to be computable without a priori information on the solution of problems. In this paper, we provide a general presentation of PGD methods in an abstract variational framework and we introduce and compare different definitions of tensor approximations. Convergence results are provided for some classes of variational problems and some variants of PGD. We also present how the PGD can be judiciously coupled with classical iterative methods where it is used as a solver of successive linear problems, thus allowing the use of a wider class of iterative methods compared to other classical tensor-based methods.

Key words. High-dimensional problems, Tensor approximation, Proper Generalized Decomposition, Greedy Algorithms, Model Reduction.

AMS subject classifications. 15A69, 41A63, 65N99, 65D15, 90C06

1. Introduction. Tensor-based methods are receiving a growing interest in computational science and engineering for the numerical solution of problems defined in high dimensional tensor spaces

$$A(u) = b, \quad u \in V_1 \otimes \dots \otimes V_d \quad (1.1)$$

Typical examples include partial differential equations arising in stochastic calculus (e.g. Fokker-Planck equations), stochastic parametric partial differential equations in uncertainty quantification with functional approaches, and many mechanical or physical models involving extra parameters (for parametric analyses, optimization or inverse problems). Classical numerical methods consists in searching an approximation of the solution in approximation spaces which are tensor products of pre-defined approximation spaces. These approximation methods suffer from the so called *curse of dimensionality* associated with the dramatic increase of the dimension of the resulting approximation spaces when d increases. The idea of tensor-based methods is to construct an approximation of the solution under the form

$$u_m = \sum_{i=1}^m w_i^1 \otimes \dots \otimes w_i^d, \quad w_i^k \in V_k \quad (1.2)$$

which is called a rank- m separated representation or rank- m canonical tensor decomposition. The interest of representation (1.2) is that its dimensionality only grows linearly with the dimension d . The construction of a separated representation u_m of a given element of a tensor space has been extensively studied over the past years and different definitions and associated algorithms have been proposed [25]. These

*LUNAM Université, GeM, UMR CNRS 6183, Ecole Centrale Nantes, Université de Nantes, 1 rue de la Noë, BP 92101, 44321 Nantes, FRANCE (gael.bonithon@ec-nantes.fr).

†LUNAM Université, GeM, UMR CNRS 6183, Ecole Centrale Nantes, Université de Nantes, 1 rue de la Noë, BP 92101, 44321 Nantes, FRANCE (anthony.nouy@ec-nantes.fr).

definitions can be seen as multidimensional versions of the Singular Value Decomposition (SVD). The question of finding an optimal decomposition of a given rank m is not a trivial question. Indeed, in general, an optimal representation u_m can not be simply defined as an optimization problem on the set of rank- m tensors \mathcal{R}_m (elements of the form (1.2)) since it may lead to an ill-posed problem (for $d \geq 3$) [10]. Various alternative definitions of separated representations have been proposed, leading to well posed tensor approximation problems. They are based on progressive constructions of separated representations, consisting in successive well-posed approximation problems in small tensor subsets (*e.g.* rank-one tensors) [9], or on the introduction of suitable tensor subsets such as rank- m tensors with orthogonality or boundedness constraints [7, 37], Tucker tensors [17], Hierarchical tensors [20, 16], or tensor-train tensors [38].

A first family of numerical methods based on the construction of tensor approximations have been recently proposed for the solution of high-dimensional partial differential equations [19, 3, 24, 29]. They rely on the use of classical tensor approximations inside classical iterative methods for the solution of (1.1) (*e.g.* Gradient or Krylov-type iterative methods, Newton methods for nonlinear equations, power iterations for eigenproblems...). These strategies are currently receiving a growing interest and their analysis is closely related to the analysis of the impact of perturbations on the behavior of iterative methods [18].

Another family of methods, called Proper Generalized Decomposition (PGD) methods, have been introduced for the a priori construction of approximations of the solution of problems defined in tensor spaces [27, 2, 30, 28, 8, 33, 32, 11]. PGD methods introduce alternative definitions of tensor approximations, not based on natural best approximation problems, for the approximation to be computable without a priori information on the solution u . Formally, PGD methods can be summarized as the construction of a sequence $\{u_m\}$ of approximations $u_m \in \mathcal{S}_m$, where \mathcal{S}_m is a given tensor subset (subset of the algebraic tensor space). Let us note that this definition is quite general and does not justify the terminology ‘‘Proper Generalized Decomposition’’. However, the particular nature of the tensor subsets \mathcal{S}_m (typically $\mathcal{S}_m = \mathcal{R}_m$) and some of the proposed definitions of sequences of approximations make this construction a generalization of Proper Orthogonal Decomposition (or Singular Value Decomposition) [13]. For this reason, PGD was initially called Generalized Spectral Decomposition [30, 31, 34] in the context of stochastic partial differential equations, where it can be seen as a generalization of the Karhunen-Loève decomposition.

In recent years, different versions of PGDs have been proposed in different contexts and some convergence results have been obtained for some versions and for some particular problems [5, 1, 6, 15, 13, 14]. In this paper, we provide a general presentation of PGD methods in an abstract variational framework and we introduce and compare alternative definitions of a priori tensor approximations. Some theoretical results are provided for some classes of variational problems. We also present how the PGD can be coupled with classical iterative methods where it is used as a solver of successive linear problems, thus allowing the use of a wider class of iterative methods compared to classical tensor-based methods mentioned above.

The different versions of PGD differ by the choice of tensor subsets \mathcal{S}_m and the definition of an approximation $u_m \in \mathcal{S}_m$. We first distinguish direct constructions from progressive constructions. Direct constructions consist in defining $u_m \in \mathcal{S}_m$ independently of the other $u_k \in \mathcal{S}_k$, $k \neq m$. These constructions may become computationally expansive when increasing approximation sets \mathcal{S}_m , so that constructive

approximations based on progressive decompositions are generally preferred. Progressive constructions consist in defining $\mathcal{S}_m = \mathcal{S}_{m-1} + \mathcal{S}$, with \mathcal{S} a small tensor subset, and $u_m = u_{m-1} + w_m$, with $w_m \in \mathcal{S}$ a “good correction” of u_{m-1} .¹ Progressive definitions of PGDs can be considered as Greedy algorithms [36] for constructing tensor decompositions (with \mathcal{S} as a *dictionary* of functions). The difficulty in the definition of progressive PGDs is that a correction $w_m \in \mathcal{S}$ must be defined in such a way that it is computable without a priori information on the solution. The correction w_m can be defined by a minimization problem if the variational problem is equivalent to the minimization of a certain functional (e.g. convex optimization problems). In particular, for general variational problems, the correction can be obtained by a minimization of the residual of (1.1). This minimal residual formulation is robust in the sense that the sequence u_m monotonically converges with respect to the residual norm. However, the convergence properties are closely related to the choice of residual norms. Therefore, the main difficulty resides in the choice of good residual norms that can be easily computed within the present tensor format. The correction w_m can also be defined by Galerkin orthogonality conditions. The progressive PGD based on Bubnov-Galerkin orthogonality conditions was historically the first version of PGD and it has proved efficient in many applications. However, it is not robust in the sense that monotone convergence is not guaranteed for general non-symmetric problems and the sequence u_m may diverge in some situations. Another possible definition of a good correction w_m can be introduced, based on Petrov-Galerkin orthogonality conditions.² This definition can significantly improve convergence properties of progressive PGD with respect to some chosen norms.

The above progressive definitions are here presented in a general and unified framework. Convergence results are provided for some classes of variational problems and some versions of progressive PGDs. A particular class of problems involving so-called \mathcal{S} -tangent operators is introduced. For these problems, a monotone decrease of some residual norm is proved for PGD based on Bubnov-Galerkin orthogonality conditions, even for non-symmetric problems. A possible improvement of purely progressive decompositions consists in introducing some updating steps by taking part at step m of the previously generated information, *i.e.* of $\{u_k\}_{1 \leq k \leq m}$. This allows capturing a better approximation of an optimal decomposition which could be obtained by a direct construction. For many applications, it allows recovering good convergence properties of separated representations. In this paper, different strategies of updates are introduced and compared.

The paper is organized as follows. In section 2, we introduce general notions about tensors and their approximations. In section 3, we present an abstract setting for variational problems formulated in high dimensional tensor spaces. We also present classical iterative methods and present possible strategies based on a posteriori tensor approximation. In section 4, we introduce a general presentation of PGD methods for the a priori construction of a tensor approximation of the solution of high-dimensional variational problems. The particular case of convex optimization problems serves as a guideline in this general presentation. We also discuss the possible use of PGD

¹A typical choice for \mathcal{S} , which is the most widely used, is the set of rank-one tensors \mathcal{R}_1 , thus yielding the progressive construction of a rank- m decomposition $u_m \in \mathcal{R}_m$.

²This idea has been recently introduced in [32] in the context of time-dependent partial differential equations, where it was interpreted as a Petrov-Galerkin model reduction technique, where test and trial reduced basis functions are related by an adjoint problem.

methods inside classical iterative solvers (e.g. for the solution of nonlinear problems). In section 5, we propose alternative definitions of progressive PGDs for more general variational problems. We also discuss an interesting property of some differential operators, called \mathcal{S} -tangency, which yields good convergence properties of progressive PGD for a large class of applications. Finally, in section 6, different variants of PGDs are illustrated and compared on numerical examples.

2. Tensor product spaces and tensors representations.

2.1. Tensor Hilbert spaces. We consider Hilbert spaces V_k , $1 \leq k \leq d$, equipped with inner products $\langle \cdot, \cdot \rangle_k$ and associated norms $\| \cdot \|_k$. We define the set of elementary tensors (or rank-one tensors)

$$\mathcal{R}_1 = \{w = w^1 \otimes \dots \otimes w^d : w^k \in V_k \text{ for } 1 \leq k \leq d\}$$

The algebraic tensor space is defined as the span of elementary tensors

$${}_a \otimes_{k=1}^d V_k = \text{span}\{v : v \in \mathcal{R}_1\}$$

For each element $v \in {}_a \otimes_{k=1}^d V_k$, there exists $m \in \mathbb{N}$ such that $v = \sum_{i=1}^m v_i$ with some $v_i \in \mathcal{R}_1$. The algebraic tensor space is now equipped with an inner product $\langle \cdot, \cdot \rangle$ and associated norm $\| \cdot \|$. The resulting normed vector space ${}_a \otimes_{k=1}^d V_k$ is a pre-Hilbert space. A Hilbert space $V_{\| \cdot \|}$ equipped with inner product $\langle \cdot, \cdot \rangle$ and associated norm $\| \cdot \|$ is obtained by the completion of the algebraic tensor space

$$V_{\| \cdot \|} = \overline{{}_a \otimes_{k=1}^d V_k}^{\| \cdot \|}.$$

If there is no ambiguity on the choice of norm, we will simply denote $V = V_{\| \cdot \|}$. Note that in the finite dimensional case, since all norms are equivalent, the resulting topological space $V_{\| \cdot \|}$ is independent of the choice of norm.

Canonical norm and inner product. We now introduce a particular but natural inner product. For elementary tensors $w = \otimes_{k=1}^d w^k \in \mathcal{R}_1$ and $v = \otimes_{k=1}^d v^k \in \mathcal{R}_1$, we let $\langle w, v \rangle_V = \langle \otimes_{k=1}^d w^k, \otimes_{k=1}^d v^k \rangle_V = \prod_{k=1}^d \langle w^k, v^k \rangle_k$. This definition is then extended by linearity on the whole algebraic tensor product space. In the sequel, $\langle \cdot, \cdot \rangle_V$ is called the canonical inner product. The norm associated with $\langle \cdot, \cdot \rangle_V$ is denoted $\| \cdot \|_V$ and will be called the canonical norm. For an elementary tensor $w = \otimes_{k=1}^d w^k \in \mathcal{R}_1$, this norm verifies $\| \otimes_{k=1}^d w^k \|_V = \prod_{k=1}^d \|w^k\|_k$, which is the property of a crossnorm.

2.2. Tensor subsets.

2.2.1. Rank- m tensors. The set of tensors with canonical rank m , called rank- m tensors for brevity, is defined by

$$\mathcal{R}_m = \{v = \sum_{i=1}^m w_i : w_i \in \mathcal{R}_1 \text{ for } 1 \leq i \leq m\} \quad (2.1)$$

Note that $\bigcup_{m \geq 1} \mathcal{R}_m = {}_a \otimes_{k=1}^d V_k$.

2.2.2. Tucker tensors. For $\mathbf{r} = (r_1, \dots, r_d) \in \mathbb{N}^d$, we define the Tucker tensors set $\mathcal{T}_{\mathbf{r}}$ as follows:

$$\begin{aligned} \mathcal{T}_{\mathbf{r}} &= \left\{ v \in V \text{ such that there exist linear subspaces } U_k \subset V_k \right. \\ &\quad \left. \text{with } \dim(U_k) = r_k \text{ such that } v \in {}_a \otimes_{k=1}^d U_k \right\} \\ &= \left\{ v = \sum_{i \in \mathcal{I}_{\mathbf{r}}} \alpha_i w_i \in V ; \alpha_i \in \mathbb{R}, w_i = \otimes_{k=1}^d w_{i_k}^k \in \mathcal{R}_1, \right. \\ &\quad \left. \text{with } \langle w_i^k, w_j^k \rangle_k = \delta_{ij} \text{ for all } k \text{ and } 1 \leq i, j \leq r_k \right\} \end{aligned} \quad (2.2)$$

where $\mathcal{I}_{\mathbf{r}} = \{\mathbf{i} = (i_1, \dots, i_d) \in \mathbb{N}^d; 1 \leq i_k \leq r_k \text{ for all } k \in \{1, \dots, d\}\}$ is a set of multi-indices. In the Tucker representation (2.2), $\boldsymbol{\alpha} = (\alpha_{\mathbf{i}})_{\mathbf{i} \in \mathcal{I}_{\mathbf{r}}} \in \mathbb{R}^{r_1 \times \dots \times r_d}$ is called the core tensor³. Note that $\mathcal{T}_{\mathbf{r}} \subset \mathcal{R}_{\mathbf{r}^*}$ with $\mathbf{r}^* = r_1 \dots r_d$. Let us also note that $\mathcal{T}_{(1, \dots, 1)} = \mathcal{R}_1$ and, for $d = 2$, $\mathcal{T}_{(m, m)} = \mathcal{R}_m$ for all $m \geq 1$. However, for $m \geq 2$ and for $d \geq 3$, we have $\mathcal{R}_m \subsetneq \mathcal{T}_{(m, \dots, m)}$ with strict inclusion.

2.2.3. Other tensor formats. Other tensors formats have been recently introduced, such as hierarchical tensor format [16] or tensor train format [35]. These formats can be seen as subsets of the Tucker tensors set with a particular structure of the core tensor. These particular formats allow circumventing the curse of dimensionality in the representation of the core tensor (dimensionality grows only linearly with d). Other tensors sets are also available, such as rank- m tensors with orthogonality constraints (see [37] for some properties of these tensors sets).

2.3. Best approximation in tensor subsets. Let $u \in V_{\|\cdot\|}$ and let $\mathcal{S} \subset V$ denote a tensor subset. The best approximation of u in \mathcal{S} in the sense of the $\|\cdot\|$ can be naturally defined by the problem

$$\inf_{v \in \mathcal{S}} \|u - v\| \tag{2.3}$$

This problem is a minimization problem of a convex and continuous (and hence weakly lower semi-continuous) and coercive functional on the subset \mathcal{S} . The existence of a minimizer is then conditioned by the properties of \mathcal{S} . In particular, we have that the best approximation problem (2.3) is well-posed (*i.e.* admits at least one solution) if the set \mathcal{S} is weakly closed, or simply closed in the finite dimensional framework.

Rank-1 approximation. In the finite dimensional case, the set \mathcal{R}_1 is a closed set. In the infinite dimensional case, the set \mathcal{R}_1 is weakly closed in $V_{\|\cdot\|}$ if the norm $\|\cdot\|$ is stronger than a certain crossnorm. In [12], it can be found other conditions on the norm $\|\cdot\|$ that ensure that \mathcal{R}_1 is weakly closed. In particular, it can be proved that \mathcal{R}_1 is weakly closed in Sobolev tensor spaces [14].

Rank- m approximations. It is well known that in the case when $d \geq 3$ and $m \geq 2$, the set \mathcal{R}_m is not weakly closed (nor even closed), so that the best approximation problem (2.3) is ill-posed for $\mathcal{S} = \mathcal{R}_m$ (see *e.g.* [10]).

REMARK 2.1. *It is proved in [37] that if the norm is associated with the canonical inner product, the best approximation problem (2.3) is well-posed when choosing for \mathcal{S} subsets of \mathcal{R}_m with orthogonality constraints between rank-one elements.*

Tucker approximation. In the finite dimensional framework, the Tucker set $\mathcal{T}_{\mathbf{r}}$ is a closed set. In the infinite dimensional case, it can be proved that $\mathcal{T}_{\mathbf{r}}$ is weakly closed in $V_{\|\cdot\|}$ under quite technical assumptions on the norm $\|\cdot\|$ (see [12]). In particular, it is true if the norm $\|\cdot\|$ is not weaker than the injective norm, which is a property verified by the canonical norm in Hilbert tensor spaces. For the case of Sobolev spaces, it can also be proved that the Tucker set is weakly closed by using other arguments (using properties of intersections of tensor spaces) [12].

2.4. Progressive definition of a best approximation. A way of constructing tensor approximations consists in defining a sequence of approximations $\{u_m\}_{m \geq 1}$ such that $u_m = u_{m-1} + w_m$, with $u_0 = 0$ and with $w_m \in \mathcal{S}$ defined by a best

³Note that the number of components $r_1 \dots r_d$ in the core tensor grows exponentially with d (except if $\#\{k \in \{1, \dots, d\} : r_k \geq 2\}$ remains bounded as $d \rightarrow \infty$), which makes Tucker format intractable for high-dimensional applications.

approximation problem on a suitable set \mathcal{S} :

$$\|u - u_{m-1} - w_m\| = \min_{v \in \mathcal{S}} \|u - u_{m-1} - v\| = \min_{v \in u_{m-1} + \mathcal{S}} \|u - v\| \quad (2.4)$$

We then obtain a progressive construction of a series $u_m = \sum_{i=1}^m w_i$ which provides a better and better approximation of u as m increases. The well-posedness of successive best approximation problems (2.4) still depends on the properties of the set \mathcal{S} . We have the following convergence result, which is a direct extension of the proof in [13], and which can be seen as a particular case of a more general result on greedy algorithms [36].

THEOREM 2.2. *Under the following conditions on the subset \mathcal{S}*

- \mathcal{S} is weakly closed in $V_{\|\cdot\|}$,
- $\text{span}\{v : v \in \mathcal{S}\}$ is dense in $V_{\|\cdot\|}$,
- for each $v \in \mathcal{S}$, $\lambda v \in \mathcal{S}$ for all $\lambda \in \mathbb{R}$,

we have the convergence of the sequence $\{u_m\}_{m \geq 1}$ defined by (2.4) towards u

$$\lim_{m \rightarrow \infty} \|u - u_m\| = 0$$

We now give some possible choices for \mathcal{S} .

Progressive rank- m approximation. The basic choice for \mathcal{S} consists in the set of elementary tensors \mathcal{R}_1 . We then obtain a progressive construction of rank- m approximations $u_m \in \mathcal{R}_1 + \dots + \mathcal{R}_1 = \mathcal{R}_m$. In [13], it is given an interpretation of the obtained sequence u_m as a generalized multidimensional singular value decomposition, even for the case where $\|\cdot\|$ is not the canonical norm (nor even a crossnorm).

Other progressive constructions. As discussed in section 2.3, other choices are also possible for \mathcal{S} , such as the Tucker set \mathcal{T}_r for some $r \in \mathbb{N}^d$, or the set of tensors in \mathcal{R}_m with orthogonality constraints. The interest of this progressive construction is to formulate successive optimization problems in spaces with moderate dimension (low m or low r), thus avoiding the direct costly computation of a high rank representation. Note that for $\mathcal{S} = \mathcal{T}_r$, we have $u_m \in \mathcal{T}_r + \dots + \mathcal{T}_r \subset \mathcal{T}_{r+\dots+r} = \mathcal{T}_{mr}$.

3. Multidimensional problems and tensor-based solution methods.

3.1. Variational problems in tensor product spaces. Let $V = V_{\|\cdot\|}$ a tensor Hilbert space defined as the tensor product of Hilbert spaces V_k , $1 \leq k \leq d$, and endowed with a norm $\|\cdot\|$ and associated inner product $\langle \cdot, \cdot \rangle$.

3.1.1. A general problem. We consider an abstract formulation of a problem

$$u \in V, \quad \mathcal{A}(u, v) = \mathcal{L}(v) \quad \forall v \in V \quad (3.1)$$

where \mathcal{A} is a bilinear or eventually semilinear form on V , and \mathcal{L} is a linear form on V . We introduce the operator $A : V \rightarrow V$ and the element $b \in V$ associated with \mathcal{A} and \mathcal{L} respectively, defined by

$$\mathcal{A}(u, v) = \langle A(u), v \rangle, \quad \mathcal{L}(v) = \langle b, v \rangle,$$

for all $u, v \in V$.

3.1.2. A particular class of convex optimization problems. In the paper, we will sometimes refer to a particular class of problems associated with an optimization problem. For these problems, we consider that $A(u) - b$ is the gradient of a convex functional $J : V \rightarrow \mathbb{R}$, that means

$$\langle A(u) - b, v \rangle = \langle J'(u), v \rangle$$

for all $u, v \in V$. Equation (3.1) then appears as the Euler equation associated with the minimization problem

$$J(u) = \min_{v \in V} J(v) \quad (3.2)$$

3.2. Solution methods using a posteriori tensor product approximations.

3.2.1. Principle. Tensor-based methods can be used in association with traditional iterative methods for solving problem (3.1). Classical iterative solvers (Newton iterations, conjugate gradient, ...) can be summarized into the construction of a sequence $\{u_n\}_{n \in \mathbb{N}}$ iteratively defined by

$$u_{n+1} = B(u_n), \quad (3.3)$$

with u_0 given and the mapping B such that for $\|u_0 - u\| \leq \delta$, we have $\lim_{n \rightarrow \infty} \|u - u_n\| = 0$. The idea is then to introduce tensor approximations at each iteration of this algorithm, which is equivalent to the use of an approximate iteration mapping B_ϵ . It thus results in the construction of an approximate sequence $\{v_n\}_{n \in \mathbb{N}}$ defined by

$$v_{n+1} = B_\epsilon(v_n), \quad (3.4)$$

with v_0 an approximation of u_0 .

3.2.2. Some requirements and properties. In order for this strategy to make sense, some requirements must be fulfilled:

- Application of mapping B_ϵ must preserve tensor formats and must prevent from a dramatic increase of tensor ranks.
- Algorithm must be stable with respect to perturbations. The approximate sequence should yield an error which depends continuously on the error ϵ on the iteration mapping.

The first requirement limits the choice of algorithms. In practice, simple algorithms have to be used⁴. The following theorem gives a typical situation where the second requirement is fulfilled. For further results on the effect of approximations on iterative solvers, see [18].

THEOREM 3.1. *Assume that B is contractive on a δ -neighborhood of u :*

$$\|B(w) - B(v)\| \leq \rho \|w - v\| \quad \forall v, w \in V_\delta(u) = \{v \in V : \|v - u\| < \delta\}$$

with $\rho < 1$ the contractivity constant. Assume that B_ϵ is such that

$$\|B_\epsilon(v) - B(v)\| \leq \epsilon \quad \forall v \in V_\delta(u)$$

Then, if $v_0 \in V_\delta(u)$ and if $\epsilon < \delta(1 - \rho)$, we have

$$\limsup_{n \rightarrow \infty} \|u - v_n\| \leq \frac{\epsilon}{1 - \rho}$$

Proof. We have $v_0 \in V_\delta(u)$. Suppose that $v_n \in V_\delta(u)$. Then,

$$\begin{aligned} \|v_{n+1} - u\| &= \|B_\epsilon(v_n) - B(u)\| \leq \|B_\epsilon(v_n) - B(v_n)\| + \|B(v_n) - B(u)\| \\ &\leq \epsilon + \rho \|v_n - u\| \end{aligned}$$

⁴Note that regarding the first requirement, the use of the PGD method within iterative solvers allows to consider a wider class of iterative methods, as explained in section 4.5.

Since $\epsilon < \delta(1-\rho)$, we have $\|v_{n+1}-u\| < \delta$, which implies $v_{n+1} \in V_\delta(u)$. By induction, we then have $v_n \in V_\delta(u)$ for all $n \geq 0$. Now,

$$\begin{aligned} \|v_n - u\| &\leq \epsilon + \rho\|v_{n-1} - u\| \leq \epsilon \sum_{k=0}^{n-1} \rho^k + \rho^n \|v_0 - u\| \\ &= \frac{\epsilon(1-\rho^n)}{1-\rho} + \rho^n \|v_0 - u\| \leq \frac{\epsilon}{1-\rho} + \rho^n \|v_0 - u\| \end{aligned}$$

and therefore

$$\sup_{n \geq N} \|v_n - u\| \leq \frac{\epsilon}{1-\rho} + \rho^N \|v_0 - u\| \rightarrow \frac{\epsilon}{1-\rho} \text{ as } N \rightarrow \infty$$

In other words, for all $\gamma > 0$, there exists N such that for all $n \geq N$, $v_n \in V_{\gamma'}(u)$ with $\gamma' = \frac{\epsilon}{1-\rho} + \gamma$. \square

3.2.3. An example involving a monotone operator. As an example, let us consider the case of a strongly monotone and Lipschitz continuous mapping A , and let us consider the simple algorithm (3.3) with iteration mapping

$$B(u) = u - \alpha(A(u) - b),$$

with α sufficiently small⁵ to ensure the contractivity of B . We then introduce an approximation map $T_\epsilon : V \rightarrow \otimes_{k=1}^d V_k$ which associates to any $v \in V$ an approximation $T_\epsilon(v)$ in a suitable subset of the algebraic tensor space. Mapping T_ϵ is supposed to verify

$$\|T_\epsilon(v) - v\| \leq \gamma(\epsilon) \quad \forall v \in V_\delta(u)$$

A basic approximation would consist in using

$$B_\epsilon(v) = T_\epsilon(B(v))$$

For this choice, theorem 3.1 applies for $\gamma(\epsilon) = \epsilon$. In practice, the introduction of further approximations may be required in order to preserve low rank decompositions and therefore to preserve computational efficiency. For example, one could consider the following iteration mapping $B_\epsilon(v) = T_\epsilon(v - T_\epsilon(\alpha(A_\epsilon(v) - b)))$, with A_ϵ a low rank approximation of A which preserves tensor format. In this case, further assumptions are required on A_ϵ and $\gamma(\epsilon)$ in order for theorem 3.1 to be still valid. In conclusion, this type of approximation strategy requires a good analysis of iterative algorithms and a careful approximation of the different steps of these algorithms in order to preserve convergence properties of the approximate sequence.

4. Solution methods using a priori tensor product approximations: Proper Generalized Decompositions.

⁵ $\|B(v) - B(w)\|^2 = \|v - w - \alpha(A(v) - A(w))\|^2 = \|v - w\|^2 - 2\alpha\langle A(v) - A(w), v - w \rangle + \alpha^2\|A(v) - A(w)\|^2 \leq \|v - w\|^2(1 - 2\gamma_A\alpha + \alpha^2C_A^2)$, with γ_A the strong monotonicity constant of A and C_A the continuity constant of A . Denoting $\rho^2 = (1 - 2\gamma_A\alpha + \alpha^2C_A^2)$, then for $\alpha < \frac{2\gamma_A}{C_A^2}$, we have $\rho < 1$ and therefore a contractive mapping B .

4.1. Principle. Proper Generalized Decomposition (PGD) methods introduce alternative definitions of tensor approximations, not based on natural best approximation problems, for the approximation to be computable without a priori information on the solution u . PGD consists in constructing a sequence $\{u_m\}$ of approximations $u_m \in \mathcal{S}_m$, where \mathcal{S}_m is a given tensor subset (subset of the algebraic tensor space). Let us note that this definition is quite general and does not justify the terminology Proper Generalized Decomposition. However, the particular nature of the subsets \mathcal{S}_m and some of the proposed definitions of sequences of approximations make this construction a generalization of singular value decomposition (or Proper Orthogonal Decomposition). This will be clarified later. Different versions of Proper Generalized Decompositions can be defined according to

- (i) the choice of tensor subsets \mathcal{S}_m , and
- (ii) the definition of an approximation $u_m \in \mathcal{S}_m$.

4.2. Choice of tensor subsets. The sequence of sets \mathcal{S}_m will be classically chosen such that:

- (c1) $\mathcal{S}_m \subset \mathcal{S}_{m+1}$,
- (c2) for all $v \in V$, there exists a sequence $\{v_m \in \mathcal{S}_m\}_{m \geq 1}$ that strongly converges to v .

Condition (c1) is necessary for u_{m+1} being potentially better than u_m (if u_m is defined using an optimality criterium). Condition (c2) is a necessary condition for having the convergence of the sequence u_m in all cases. These two conditions are fulfilled with the following choices:

- $\mathcal{S}_m = \mathcal{R}_{m-1} + \mathcal{R}_1 = \mathcal{R}_m$,
- $\mathcal{S}_m = \mathcal{S}_{m-1} + \mathcal{T}_r$,
- $\mathcal{S}_m = \mathcal{T}_{(m, \dots, m)}$.

4.3. Definitions of tensors approximations. Different definitions of an element $u_m \in \mathcal{S}_m$ are possible. We can distinguish direct or progressive definitions. A direct construction, detailed in section 4.3.1, consists in defining $u_m \in \mathcal{S}_m$ independently of the other $u_k \in \mathcal{S}_k$, $k \neq m$. A progressive construction, detailed in section 4.3.2, consists in defining u_m as a correction of u_{m-1} . A progressive construction with updates consists in defining u_m by taking part of previously generated information, *i.e.* knowing $\{u_k\}_{1 \leq k \leq m-1}$. It will be detailed in sections 4.3.3 and 4.3.4. In this section, we present possible definitions for the particular class of problems associated with optimization problem (3.2). For more general problems, the definition of a “good” sequence u_m is less trivial and will be discussed in section 5.

4.3.1. Direct constructions. A direct PGD can formally be seen as the construction of a sequence of approximations $\{u_m\}_{m \geq 1}$, with u_m such that

$$u_m \in \Pi_{\mathcal{S}_m}(u) \subset \mathcal{S}_m$$

where $\{\mathcal{S}_m\}_{m \geq 1}$ is a sequence of tensor subsets satisfying conditions (c1) and (c2), and where $\Pi_{\mathcal{S}_m}(u)$ is a set of “good candidates” in \mathcal{S}_m for the approximation of u .

The particular case of optimization problems. For the case of optimization problem (3.2), the direct PGD can be naturally defined by choosing $u_m \in \mathcal{S}_m$ such that

$$J(u_m) \leq \inf_{v \in \mathcal{S}_m} J(v) + \epsilon_m \tag{4.1}$$

with $\{\epsilon_m\}_{m \geq 1}$ a bounded positive sequence. This sequence may reflect an error in the solution of the optimization problem or may reflect the fact that the optimization

problem of J on \mathcal{S}_m does not admit a solution. Note that if \mathcal{S}_m is weakly closed, there exists a minimizer in \mathcal{S}_m and therefore, $\epsilon_m = 0$ makes sense. The candidates set $\Pi_{\mathcal{S}_m}(u)$ is the set of elements satisfying (4.1), *i.e.*

$$\Pi_{\mathcal{S}_m}(u) = \{v_m \in \mathcal{S}_m : J(v_m) \leq \inf_{v \in \mathcal{S}_m} J(v) + \epsilon_m\}$$

We have the following convergence result.

THEOREM 4.1. *Assume $J : V \rightarrow \mathbb{R}$ is strongly convex, coercive and Fréchet differentiable, with Fréchet differential Lipschitz continuous on bounded sets. Further assume that for any sequence u_m that weakly converges to $v \in V$, we have $(J'(u_m), u_m) \rightarrow (J'(v), v)$ and $J'(u_m) \rightarrow J'(v)$ (weak convergence in V). Then a sequence $\{u_m\}_{m \geq 1}$ with $u_m \in \mathcal{S}_m$ satisfying (4.1) verifies*

$$\limsup_{m \rightarrow \infty} \|u - u_m\|^2 \leq C\epsilon^*$$

with $\epsilon^* = \liminf_{m \rightarrow \infty} \epsilon_m$ and C a constant depending on the properties of J . In particular, if $\epsilon^* = 0$, then u_m strongly converges towards the unique minimizer u of J in V .

Proof. Since $u_1 \in \mathcal{S}_1 \subset \dots \subset \mathcal{S}_m$, we have $J(u_m) \leq J(u_1) + \epsilon_m$. Therefore $J(u_m)$ is bounded since $J(u_1)$ is finite and ϵ_m is a convergent and therefore bounded sequence. Then, we can extract a subsequence that weakly converges to an element $u^* \in V$, still denoted u_m for simplicity. J being continuous and convex, it is weakly lower semi-continuous. Therefore, $J(u^*) \leq \liminf_{m \rightarrow \infty} J(u_m)$. Introducing a sequence $\{v_m \in \mathcal{S}_m\}_{m \geq 1}$ that converges strongly to u (which is possible thanks to condition (c2) on \mathcal{S}_m), we have $J(v_m) \rightarrow J(u)$ (by continuity of J) and $J(u_m) \leq J(v_m) + \epsilon_m$. Therefore

$$J(u^*) \leq \liminf_{m \rightarrow \infty} J(u_m) \leq \liminf_{m \rightarrow \infty} (J(v_m) + \epsilon_m) = J(u) + \epsilon^*$$

Strong convexity of J implies

$$\alpha \|u - u_m\|^2 \leq \langle J'(u_m) - J'(u), u_m - u \rangle = \langle J'(u_m), u_m \rangle - \langle J'(u_m), u \rangle$$

where we have used $J'(u) = 0$. Then, we have that

$$\begin{aligned} \limsup_{m \rightarrow \infty} \alpha \|u - u_m\|^2 &\leq \langle J'(u^*), u^* - u \rangle = \langle J'(u^*) - J'(u), u^* - u \rangle \\ &\leq \|J'(u^*) - J'(u)\| \|u^* - u\| \end{aligned}$$

Again using the strong convexity of J , we have

$$\frac{\alpha}{2} \|u^* - u\|^2 \leq \langle J'(u), u - u^* \rangle + J(u^*) - J(u) \leq \epsilon^*$$

Denoting by M the Lipschitz continuity constant of J on the bounded set $\{v \in V; \|v\| \leq \max\{\|u\|, \|u^*\|\}\}$, we obtain

$$\limsup_{m \rightarrow \infty} \|u - u_m\|^2 \leq \frac{M}{\alpha} \|u^* - u\|^2 \leq \frac{2M\epsilon^*}{\alpha^2}$$

Finally, if $\epsilon^* = 0$, we obtain $\lim_{m \rightarrow \infty} \|u - u_m\| = 0$. \square

4.3.2. Progressive constructions. For progressive constructions, we introduce a (small) tensor subset \mathcal{S} (typically the set of rank-one tensors \mathcal{R}_1) and we define $\mathcal{S}_m = \mathcal{S}_{m-1} + \mathcal{S}$. We then define the sequence u_m in a greedy fashion by

$$u_m = u_{m-1} + w_m$$

where $w_m \in \mathcal{S}$ is a correction of $u_{m-1} \in \mathcal{S}_{m-1}$ such that

$$w_m \in \Pi_{\mathcal{S}}(u - u_{m-1}) \subset \mathcal{S}$$

where $\Pi_{\mathcal{S}}(v)$ denotes a set of “good candidates” for the approximation of v .

The particular case of optimization problems. For optimization problem (3.2), a natural definition is

$$J(u_{m-1} + w_m) = \min_{w \in \mathcal{S}} J(u_{m-1} + w) \quad (4.2)$$

which corresponds to the following definition of the set of “good candidates”:

$$\Pi_{\mathcal{S}}(u - u_{m-1}) = \arg \min_{w \in \mathcal{S}} J(u_{m-1} + w)$$

For each minimization problem to have a solution, the set \mathcal{S} should be chosen as a weakly closed set. Then, under classical assumptions on functional J (Fréchet differentiable and elliptic functional, with Fréchet differential uniformly bounded on bounded sets), the sequence strongly converges to u (see [6] for a convergence proof in the case $\mathcal{S} = \mathcal{R}_1$ in Hilbert Sobolev tensor spaces, and see also [14] for a more general result with tensor subsets \mathcal{S} in tensor Banach spaces).

4.3.3. Progressive construction with updates. Updated progressive PGD methods consist in constructing a sequence u_m obtained by first computing a “good correction” $w_m \in \mathcal{S}$ of u_{m-1} and then by updating $u_{m-1} + w_m$ using the information generated at steps $k \leq m$. Formally, we first define

$$w_m \in \Pi_{\mathcal{S}}(u - u_{m-1})$$

with $\Pi_{\mathcal{S}}$ defined as previously. Then, we construct a linear space \mathcal{V}_m based on the information generated at previous iterations and we define

$$u_m \in P_{\mathcal{V}_m}(u)$$

with $P_{\mathcal{V}_m}(u) \in V$ being a suitable approximation of u in \mathcal{V}_m . In practice, we choose \mathcal{V}_m such that

$$u_{m-1} + w_m \in \mathcal{V}_m \quad (4.3)$$

in order for u_m to be potentially better than $u_{m-1} + w_m$. The construction of spaces \mathcal{V}_m will be discussed below.

Sequence of updates at each iteration. A sequence of updates can also be performed, by defining a sequence of linear subspaces $\mathcal{V}_m^{(k)}$ and approximations $v_m^{(k)} \in \mathcal{V}_m^{(k)}$ as follows: we let $v_m^{(0)} = u_{m-1} + w_m$,

$$v_m^{(k)} = P_{\mathcal{V}_m^{(k)}}(u), \quad 1 \leq k \leq N_m,$$

with N_m the number of updates, and finally, we let $u_m = v_m^{(N_m)}$. The linear spaces should be constructed such that $v_m^{(k-1)} \in \mathcal{V}_m^{(k)}$.

Definition of updates. For optimization problem (3.2), $P_{\mathcal{V}_m}(u)$ is naturally chosen as the unique minimizer of J on the linear subspace \mathcal{V}_m , i. e.

$$u_m = P_{\mathcal{V}_m}(u) = \arg \min_{v \in \mathcal{V}_m} J(v)$$

uniquely characterized by

$$\langle J'(u_m), v \rangle = 0 \quad \forall v \in \mathcal{V}_m$$

or equivalently by

$$\mathcal{A}(u_m, v) = \mathcal{L}(v) \quad \forall v \in \mathcal{V}_m \quad (4.4)$$

For more general problems (3.1), $u_m = P_{\mathcal{V}_m}(u)$ can typically be selected as the Galerkin projection of u on \mathcal{V}_m , defined by (4.4).

4.3.4. Strategies for updates. We here consider the case where $\mathcal{S} = \mathcal{R}_1$ (rank-one updates) and we denote by $\{w_i\}_{i=1}^m$ with $w_i \in \mathcal{R}_1$ the set of rank-one updates that have been generated from iterations 1 to m . We now propose different ways of constructing linear subspaces \mathcal{V}_m :

1. $\mathcal{V}_m = \{\sum_{i=1}^m \alpha_i w_i; \alpha_i \in \mathbb{R}\}$. In this case, we have $u_m \in \mathcal{R}_m$ and $\dim(\mathcal{V}_m) = m$.
2. $\mathcal{V}_m = \mathcal{R}_m^l(v_m) = \{\sum_{i=1}^m w_i^1 \otimes \dots \otimes \delta w_i^l \otimes \dots \otimes w_i^d; \delta w_i^l \in V_l\}$. It corresponds to an actualization of all the functions in V_l that have been generated. In this case, we have $u_m \in \mathcal{R}_m$ and $\dim(\mathcal{V}_m) = m \times \dim(V_l)$. A sequence of updates can be introduced by choosing successively $\mathcal{V}_m^{(k)} = \mathcal{R}_m^{l_k}(v_m^{(k-1)})$ with a particular sequence l_1, \dots, l_{N_m} . It corresponds to updates of functions along N_m selected dimensions.⁶
3. $\mathcal{V}_m = U_1^m \otimes \dots \otimes U_d^m$ with $U_k^m = \text{span}\{w_i^k\}_{i=1}^m \subset V_k$. In this case, we have $u_m \in \mathcal{T}_{\mathbf{r}^m}$, the set of rank- \mathbf{r}^m tensors, with $\mathbf{r}^m = (r_1^m, \dots, r_d^m)$ and $r_k^m \leq m$ for all k . We have $\dim(\mathcal{V}_m) = \prod_{k=1}^d r_k^m \leq m^d$, which makes this updating strategy tractable only for small d (typically $d \leq 3$) or eventually for higher d if most of the $\{r_k^m\}_{k=1}^d$ remain very small when increasing m , e.g. if the V_k have very small dimensions or if the solution u admits a small minimal subspace $U_{k,\min}(u) \subset V_k$ associated with dimension k (see [12] for a definition of minimal subspaces).

REMARK 4.2 (An alternative to the construction of linear spaces). *Instead of constructing a linear subspace \mathcal{V}_m , any other subset satisfying $u_{m-1} + w_m \in \mathcal{V}_m$ could also be introduced. Let us consider the case where $u_{m-1} + w_m = \sum_{i=1}^m w_i \in \mathcal{R}_m$. A possible strategy consists in defining $\mathcal{V}_m = \sum_{i=1, i \neq l}^m w_i + \mathcal{S}$. The updating step then corresponds to the actualization of the term $w_l \in \mathcal{S}$. Several successive updates can be performed in order to update other rank-one elements. A succession of such updating steps at iteration m can be seen as an iterative procedure for approximating a best approximation in \mathcal{R}_m .*

4.4. SVD as a particular case of PGD. Let us consider the particular case where

$$J(v) = \frac{1}{2} \mathcal{A}(v, v) - \mathcal{L}(v)$$

⁶Note that the updating step along a particular dimension l could be unaffordable in some practical applications, because of the possibly high dimension of the space V_l .

with \mathcal{A} a symmetric coercive and continuous bilinear form on $V \times V$, and \mathcal{L} a continuous linear form on V . \mathcal{A} defines a norm on V , denoted $\|\cdot\|_{\mathcal{A}}$, which is equivalent to the norm $\|\cdot\|$. We have

$$J(v) = \frac{1}{2}\|v - u\|_{\mathcal{A}}^2 - \frac{1}{2}\|u\|_{\mathcal{A}}^2$$

The direct PGD then corresponds to a best approximation of the solution on a particular tensor subset \mathcal{S}_m in the following sense:

$$u_m \in \arg \min_{v \in \mathcal{S}_m} J(v) = \arg \min_{v \in \mathcal{S}_m} \|v - u\|_{\mathcal{A}}$$

The progressive PGD corresponds to successive best approximation problems on a tensor subset \mathcal{S} in the following sense:

$$w_m \in \arg \min_{w \in \mathcal{S}} J(u_{m-1} + w) = \arg \min_{w \in \mathcal{S}} \|u_{m-1} + w - u\|_{\mathcal{A}}$$

For the case where $\mathcal{S} = \mathcal{R}_1$, this problem has been interpreted as a generalization of a singular value decomposition [13] and we have the following result: for the progressive PGD,

$$\|u - u_m\|_{\mathcal{A}}^2 = \|u\|_{\mathcal{A}}^2 - \sum_{i=1}^m \sigma_i^2 \xrightarrow{m \rightarrow \infty} 0$$

where σ_m can be interpreted as the (generalized) dominant singular value of $(u - u_{m-1})$, defined by

$$\sigma_m = \arg \max_{w \in \mathcal{R}_1, \|w\|_{\mathcal{A}}=1} \langle w, u - u_{m-1} \rangle_{\mathcal{A}}$$

with $\langle \cdot, \cdot \rangle_{\mathcal{A}}$ the inner product associated with $\|\cdot\|_{\mathcal{A}}$. In the case where \mathcal{A} is a rank-one bilinear form, i.e.

$$\mathcal{A}(\otimes_{k=1}^d w^k, \otimes_{k=1}^d v^k) = \prod_{k=1}^d \mathcal{A}_k(w^k, v^k)$$

with the $\mathcal{A}_k : V_k \times V_k \rightarrow \mathbb{R}$ being coercive symmetric continuous bilinear forms, then it can be proved that for $d = 2$, the PGD (direct or progressive) is equivalent to a classical singular value decomposition of u (see proposition 9 in [13]), where each space V_k is equipped with the particular metric induced by the inner product associated with \mathcal{A}_k . In a more general case, this analogy with a generalization of a SVD has not been explored rigourously (see [31] for some results and open questions in the case $d = 2$). Nevertheless, it is clear that the above results make the PGD a little more specific than a classical best approximation problem (because of the specific structure of the set of optimization). It is also clear that a better understanding of these decompositions could yield the development of new (and maybe more efficient) algorithms for their constructions.

4.5. Solution methods based on classical iterative solvers and PGD.

Proper Generalized Decomposition methods can be used in conjunction with classical iterative solvers. Compared to solution methods using a posteriori tensor approximations (see section 3.2), it allows the introduction of a larger class of iterative solvers based on the following definition of a sequence $\{u_n\}_{n \geq 0}$:

$$C_n(u_n) = D_n(u_{n-1}) \tag{4.5}$$

with $C_n : V \rightarrow V$ and $D_n : V \rightarrow V$ some suitable mappings. An approximate sequence $\{v_n\}_{n \geq 0}$ is constructed as follows. v_{n-1} being given, we define $v_n \in \mathcal{a} \otimes_{k=1}^d V_k$ as a Proper Generalized Decomposition of the solution v_n^* of

$$C_n(v_n^*) = D_n(v_{n-1}) \quad (4.6)$$

and such that

$$\|v_n^* - v_n\| \leq \epsilon$$

with ϵ a given precision. Iteration mappings C_n and D_n should satisfy the following requirements:

- C_n and D_n must preserve tensor formats and application of D_n must prevent from a dramatic increase of tensors ranks. Eventually, C_n and D_n could be approximated in order to fulfill this requirement.
- C_n should allow for the definition of a robust, convergent and controlled PGD.
- Algorithm must be stable with respect to perturbations. The approximate sequence should yield an error which depends continuously on the error ϵ .

Note that the approximate sequence $\{v_n\}$ can be formally defined as in equation (3.4), with iteration mapping B_ϵ involving the approximation with a PGD algorithm. The convergence result of Theorem 3.1 then also applies to this context.

A first example: Newton-type algorithm. Let us consider the case of a strongly monotone mapping A . Newton algorithm consists in defining the iteration mappings C_n and D_n as follows:

$$C_n(v) = A'(v_{n-1})(v),$$

with $A'(v_{n-1}) : V \rightarrow V$ the Fréchet differential of A at v_{n-1} , which is a linear coercive operator, and

$$D_n(v) = C_n(v) - (A(v) - b)$$

It corresponds to the following definition of iteration mapping B :

$$B(v) = v - A'(v)^{-1}(A(v) - b)$$

Quasi-Newton methods consist in using for C_n a reasonable approximation of $A'(v_{n-1})$. They can allow the introduction of simpler operators C_n satisfying the desired requirements for an efficient and robust application of PGD.

A second example: Operator splitting method. Operator splitting methods consist in decomposing operator A as follows:

$$A = L - N$$

with $L : V \rightarrow V$ a linear operator satisfying the requirements for an efficient and robust application of PGD (*e.g.* a symmetric coercive continuous operator). Iteration mappings can then be defined as follows:

$$C_n = L, \quad D_n = N$$

corresponding to the following definition of iteration mapping B :

$$B(v) = L^{-1}(N(v) + b)$$

A third example: the LATIN method. Note that the PGD was initially introduced in conjunction with an iterative method called the LATIN method [27]. The LATIN method is a general nonlinear iterative solver for evolution problems encountered in nonlinear solid mechanics. This iterative solver is defined on the whole space-time domain and constructs a sequence of approximations that are solutions of a sequence of linear evolution problems (4.6). The PGD was then introduced in the early 80's in order to obtain an efficient solution of these linear evolution problems and an efficient storage of intermediate space-time solutions. For quasi static problems, LATIN algorithms can be seen as modified Quasi-Newton algorithms, and the operator C_n is usually a simple parabolic operator which makes robust the basic definitions of PGD based on Galerkin orthogonality criteria (see the following section).

5. Alternative definitions of progressive PGDs. In this section, we answer the questions: given an approximation $u_m \in \mathcal{S}_m$, how to define a correction $w \in \mathcal{S}$ and which algorithm can we use to construct it? In fact, different possible answers are introduced, depending on the properties of the considered variational problem. For each definition, we will detail the important case $\mathcal{S} = \mathcal{R}_1$ and introduce heuristic alternated direction algorithms.

5.1. Convex optimization problems. We first recall the case of optimization problem (3.2), already mentioned in the previous section 4.3.2.

5.1.1. Definition of a candidates set. In this case, the natural definition of an optimal correction $w \in \mathcal{S}$ is

$$w \in \Pi_{\mathcal{S}}(u - u_m) = \arg \min_{v \in \mathcal{S}} J(u_m + v) \quad (5.1)$$

The convergence of the resulting definition of PGD has been studied in [6] for the case $\mathcal{S} = \mathcal{R}_1$ in Hilbert Sobolev tensor spaces, and in [14] in a more general framework in tensor Banach spaces. Let us now discuss the problem (5.1) and its resolution.

5.1.2. Necessary conditions of optimality. A necessary (but not sufficient) condition of optimality of $w \in \mathcal{S}$ reads:

$$\langle J'(u_m + w), \delta w \rangle = 0 \quad \forall \delta w \in T_w(\mathcal{S})$$

where $T_w(\mathcal{S})$ is the tangent space⁷ to the manifold \mathcal{S} at w , or equivalently

$$\mathcal{A}(u_m + w, \delta w) = \mathcal{L}(\delta w) \quad \forall \delta w \in T_w(\mathcal{S}) \quad (5.2)$$

For the particular case $\mathcal{S} = \mathcal{R}_1$, one can prove that the tangent space $T_w(\mathcal{R}_1)$ at $w = \otimes_{k=1}^d w^k$ is

$$T_w(\mathcal{R}_1) = \mathcal{R}_1^1(w) + \dots + \mathcal{R}_1^d(w) \quad (5.3)$$

with the $\mathcal{R}_1^k(w)$ being linear subspaces defined by

$$\mathcal{R}_1^k(w) = \{w^1 \otimes \dots \otimes w^{k-1} \otimes \delta w^k \otimes w^{k+1} \otimes \dots \otimes w^d; \delta w^k \in V_k\} \quad (5.4)$$

Necessary optimality condition (5.2) is then equivalent to a coupled system of d orthogonality conditions:

$$\mathcal{A}(u_m + w, \delta w) = \mathcal{L}(\delta w) \quad \forall \delta w \in \mathcal{R}_1^k(w), 1 \leq k \leq d \quad (5.5)$$

⁷We here define the tangent space by $T_w(\mathcal{S}) = \{z = \alpha z^* : \alpha \in \mathbb{R}, z^* \in V, \exists \{w^k\}_{k \in \mathbb{N}} \subset \mathcal{S} \text{ such that } w = \lim w^k \text{ and } z^* = \lim \frac{w - w^k}{\|w - w^k\|}\}$

5.1.3. Construction for the particular case $\mathcal{S} = \mathcal{R}_1$. For the case $\mathcal{S} = \mathcal{R}_1$, a possible way for trying to construct an optimal $w = \otimes_{k=1}^d w^k \in \mathcal{R}_1$ consists in using an alternated direction algorithm, by solving successively minimization problems of type

$$\min_{w^l \in V_l} J(u_m + \otimes_{k=1}^d w^k) \quad (5.6)$$

for $l \in \{1 \dots d\}$. Equation (5.6) is then equivalent, for a given $w = \otimes_{k=1}^d w^k \in \mathcal{R}_1$, to find $w^\diamond \in \mathcal{R}_1^l(w)$ such that

$$w^\diamond = \arg \min_{\hat{w} \in \mathcal{R}_1^l(w)} J(u_m + \hat{w}) \quad (5.7)$$

Euler-Lagrange equation associated with problem (5.7) is

$$w^\diamond \in \mathcal{R}_1^l(w), \quad \mathcal{A}(u_m + w^\diamond, \delta w) = \mathcal{L}(\delta w) \quad \forall \delta w \in \mathcal{R}_1^l(w) \quad (5.8)$$

Let us denote by $f_m^l(w)$ the unique solution of problem (5.8), where $f_m^l : \mathcal{R}_1 \rightarrow \mathcal{R}_1$ is a well-defined mapping. Starting from an initial guess $w^{(0)} \in \mathcal{R}_1$, we then construct a sequence $\{w^{(n)}\}_{n \in \mathbb{N}}$ defined by

$$w^{(n+1)} = f_m^d \circ \dots \circ f_m^1(w^{(n)}) \quad (5.9)$$

REMARK 5.1. *In practice, we often observe a relatively fast convergence of the sequence $\{w^{(n)}\}$ towards an element \tilde{w} which satisfies simultaneously the stationarity conditions $\tilde{w} = f_m^l(\tilde{w})$ for all $l \in \{1 \dots d\}$. The behavior of the algorithm can be understood by the analogy between this alternated direction algorithm and a power iteration algorithm for capturing the dominant singular value of a generalized singular value decomposition [13]. This analogy is fully justified for $d = 2$ and for a particular structure of the problem [32]. However, to the knowledge of the authors, there is still no theoretical result about the properties of this algorithm in the general case.*

5.2. Minimal Residual PGD (MR-PGD).

5.2.1. Definition of a candidates set. For general problems (3.1), nonsymmetric or even nonlinear, a possible strategy consists in reformulating the problem as an optimization problem with a minimal residual formulation. We introduce a symmetric coercive and continuous operator $M : V \rightarrow V$, defining on V an inner product $\langle \cdot, \cdot \rangle_M = \langle M(\cdot), \cdot \rangle$ and associated norm $\|\cdot\|_M$, equivalent to the initial norm $\|\cdot\|$. We then introduce the following definition of an optimal update $w \in \mathcal{S}$:

$$w \in \Pi_{\mathcal{S}}(u - u_m) = \arg \min_{v \in \mathcal{S}} \|A(u_m + v) - b\|_M$$

Necessary conditions of optimality of $w \in \mathcal{S}$ reads:

$$\langle A(u_m + w), A'(u_m + w)(\delta w) \rangle_M = \langle b, A'(u_m + w)(\delta w) \rangle_M \quad \forall \delta w \in T_w(\mathcal{S})$$

with $A'(u_m + w) : V \rightarrow V$ the tangent operator of A at $u_m + w$. For linear problems, since $A'(v) = A$, it is equivalent to

$$\langle A^* M A(u_m + w), \delta w \rangle = \langle A^* M b, \delta w \rangle \quad \forall \delta w \in T_w(\mathcal{S}) \quad (5.10)$$

where A^* is the adjoint operator of A .

5.2.2. Construction for the particular case $\mathcal{S} = \mathcal{R}_1$. For the construction of an optimizer in $\mathcal{S} = \mathcal{R}_1$, an alternated minimization algorithm can be applied. It consists in constructing the sequence $\{w^{(n)}\}_{n \in \mathbb{N}} \subset \mathcal{R}_1$ defined by iterations (5.9), where for a given $l \in \{1, \dots, d\}$, $w^\diamond = f_m^l(w) \in \mathcal{R}_1^l(w)$ is the minimizer of the residual on subspace $\mathcal{R}_1^l(w)$, solution of

$$\langle A(u_m + w^\diamond), A'(u_m + w^\diamond)(\delta w) \rangle_M = \langle b, A'(u_m + w^\diamond)(\delta w) \rangle_M \quad \forall \delta w \in \mathcal{R}_1^l(w)$$

For linear problems, $w^\diamond = f_m^l(w) \in \mathcal{R}_1^l(w)$ is the solution of

$$\langle A^* M A(u_m + w^\diamond), \delta w \rangle = \langle A^* M b, \delta w \rangle \quad \forall \delta w \in \mathcal{R}_1^l(w) \quad (5.11)$$

This construction of PGD is robust in the sense that the residual norm monotonically decreases with m :

$$\|A(u_{m+1}) - b\|_M \leq \|A(u_m) - b\|_M$$

However, these reformulations are often uneasy from theoretical and technical point of views. Moreover, they often lead to PGD decompositions with bad convergence properties with respect to usual norms [32], as it will be illustrated in the examples. Indeed, the convergence strongly depends on the choice of norm $\|\cdot\|_M$ and suitable residual norms are often difficult to compute in practice (“good” residual norms usually do not preserve tensor format).

5.3. Galerkin PGD (G-PGD).

5.3.1. Definition of a candidates set. Galerkin PGD consists in defining a new element $w \in \mathcal{S}$ from a Galerkin orthogonality condition:

$$\mathcal{A}(u_m + w, \delta w) = \mathcal{L}(\delta w) \quad \forall \delta w \in T_w(\mathcal{S}) \quad (5.12)$$

The set of solutions of (5.12) defines a candidate set $\Pi_{\mathcal{S}}(u - u_m) \subset \mathcal{S}$. The problem with this definition is that there is no clear criterium for selecting a good solution. Indeed, the set $\Pi_{\mathcal{S}}(u - u_m)$ not only contains good solutions. In fact, this selection comes in practice from the proposed algorithms (see below).

5.3.2. Construction for the particular case $\mathcal{S} = \mathcal{R}_1$. Knowing $u_m \in \mathcal{R}_m$, we construct a sequence $\{w^{(n)}\}_{n \in \mathbb{N}} \subset \mathcal{R}_1$ defined by iterations (5.9), where $w^\diamond = f_m^l(w) \in \mathcal{R}_1^l(w)$ is the solution of equation

$$\mathcal{A}(u_m + w^\diamond, \delta w) = \mathcal{L}(\delta w) \quad \forall \delta w \in \mathcal{R}_1^l(w) \quad (5.13)$$

$w^\diamond = f_m^l(w)$ is a Galerkin approximation of $(u - u_m)$ on the subspace $\mathcal{R}_1^l(w)$. If the sequence $w^{(n)}$ converges towards an element $\tilde{w} \in \mathcal{R}_1$, this element \tilde{w} generally verifies simultaneously the Galerkin orthogonality conditions $\tilde{w} = f_m^l(\tilde{w})$, for all $l \in \{1, \dots, d\}$. We may also observe a non convergence of the sequence $w^{(n)}$. However, after a few iterations, the iterate $w^{(n)}$ can be selected as a good candidate. Note that in the case where the problem is associated with the optimization of a functional J , this algorithm is the same as the one presented in section 5.1.3 (alternated minimization algorithm).

Although there is no guaranty for convergence of the sequence u_m with m , good convergence properties of u_m are observed for a large class of linear problems [31, 32] or nonlinear problems [34], for the case of stochastic/deterministic separation, and for linear problems in the case of multidimensional separation [33]. However, for some nonsymmetric problems, the decomposition may present very bad convergence properties or even diverge [32].

5.4. Minimax PGD (MM-PGD). An alternative definition of PGD, called Minimax PGD, has been proposed in [32] in order to improve convergence properties of progressive PGD with respect to a specified metric. It can be interpreted as a Petrov-Galerkin PGD. It has been initially introduced for time-dependent partial differential equations. Here, it is extended to general non symmetric linear problems formulated in tensor product spaces. The construction of the Minimax PGD leads to computation times that are similar to those of the classical Galerkin PGD. Therefore, for a given accuracy, it usually leads to lower rank approximations and lower computation times, compared to Galerkin and Minimal Residual PGD.

5.4.1. Definition of a candidates set. We here restrict the presentation to linear problems. Let us introduce the functional $\mathcal{M}_m : V \times V \rightarrow \mathbb{R}$ defined by

$$\mathcal{M}_m(w, \tilde{w}) = \frac{1}{2} \langle w, w \rangle_* - \mathcal{A}(u_m + w, \tilde{w}) + \mathcal{L}(\tilde{w}) \quad (5.14)$$

with $\langle \cdot, \cdot \rangle_*$ an inner product on V with associated norm $\| \cdot \|_*$ equivalent to the norm $\| \cdot \|$. The choice of this norm will have a consequence on the constructed sequence u_m . An optimal update w of u_m can then be defined with the following max-min problem:

$$\max_{\tilde{w} \in \mathcal{S}} \min_{w \in \mathcal{S}} \mathcal{M}_m(w, \tilde{w}) \quad (5.15)$$

where two elements w and \tilde{w} in \mathcal{S} are constructed simultaneously. The candidates set $\Pi_{\mathcal{S}}(u - u_m)$ can here be defined as

$$\Pi_{\mathcal{S}}(u - u_m) = \{w \in \arg \min_{w \in \mathcal{S}} \mathcal{M}_m(w, \tilde{w}); \tilde{w} \in \arg \max_{\tilde{w} \in \mathcal{S}} \min_{w \in \mathcal{S}} \mathcal{M}_m(w, \tilde{w})\}.$$

Let us note that the problem

$$\mathcal{M}_m(v^\diamond, \tilde{v}^\diamond) = \max_{\tilde{v} \in V} \min_{v \in V} \mathcal{M}_m(v, \tilde{v}) \quad (5.16)$$

admits a unique solution with $v^\diamond = u - u_m$ (where u is the exact solution of the initial problem) and $\tilde{v}^\diamond = A^{*-1}v^\diamond$. Optimal w and \tilde{w} can then be seen as approximations in \mathcal{S} of v^\diamond and \tilde{v}^\diamond respectively.

REMARK 5.2. Note that functional \mathcal{M}_m can be equivalently written

$$\mathcal{M}_m(w, \tilde{w}) = \frac{1}{2} \|w - A^*(\tilde{w})\|_*^2 - \frac{1}{2} \|A^*(\tilde{w}) - (u - u_m)\|_*^2 + \frac{1}{2} \|u - u_m\|_*^2$$

where $A^*(\tilde{v}) \in V$ denotes the Riesz representant of linear form $\mathcal{A}(\cdot, \tilde{v})$ with respect to inner product $\langle \cdot, \cdot \rangle_*$, i.e. such that $\langle v, A^*(\tilde{v}) \rangle_* = \mathcal{A}(v, \tilde{v})$.

5.4.2. Construction for the particular case $\mathcal{S} = \mathcal{R}_1$. A possible way for constructing an optimal couple $(w, \tilde{w}) \in \mathcal{R}_1 \times \mathcal{R}_1$ consists in using an alternated direction algorithm, by solving successively problems of type

$$\max_{\tilde{w}^l \in V_l} \min_{w^l \in V_l} \mathcal{M}_m(\otimes_{k=1}^d w^k, \otimes_{k=1}^d \tilde{w}^k) \quad (5.17)$$

For given $(w, \tilde{w}) \in \mathcal{R}_1 \times \mathcal{R}_1$, problem (5.17) is equivalent to

$$\max_{\tilde{w}^\diamond \in \mathcal{R}_1^l(\tilde{w})} \min_{w^\diamond \in \mathcal{R}_1^l(w)} \mathcal{M}_m(w^\diamond, \tilde{w}^\diamond) \quad (5.18)$$

where linear spaces $\mathcal{R}_1^l(w)$ and $\mathcal{R}_1^l(\tilde{w})$ are defined by (5.4). Equation (5.18) is a classical saddle point problem which admits a unique solution $(w^\diamond, \tilde{w}^\diamond) \in \mathcal{R}_1^l(w) \times \mathcal{R}_1^l(\tilde{w})$ characterized by

$$\mathcal{A}(u_m + w^\diamond, \delta\tilde{w}) = \mathcal{L}(\delta\tilde{w}) \quad \forall \delta\tilde{w} \in \mathcal{R}_1^l(\tilde{w}) \quad (5.19)$$

$$\mathcal{A}(\delta w, \tilde{w}^\diamond) = \langle \delta w, w^\diamond \rangle_* \quad \forall \delta w \in \mathcal{R}_1^l(w) \quad (5.20)$$

Let us note that equations (5.19) and (5.20) can be solved one after the other. Equation (5.19) defines w^\diamond as a Petrov-Galerkin approximation of $(u - u_m)$ in approximation space $\mathcal{R}_1^l(w)$, with test space $\mathcal{R}_1^l(\tilde{w})$. Equation (5.20) is an adjoint problem. We denote by $(w^\diamond, \tilde{w}^\diamond) = F_m^l(w, \tilde{w})$ the solution of the system of equations (5.19)-(5.20). Starting for an initial guess $(w^{(0)}, \tilde{w}^{(0)}) \in \mathcal{R}_1 \times \mathcal{R}_1$, we then construct a sequence $\{(w^{(n)}, \tilde{w}^{(n)})\}_{n \in \mathbb{N}}$ defined by

$$(w^{(n+1)}, \tilde{w}^{(n+1)}) = F_m^d \circ \dots \circ F_m^1(w^{(n)}, \tilde{w}^{(n)}) \quad (5.21)$$

In practice, we often observe a convergence of the sequence $\{(w^{(n)}, \tilde{w}^{(n)})\}_{n \in \mathbb{N}}$. However, this algorithm requires further theoretical investigations and possible improvements.

5.5. \mathcal{S} -tangent problems. In this section, we introduce a particular class of problems for which we can ensure a priori the monotonic decrease of some residual error for the Galerkin PGD. We first describe the prototypical case of \mathcal{S} -tangent linear operators, before to generalize \mathcal{S} -tangency to bilinear forms. Then we detail the particular case $\mathcal{S} = \mathcal{R}_1$ and we give a class of practical examples of \mathcal{R}_1 -tangent problems.

5.5.1. \mathcal{S} -tangent operators. DEFINITION 5.3. *An operator $A : V \mapsto V$ is said \mathcal{S} -tangent if for all $w \in \mathcal{S}$, $Aw \in T_w(\mathcal{S})$, where $T_w(\mathcal{S})$ is the tangent space at w of the manifold \mathcal{S} . We now give the basic result about Galerkin PGD in the case of an \mathcal{S} -tangent operator.*

PROPOSITION 5.4. *Let $u_m = u_{m-1} + w_m$ be a m -term progressive Galerkin PGD approximation of u , with $w_m \in \mathcal{S}$ satisfying the following Galerkin orthogonality condition:*

$$\langle A(u_{m-1} + w_m), \delta w \rangle = \langle b, \delta w \rangle \quad \forall \delta w \in T_{w_m}(\mathcal{S}) \quad (5.22)$$

We assume that the operator A is linear and \mathcal{S} -tangent. Then, we have

$$\|b - Au_m\|^2 = \|b - Au_{m-1}\|^2 - \|Aw_m\|^2 \quad (5.23)$$

Proof. As the operator A is \mathcal{S} -tangent, we can take $\delta w = Aw_m$ as a test function in equation (5.22), which yields $\|Aw_m\|^2 = \langle b - Au_{m-1}, Aw_m \rangle$. Therefore, we obtain

$$\begin{aligned} \|b - Au_m\|^2 &= \|b - Au_{m-1}\|^2 + \|Aw_m\|^2 - 2\langle b - Au_{m-1}, Aw_m \rangle \\ &= \|b - Au_{m-1}\|^2 - \|Aw_m\|^2 \end{aligned}$$

□

EXAMPLE 5.5. *A basic example of \mathcal{S} -tangent operator is given by $A = I_V$, the identity of V . In this particular case, equation (5.23) gives the monotonic decrease of $u_m - u$ in V -norm. An example of such a practical situation is given by $V = H_0^1(\Omega)$ equipped with the inner product $\langle u, v \rangle = \int_\Omega \nabla u \cdot \nabla v$, which corresponds to the variational formulation of the Laplace equation with homogeneous Dirichlet boundary conditions.*

5.5.2. \mathcal{S} -tangent bilinear forms. DEFINITION 5.6. *Let \mathcal{A} be a continuous bilinear form on $V \times V$. Then \mathcal{A} is said \mathcal{S} -tangent if there exists a \mathcal{S} -tangent operator $A_T : V \rightarrow V$ and a continuous, coercive and symmetric linear operator $M : V \rightarrow V$ such that*

$$\mathcal{A}(u, v) = \langle MA_T u, v \rangle \quad \forall u, v \in V$$

In other words, \mathcal{A} is \mathcal{S} -tangent if there exists a metric $\|\cdot\|_M$ on V (an inner product norm), equivalent to $\|\cdot\|$, in which \mathcal{A} is represented by an \mathcal{S} -tangent operator. The following result generalizes proposition 5.4 to \mathcal{S} -tangent bilinear forms. Roughly speaking, it says that the Galerkin PGD will not diverge if it is applied to an \mathcal{S} -tangent problem.

PROPOSITION 5.7. *Let $u_m = u_{m-1} + w_m$ be a m -term progressive Galerkin PGD approximation of u , satisfying the following Galerkin orthogonality condition:*

$$\langle A(u_{m-1} + w_m), \delta w \rangle = \langle b, \delta w \rangle \quad \forall \delta w \in T_{w_m}(\mathcal{S}) \quad (5.24)$$

We suppose that the bilinear form \mathcal{A} is \mathcal{S} -tangent and can be associated with a couple of operators (M, A_T) as defined in definition 5.6. Then, we have

$$\|b - Au_m\|_{M^{-1}}^2 = \|b - Au_{m-1}\|_{M^{-1}}^2 - \|Aw_m\|_{M^{-1}}^2 \quad (5.25)$$

Proof. We inject $\delta w = A_T w_m = M^{-1}Aw_m$ as test function in (5.24). This gives

$$\langle A(u_{m-1} + w_m), M^{-1}Aw_m \rangle = \langle b, M^{-1}Aw_m \rangle,$$

from which we deduce $\langle Aw_m, Aw_m \rangle_{M^{-1}} = \langle b - Au_{m-1}, Aw_m \rangle_{M^{-1}}$. Equation (5.25) follows immediately. \square

REMARK 5.8. *Note that proposition 5.7 does not provide a convergence result for the sequence u_m constructed with the progressive Galerkin PGD.*

EXAMPLE 5.9. *As a generalization of the basic example 5.5, we can consider the case of a symmetric and coercive bilinear form $\mathcal{A}(u, v) = \langle Au, v \rangle$. Then $M = A$ defines an equivalent metric on V and $A_T = I_V$ is a \mathcal{S} -tangent operator associated with A . Thus, convex optimization problems of section 5.1 appear as particular cases of \mathcal{S} -tangent problems. An example of such a practical situation is given by $V = H_0^1(\Omega)$ equipped with the inner product $\langle u, v \rangle = \int_{\Omega} \nabla u \cdot \nabla v$, and $\mathcal{A}(u, v) = \int_{\Omega} \nabla u \cdot \nabla v + \int_{\Omega} \alpha uv$ for any positive function α . This corresponds to the variational formulation of a diffusion-reaction equation with homogeneous Dirichlet boundary conditions.*

5.5.3. The particular case of \mathcal{R}_1 -tangent problems. We remind that for the particular case $\mathcal{S} = \mathcal{R}_1$, the tangent space $T_w(\mathcal{R}_1)$ at $w = \otimes_{k=1}^d w^k$ is defined by (5.3). Thus, the general form of an \mathcal{R}_1 -tangent operator A_T is given by

$$A_T = \sum_{i=1}^d I_1 \otimes \cdots \otimes A_T^i \otimes \cdots \otimes I_d$$

with I_i the identity operator on V_i and $A_T^i : V_i \rightarrow V_i$ an arbitrary operator on V_i . This kind of operator is known in the literature as the tensor sum of the A_T^i , or the kronecker sum in the finite dimensional case. Algebraic properties and various applications of kronecker sum can be found for instance in [21] and [4]. Some of the spectral properties of kronecker sum extend to the infinite dimensional case [22, 23],

and preservation properties of tensor sum have been recently investigated [26]. We denote this operation by \boxplus , so that $A_T = \boxplus_{i=1}^d A_T^i$.

Examples 5.5 and 5.9 of \mathcal{S} -tangent problems are quite trivial. They correspond to the most simple case of bilinear forms associated with \mathcal{S} -tangent operators $A_T = I_V$ and transformation operators $M = I_V$ and $M = A$ respectively. We now give a more interesting and representative class of \mathcal{S} -tangent problems in the particular case $\mathcal{S} = \mathcal{R}_1$, as a proposition whose proof is straightforward.

PROPOSITION 5.10. *For each $i \in \{1, \dots, d\}$, let A_i be a bounded linear operator on V_i and M_i be a linear bounded coercive symmetric operator on V_i . Suppose that the norm $\|\cdot\|$ on V is a crossnorm, that is $\|\otimes_{k=1}^d w^k\| = \prod_{k=1}^d \|w^k\|_k$ for all $\otimes_{k=1}^d w^k \in \mathcal{R}_1$. We define the bilinear form \mathcal{A} for $v = \otimes_{k=1}^d v^k \in \mathcal{R}_1$ and $w = \otimes_{k=1}^d w^k \in \mathcal{R}_1$ by*

$$\mathcal{A}(v, w) = \sum_{i=1}^d \langle A_i v^i, w^i \rangle_i \prod_{\substack{k=1 \\ k \neq i}}^d \langle M_k v^k, w^k \rangle_k$$

and we extend this definition by linearity to the entire tensor space V . Then, \mathcal{A} is \mathcal{R}_1 -tangent and an associated couple (M, A_T) (in the sense of definition 5.6) is given by $M = \otimes_{i=1}^d M_i$ and $A_T = \boxplus_{i=1}^d M_i^{-1} A_i$.

EXAMPLE 5.11. *In the case of finite dimensional tensor spaces, where all norms are equivalent to each other, we can always choose for $\|\cdot\|$ the canonical norm which is a crossnorm. Thus proposition 5.10 corresponds to many discretized versions of variational problems encountered in practical applications. A typical example is the discretized version, through Galerkin approximation, of the advection-diffusion-reaction equation*

$$-\Delta u + \beta \cdot \nabla u + \alpha u = f,$$

when $\alpha(x_1, \dots, x_d) = \sum_{i=1}^d \alpha_i(x_i)$ and $\beta(x_1, \dots, x_d) = \sum_{i=1}^d \beta_i(x_i)$. This example is discussed in section 6. However, even in such a particular case, the question of the \mathcal{R}_1 -tangency of \mathcal{A} in the continuous framework remains open.

6. Examples.

6.1. Notations. In this section, we consider progressive PGDs with the set $\mathcal{S} = \mathcal{R}_1$. For the different definitions of PGDs, we use the notation $(\lambda)\mu$ -PGD with $\mu = G, MR,$ and MM respectively for the Galerkin PGD, the Minimal Residual PGD and the MiniMax PGD (see section 5). λ refers to the type of updates eventually performed (see sections 4.3.3 and 4.3.4 for details). We remind that after having computed a correction $w_m \in \mathcal{R}_1$ of u_{m-1} , performing an update consists in constructing a subspace \mathcal{V}_m and in defining u_m as an approximation of the solution in \mathcal{V}_m . We denote by $\{w_i\}_{i=1}^m$, with $w_i = \otimes_{k=1}^k w_i^k \in \mathcal{R}_1$, the set of rank-one corrections that have been generated from iterations 1 to m of the progressive construction. $\lambda = S$ corresponds to the update of the first kind, i.e. $\mathcal{V}_m = \text{span}\{w_i\}_{i=1}^m$. $\lambda = T$ corresponds to the update of the third kind, i.e. $\mathcal{V}_m = U_1^m \otimes \dots \otimes U_d^m$ with $U_k^m = \text{span}\{w_i^k\}_{i=1}^m$. For the second kind of updates, all the dimensions of the problem will be updated (successive updates in $\mathcal{R}_m^k(u_m)$ for $1 \leq k \leq d$) and $\lambda = n \in \mathbb{N}$ will indicate the number of times these updates are performed.

6.2. Example 1: a linear symmetric problem. As a first example, we consider a linear symmetric partial differential equation on a hyper-rectangular domain

$\Omega = \Omega_1 \times \dots \times \Omega_d$:

$$\begin{cases} -\Delta u + \alpha u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \quad (6.1)$$

We here introduce piecewise linear finite element approximations space $V_k \subset H_0^1(\Omega_k)$, with 201 elements. We then work in a finite dimensional tensor space $V = V_1 \otimes \dots \otimes V_d \subset H_0^1(\Omega)$ with dimension $2^d 10^{2d}$. The only exception occurs when we use a finite element reference solution for $d = 3$. In this case, for memory usage reasons, we reduce the meshing to 30 elements, so that the dimension of the approximation space becomes $3^d 10^d$.

We here focus on the behavior of the G-PGD and the impact of updates for different types of problems: elliptic, non elliptic, and \mathcal{R}_1 -tangent. Different source terms will be selected in order to illustrate typical situations and corresponding behaviors of PGD methods. We will first consider finite rank source terms. We will then consider source terms built from manufactured solutions of sufficiently high representation rank. Finally, we will consider source terms built from manufactured solutions with low rank. This latter case deserves additional comments and will be treated separately. When studying the impact of the dimension d on the convergence of (λ) G-PGD, we often distinguish the d -dimensional cases for $d \geq 3$ and the 2-dimensional case, this latter case presenting some specificities.

6.2.1. Elliptic problems. We first consider the favorable case of elliptic problems, for which the G-PGD converges monotonically in operator norm (see section 4.4). In order to naturally extend to non elliptic and \mathcal{R}_1 -tangent problems in the following, we take for α the elementary tensor $\alpha(x_1, \dots, x_d) = a |\cos(b\pi x_1)| \dots |\cos(b\pi x_d)|$, with $a = 30$ and $b = 10$.

Low rank source terms. We here consider a source term $f = \sum_{i=1}^{r_f} \otimes_{k=1}^d f_i^k$, with $r_f = 10$ and $f_i^k(x_k)$ monomials of degree p_i^k randomly selected in $\{0, \dots, p_{max}\}$. The error is computed in residual norm. Figure 6.1 compares the convergence of G-PGD, (S)G-PGD, (T)G-PGD and (1)G-PGD for the 2-dimensional case.

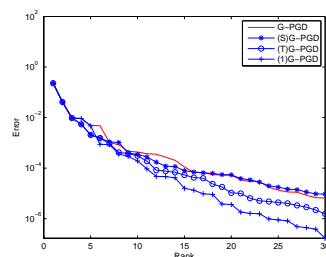


Fig. 6.1: (λ) G-PGD convergence. Error in residual norm. $d = 2$.

We remark that (S)update does not improve significantly the convergence of the G-PGD. This is observed in many applications. (T)update is more efficient but, as mentioned in section 4.3.4, this kind of update is dedicated to low dimensional problems (typically $d \leq 3$). (1)update is not limited to low dimensional problems, and it gives the most significant convergence improvement. In the following, for higher dimensional problems, we will restrict ourselves to this latter kind of updates.

REMARK 6.1. We remind that a (1)update, in order to compute a rank- m approximation u_m , requires the solution of a succession of problems in $(V_k)^m$, for $k = 1 \dots d$.

In the present application, it corresponds to linear problems of dimension $2 \cdot 10^2 \times m$. These updates become more and more computationally expensive as m increases.

Figures 6.2a, 6.2b and 6.2c compare the convergence of (n) G-PGD for $0 \leq n \leq 3$ respectively for $d = 3, 4$ and 5 . Again, we observe a significant improvement of the convergence with the (1)G-PGD, but no significant further improvement with additional updates ($n \geq 2$). However, we observe that the error increases with the dimension for a fixed decomposition rank. Furthermore, convergence of G-PGD in residual norm becomes non monotonic for high dimensions.

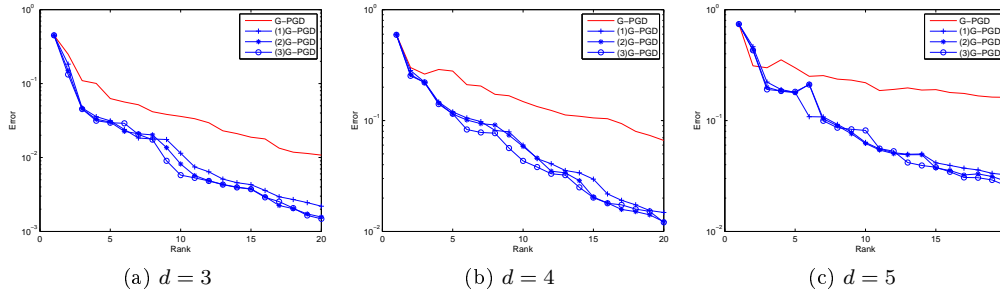
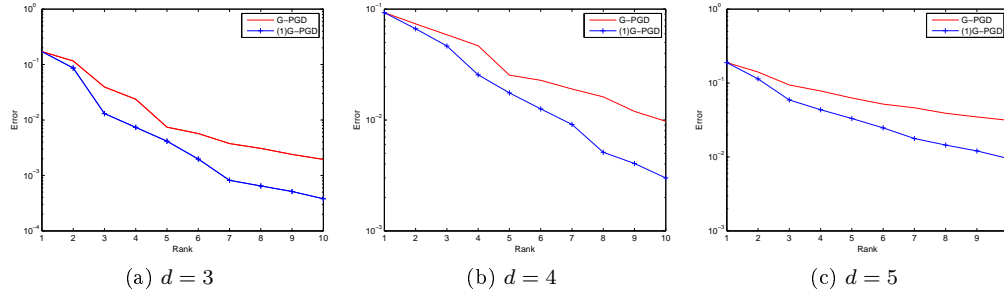
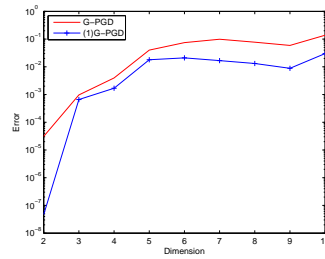


Fig. 6.2: (n) G-PGD convergence. Error in residual norm.

Manufactured solutions. For a better analysis of the behavior of algorithms when increasing the dimension, we will now consider manufactured solutions in order to have a reference solution and to measure convergence properties in usual norms. We now consider a source term f built from a manufactured solution $u = \sum_{i=1}^{r_u} \otimes_{k=1}^d u_i^k$, with $u_i^k(x_k)$ monomials of degree p_i^k randomly selected in $\{0, \dots, 10\}$. The representation rank r_u is first set to 20. The error is computed in operator norm (equivalent to the H^1 norm in the present case), for which we can expect monotonic decrease. Figures 6.3a, 6.3b and 6.3c illustrate this fact by plotting convergences of (n) G-PGDs respectively for $d = 3, 4$ and 5 . Figure 6.4 gives the evolution of error with respect to the dimension for a fixed decomposition rank $m = 10$. When increasing dimension, one observes a slowly increasing gap between G-PGD and (1)G-PGD. However, we observe that for high dimension, the error for fixed rank $m = 10$ is quite insensitive to the dimension d . Figure 6.5a (resp. figure 6.5b) plots the evolution with d of the approximation rank m required to reach a fixed error $\varepsilon = 10^{-2}$ (resp. $\varepsilon = 10^{-3}$), for a solution rank $r_u = 30$ (resp. $r_u = 50$). The interest of performing updates is very clear in both cases. Figure 6.5b shows that a second update allows to slightly improve convergence from dimension $d = 5$, although this is not necessarily a relevant choice in terms of computational cost.

6.2.2. Non elliptic problems. We now consider a non elliptic version of the above problem, where we choose $\alpha(x_1, \dots, x_d) = a \cos(b\pi x_1) \otimes \dots \otimes \cos(b\pi x_d)$, and $f = 1$. PGD approximations are compared to the reference finite element approximation u and the error is measured in L^2 norm. Figure 6.6a plots the convergence of (λ) G-PGD for dimension $d = 2$. For this choice of α , we no longer have any property of convergence for the G-PGD and actually, we here observe a divergence. This figure also shows that it is possible to recover convergence by performing updates. We

Fig. 6.3: (n) G-PGD convergence. Error in operator norm.Fig. 6.4: (n) G-PGD error in operator norm versus d at fixed $m = 10$.

remark that a S-update is sufficient to recover convergence⁸, T-update and (1)-update significantly improve the convergence. Figure 6.6b plots the convergence of G-PGD for $d = 3$. It shows that for dimension $d = 3$, convergence is no longer recoverable with the proposed updates⁹.

We then propose to use the MR-PGD (PGD based on minimal residual formulation). Figure 6.7 shows the convergence of (λ) MR-PGD for the same case $d = 3$. We observe a slow convergence rate for MR-PGD, and only a slight improvement with updates of type (S). Updates of type (T) are unaffordable in this case because they lead to prohibitive memory costs. However, we observe that (1)MR-PGD presents nice convergence results, thus showing again the significant improvement obtained with the second strategy of update.

6.2.3. \mathcal{R}_1 -tangent problems. As seen in section 5.5, in the particular case of equation (6.1), the operator A associated with the differential operator $(-\Delta + \alpha I)$ is \mathcal{R}_1 -tangent if and only if $\alpha(x_1, \dots, x_d) = \sum_{i=1}^d \alpha_i(x_i)$. To illustrate the impact of \mathcal{R}_1 -tangency, we choose such an α with $\alpha_i(x_i) = a \cos(b\pi x_i)$. It leads to a non-elliptic but \mathcal{R}_1 -tangent operator A . We choose a source term $f = 1$. Figures 6.8a, 6.8b and 6.8c illustrate the convergences of (n) G-PGDs in residual norm respectively for $d = 3$, $d = 4$ and $d = 5$. We observe that the property of \mathcal{R}_1 -tangency yields good convergence properties.

Figure 6.9 plots the evolution of the error (in residual norm) with respect to the

⁸In our knowledge, it is the only typical case where this S-update is really helpful.

⁹We make the same observation for higher dimensions d .

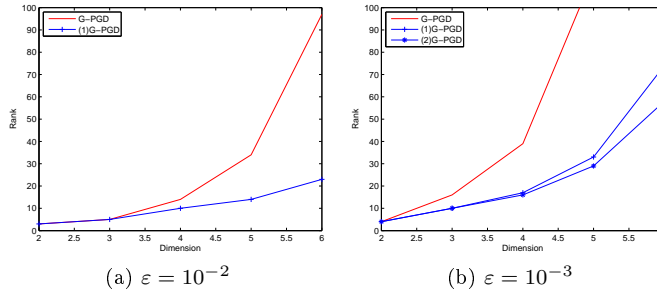


Fig. 6.5: (n) G-PGD rank vs d for reaching a fixed error ε .

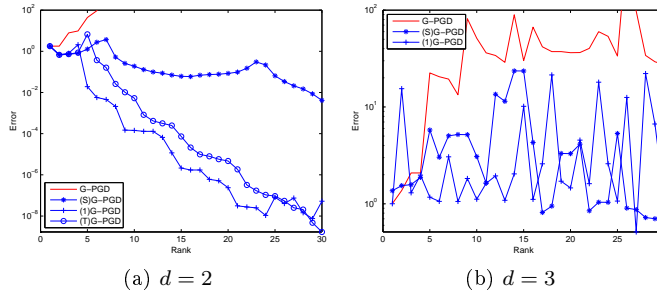
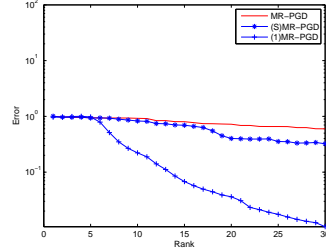
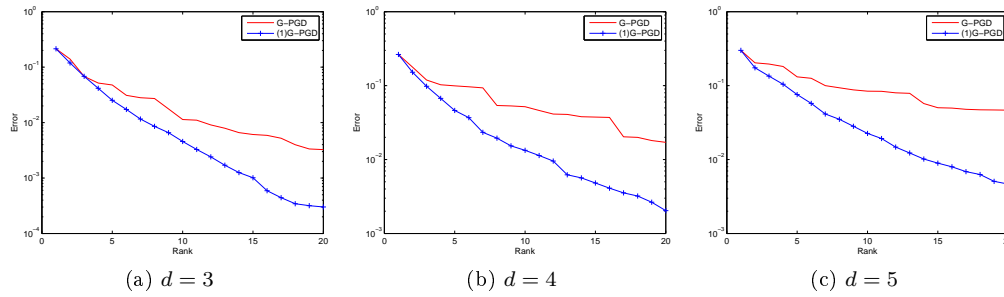


Fig. 6.6: (λ) G-PGD convergence. Error in L^2 norm.

dimension d for a fixed approximation rank $m = 10$. In this case, which is perhaps more persuasive than that of section 6.2.1 where we have used manufactured solutions, we have let the dimension goes to $d = 50$. Thus it confirm the stabilization of the error when the dimension grows, and it shows the capacity of G-PGD to solve really high-dimensional problems. Figure 6.10a (resp. figure 6.10b) plots the evolution with respect to the dimension d of the approximation rank m which is required to reach a residual norm of $\varepsilon = 10^{-2}$ (resp. $\varepsilon = 10^{-3}$). When increasing dimension, we observe a slowly increasing gap between G-PGD and (1)G-PGD, and also between (1)G-PGD and (2)G-PGD on figure 6.9. The above figures clearly illustrate the interest of performing updates.

6.2.4. The particular case of low rank solutions. Up to now, we have solved problems with low rank source terms, or manufactured solutions with a relatively high representation rank. This corresponds to typical situations and the previous results are representative of a general behavior of PGD methods. Now, we want to address the particular case where the solution is low rank and see if the proposed algorithm are able to efficiently capture this low rank solution.

First, we consider a solution $u = \sum_{i=1}^{r_u} a_i \otimes_{k=1}^d \phi_i^k$, with $\otimes_{k=1}^d \phi_i^k$ being orthonormal eigenfunctions of the operator $D = (-\Delta + \alpha I)$, and $\phi_i^k \neq \phi_j^k$ for $i \neq j$. It can be proved that in this case, the Singular Value Decomposition (SVD) (or its multidimensional counterpart for $d \geq 3$ [13]), the G-PGD and the proposed updated G-PGD are equivalent to each other and return the exact separated representation of u . Fig-

Fig. 6.7: (λ) MR-PGD convergence. Error in L^2 norm. $d = 3$.Fig. 6.8: (n) G-PGD convergence. Error in residual norm.

ure 6.11 illustrates an example of such a case for a solution rank $r_u = 10$ and for dimension $d = 3$.

More generally, if we do not enforce $\phi_i^k \neq \phi_j^k$ for all $i \neq j$, or if we take for $\otimes_{k=1}^d \phi_i^k$ eigenfunctions of $(-\Delta)$ in place of eigenfunctions of D , then SVD, G-PGD and updated G-PGD will partially match, giving non representative convergence properties.

Contrariwise, if the functions $\otimes_{k=1}^d \phi_i^k$ are not close to eigenfunctions of D , (n) G-PGD may fail to find the expected approximation rank $m = r_u$ if n is not sufficiently high. Figure 6.12 illustrates this observation in the 2-dimensional case with $r_u = 2$, the ϕ_i^k being monomials of degree p_i^k randomly selected in $\{0, \dots, 5\}$. The case $d \geq 3$, not plotted here, is generally worse, that is (n) G-PGD may fail to reach an arbitrarily small precision for $m = r_u$. Let us mention that this behavior seems related to the progressive character of the construction, and not to the particular definitions of rank-one corrections. Indeed, MR-PGD or MM-PGD give similar results in this particular case of low rank solutions.

6.3. Example 2 : a linear non symmetric problem. Let's now consider the following linear non symmetric partial differential equation defined on a domain $\Omega = \Omega_1 \times \dots \times \Omega_d \subset \mathbb{R}^d$:

$$\begin{cases} -\varepsilon \Delta u + \beta \cdot \nabla u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \quad (6.2)$$

We here introduce piecewise linear finite element approximations space $V_k \subset H_0^1(\Omega_k)$, with 1001 elements. We then work in a finite dimensional tensor space $V = V_1 \otimes \dots \otimes V_d \subset H_0^1(\Omega)$ with dimension 10^{3d} . Such a meshing becomes necessary only for low

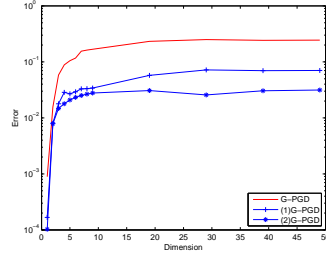


Fig. 6.9: (n) G-PGD error vs dimension at fixed rank $m = 10$

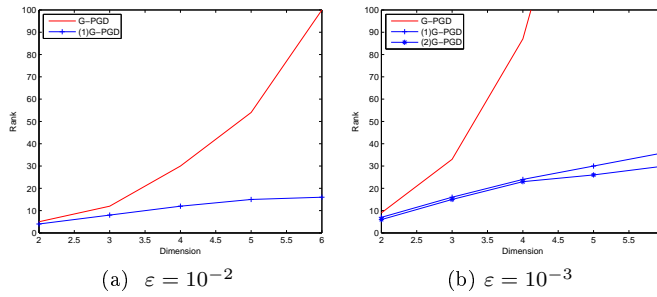


Fig. 6.10: (n) G-PGD rank vs d for reaching a fixed error ε .

values of ε , but we have kept it for all the simulations of this section by the sake of simplicity.

We will first analyze the influence of the parameter ε and of the dimension d on the convergence of the $(\lambda)\mu$ -PGDs on an elliptic problem. We will then investigate an non elliptic \mathcal{R}_1 -tangent problem. These examples will show the interest of introducing the MM-PGD and will also illustrate the utility (and sometimes the necessity) of performing updates.

6.3.1. Influence of ε . We here focus on the influence of ε , and we simply take $\beta = (1, \dots, 1)$ and $f = 1$. The error is measured in residual norm. Figure 6.13a (resp. figure 6.13b) gives the behavior of (n) G-PGD (resp (n) MM-PGD) for ε varying from 1 to 10^{-4} for the case $d = 2$. As the operator of the problem is \mathcal{R}_1 -tangent, G-PGD can not diverge but for $\varepsilon = 10^{-3}$, it presents very small convergence rates. Thus, when using G-PGD for these cases, performing updates is not only advised but mandatory. On the contrary, we notice a significant improvement with MM-PGD, which presents a good convergence rate down to $\varepsilon = 10^{-4}$. When performing updates, we observe similar behaviors for (n) MM-PGD and (n) G-PGD (see figure 6.14, where (n) G-PGD and (n) MM-PGD coincide for $\varepsilon = 10^{-4}$).

When dimension increases, we observe a rather different behavior: while updated versions of G-PGD and MM-PGD still give similar results, non updated MM-PGD is no longer better than non updated G-PGD. This is illustrated by figures 6.15a, 6.15b and 6.15c (resp. figures 6.16a, 6.16b and 6.16c) which plot the behavior of (n) G-PGD (resp. (n) MM-PGD) for ε varying from 1 to 10^{-4} in dimension $d = 3$, $d = 4$ and $d = 5$.

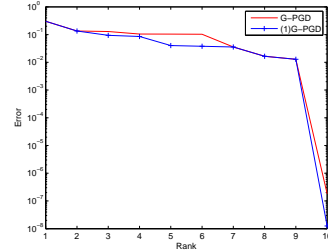


Fig. 6.11: (n) G-PGD convergence. Error in L^2 norm. $d = 3$. The solution u is of rank $r_u = 3$ and is a sum of eigenfunctions of $(-\Delta + \alpha I)$.

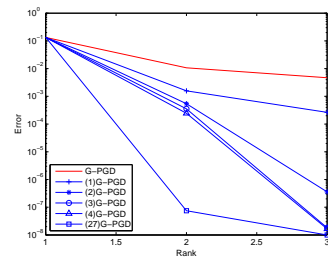


Fig. 6.12: G-PGD convergence. Error in L^2 norm. $d = 2$. The solution u is of rank $r_u = 10$ and is a sum of rank-one polynomial functions.

However, even if the error of G-PGD monotonically decreases due to the \mathcal{R}_1 -tangency property, it decreases too slowly in practice for small values of ε . This makes updates mandatory both for G-PGD and MM-PGD in these cases. For higher values of ε , G-PGD and MM-PGD present similar results, both in their non updated and updated versions. Thus we can consider that MM-PGD is at worst as good as G-PGD from the convergence point of view. Furthermore, we must mention that the alternated direction algorithm, which is used to compute the order m correction w_m , fails to converge for most m in the case of G-PGD, while it converges for all m in the case of MM-PGD (at least in the present application).¹⁰ This makes MM-PGD a more robust algorithm than G-PGD, even if it does not significantly improve convergence properties in all cases.

Finally, concerning the similarity of G-PGD and MM-PGD in their updated versions, we can mention that the definitions of updates that are used for MM-PGD are not really appropriate. Indeed, a Galerkin projection is used in both cases, while the natural definition in the case of the MM-PGD would rather be a Petrov-Galerkin projection, from which we could expect a more significant convergence improvement. However, the way to introduce a pertinent definition of updates based on Petrov-Galerkin projections for MM-PGD is still an open question.

6.3.2. A non elliptic \mathcal{R}_1 -tangent problem. We now consider a non elliptic but \mathcal{R}_1 -tangent version of problem (6.2). We take $\varepsilon = 1$ and $\beta_i = a \cos(b\pi x_i)$ with

¹⁰This can explain the loss of monotonic convergence in residual norm which is sometimes observed for G-PGDs, as illustrated in figure 6.15a.

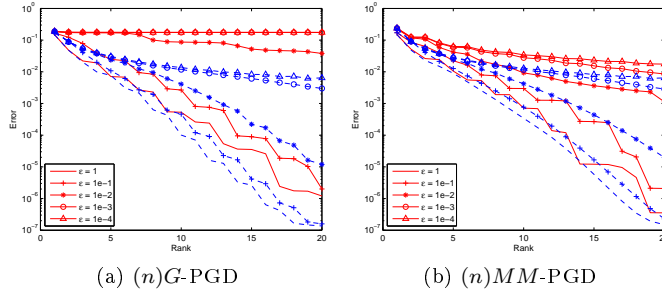


Fig. 6.13: Convergence of μ -PGDs (solid lines) and $(1)\mu$ -PGDs (dashed lines). Error in residual norm. $d = 2$.

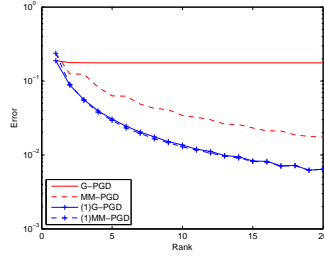


Fig. 6.14: Convergence of $(n)\mu$ -PGDs. $d = 2$, $\varepsilon = 10^{-4}$.

$a = 30$ and $b = 10$. In the case $d = 2$, figure 6.17a and 6.17b shows that G-PGD and MM-PGD give very similar results. They can be used without performing updates, although updates significantly improve their convergence. In both cases, we also observe that performing more than one update ($n \geq 2$) does not significantly improve the convergence.

For this example, the increase in dimension d does not have a significant impact on the behavior of the algorithms. We then restrict the following analysis to $d = 3$. Again, figure 6.18a and 6.18b show that G-PGD and MM-PGD give very similar results, both for updated and non updated versions. We can also check that performing more than one update does not improve the convergence. Finally, we see that in the case of a non elliptic and non symmetric \mathcal{R}_1 -tangent problem, there is no need to introduce MM-PGD since G-PGD possesses good convergence properties.

6.4. Example 3 : a convex non linear problem. We consider the following non linear partial differential equation defined on a hyper-rectangular domain $\Omega = \Omega_1 \times \dots \times \Omega_d$:

$$\begin{cases} -\varepsilon \Delta u + u^3 = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \quad (6.3)$$

We choose $\varepsilon = 10^{-4}$. We use piecewise linear finite element approximation spaces $V_k \subset H_0^1(\Omega_k)$, with 1001 elements. We then work in a finite dimensional tensor space $V = V_1 \otimes \dots \otimes V_d \subset H_0^1(\Omega)$ with dimension 10^{3d} . Note that the Galerkin approximation of this problem is the unique minimizer on V of the convex functional

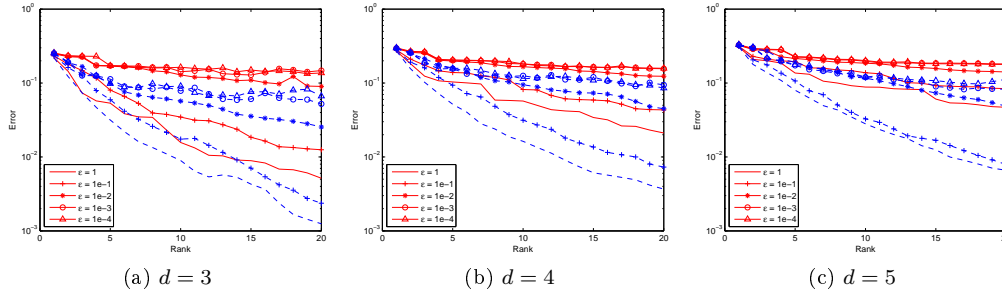


Fig. 6.15: Convergence of G-PGD (solid lines) and (1)G-PGD (dashed lines). Error in residual norm.

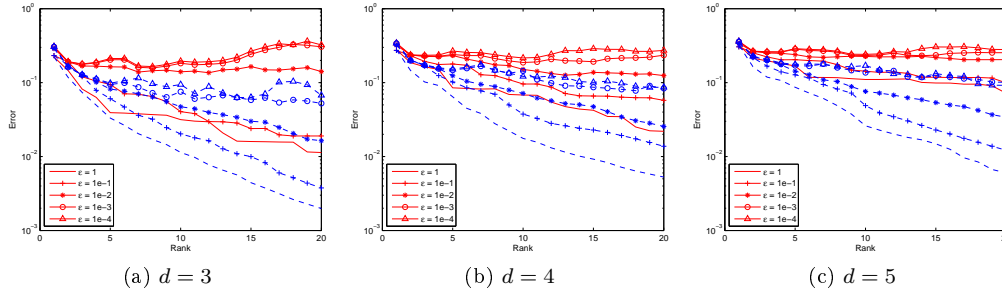


Fig. 6.16: Convergence of MM-PGD (solid lines) and (1)MM-PGD (dashed lines). Error in residual norm.

given by

$$J(v) = \frac{\varepsilon}{2} \int_{\Omega} \nabla v \cdot \nabla v + \frac{1}{4} \int_{\Omega} v^4 - \int_{\Omega} f v \quad (6.4)$$

Thus we are in the framework of section 4, where PGD approximations are clearly defined by optimization problems.

We here compare two solution strategies. First, we will use a Newton algorithm, leading to a sequence of linear problems that are solved using a PGD algorithm (strategy of section 4.5). Secondly, we will construct a unique progressive PGD by computing successive corrections in \mathcal{R}_1 (with eventual additional updates) obtained by minimization problems in \mathcal{R}_1 (see section 5.1).

6.4.1. Newton algorithm and PGD method for linear problems. We denote by $F(u) = J'(u) \in V$ the (discrete) representant in V of the residual $-\varepsilon \Delta u + u^3 - f$. We apply the Newton algorithm for solving $F(u) = 0$. Given an initial guess $u_0 = 0$, we construct a sequence $\{u_k\}_{k \geq 0} \subset V$ defined by

$$F'(u_k)u_{k+1} = F'(u_k)u_k - F(u_k) \quad (6.5)$$

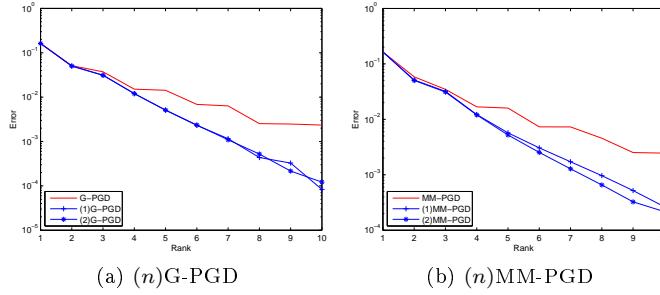


Fig. 6.17: Convergence of $(n)\mu$ -PGDs. Error in residual norm. $d = 2$.

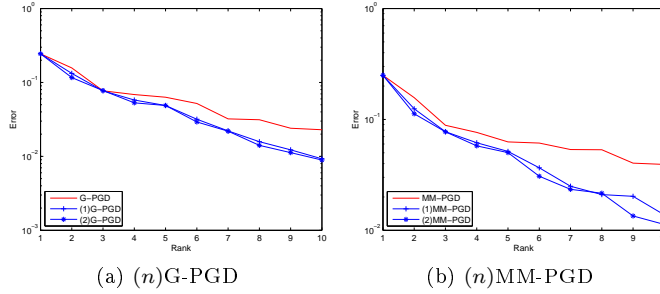


Fig. 6.18: Convergence of $(n)\mu$ -PGDs. Error in residual norm. $d = 3$.

that is u_{k+1} is the approximation in V of the solution of

$$\begin{cases} -\varepsilon\Delta u_{k+1} + 3u_k^2 u_{k+1} = f + 2u_k^3 & \text{in } \Omega \\ u_{k+1} = 0 & \text{on } \partial\Omega \end{cases} \quad (6.6)$$

At each iteration, we apply a PGD algorithm for the construction of a decomposition of u_{k+1} . The obtained decomposition have to be compared with the PGD construction which will be introduced in section 6.4.2. In the present non incremental version of the Newton algorithm, the precision required for the PGD algorithm will determinate the precision that can be expected on the final approximation¹¹. Equation (6.6) involves a symmetric and elliptic operator $-\varepsilon\Delta + 3u_k^2 I$ represented by $F'(u_k) = J''(u_k)$, and

$$u_{k+1} = \arg \min_{v \in V} G_k(v), \quad G_k(v) = \frac{1}{2} \langle J''(u_k)(v), v \rangle - \langle J''(u_k)(u_k) - J'(u_k), v \rangle$$

which is a convex optimization problem. A progressive PGD of u_{k+1} is then defined as described in section 4. It consists in constructing a sequence $\{u_{k+1}^m\}_{m \geq 0}$, with $u_{k+1}^0 = 0$, by computing successive corrections $w_m \in \mathcal{R}_1$ defined by

$$G_k(u_{k+1}^{m-1} + w_m) = \min_{w \in \mathcal{R}_1} G_k(u_{k+1}^{m-1} + w) \quad (6.7)$$

¹¹Note that another way to use Newton algorithm is to look for the increment $\delta_{k+1} = u_{k+1} - u_k$ as a solution of $F'(u_k)\delta_{k+1} = -F(u_k)$. This strategy will be used in the example of section 6.5.

and by letting u_{k+1}^m equal to $u_{k+1}^{m-1} + w_m$ or an update of this latter approximation.

REMARK 6.2. *Note that if an exact PGD were computed at each Newton iteration, then the sequence u_k would converge to the exact solution. Therefore, when u_k would approach u , we would obtain a PGD decomposition of u whose optimality is defined with the metric induced by the symmetric elliptic operator $-\varepsilon\Delta + 3u^2I$ (represented by $F'(u) = J''(u)$).*

Problem (6.7) is solved using an alternated minimization algorithm, described in section 5.1.3. Note that the resulting algorithm is no more than the G-PGD algorithm presented in section 5.3 applied to equation (6.5). We must remark that the operator $h \mapsto J''(u_k)(h)$ has to be computed once at each Newton iteration. Therefore, it remains constant during the G-PGD construction. As we will see, this is the main difference with the PGD construction presented in section 6.4.2, which will require much more actualizations of tangent operators.

In the following, we fix the decomposition rank of the G-PGD to $m = 10$ for each Newton iteration. The errors are measured in residual norm, both for Newton algorithm and G-PGD. We first consider the 2-dimensional case. Figure 6.19a shows the convergence of the Newton algorithm, while figure 6.19b gives the convergence of the G-PGD for each Newton iteration. We first observe a stagnation of the Newton iterates at a finite precision. It is explained by the fact that by imposing a maximal rank m for each PGD, the iterates of the Newton algorithm are approximated with a finite precision. On figure 6.19b, we observe that convergence properties of rank- m PGDs tend to stabilize with increasing Newton iterates. For the last Newton iterate $k = 5$ and for the maximum rank $m = 10$, we obtain an approximation u_k^m corresponding to a residual error of 10^{-7} . This error is related to the residual error of the Newton algorithm (up to a normalization factor of the residual).

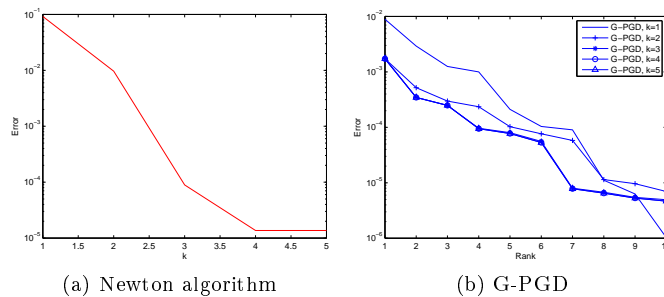


Fig. 6.19: Convergence of Newton algorithm (a) and of G-PGD for each Newton iteration (b). Error in residual norm. $d = 2$.

Figures 6.20a and 6.20b give the same convergence curves when an update is performed in the G-PGD construction. We observe the same qualitative behavior with a quantitative gain of two orders of magnitude. We can make similar comments for higher dimensional problems (see figures 6.21a and 6.21b for $d = 5$ and the updated G-PGD).

6.4.2. Construction of a unique progressive PGD. Now, we will directly construct a unique progressive PGD of the solution of (6.3), as described in section 4 for the case of convex optimization problems. It consists in constructing a sequence

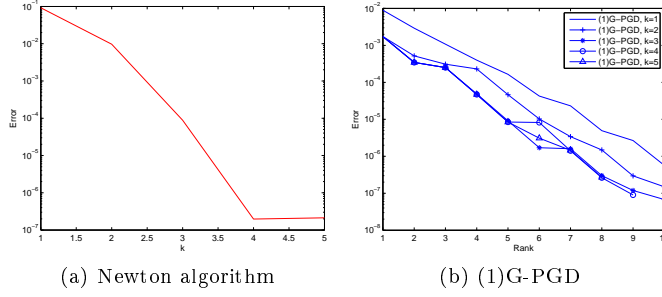


Fig. 6.20: Convergence of Newton algorithm (a) and of (1)G-PGD for each Newton iteration (b). Error in residual norm. $d = 2$.

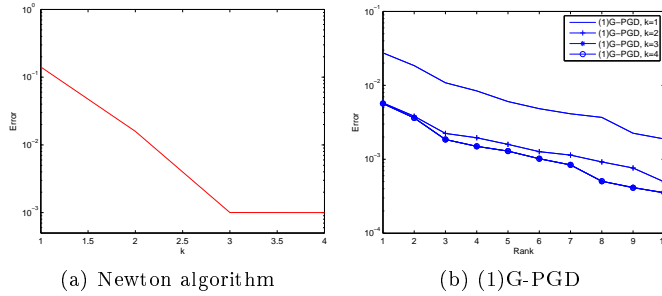


Fig. 6.21: Convergence of Newton algorithm (a) and of (1)G-PGD for each Newton iteration (b) for $d = 5$. Error in residual norm.

$\{u_m\}_{m \geq 0}$, with $u_0 = 0$, by computing successive corrections $w_m \in \mathcal{R}_1$ defined by

$$J(u_{m-1} + w_m) = \min_{w \in \mathcal{R}_1} J(u_{m-1} + w) \quad (6.8)$$

and by letting u_m equal to $u_{m-1} + w_m$ or an update of this latter approximation. This best approximation problem has been discussed in section 5.1. We recall that a necessary condition of optimality for $w_m \in \mathcal{R}_1$ reads

$$\langle J'(u_{m-1} + w_m), \delta w \rangle = 0 \quad \forall \delta w \in T_{w_m}(\mathcal{R}_1)$$

We use an alternated minimization algorithm for solving (6.8) (see section 5.1.3). At each iteration of the alternated algorithm associated with a dimension $j \in \{1, \dots, d\}$, for a current iterate $w_m \in \mathcal{R}_1$, we look for a new iterate $w_m \in \mathcal{R}_1^j(w_m)$ defined by

$$\langle J'(u_{m-1} + w_m), \delta w \rangle = 0 \quad \forall \delta w \in \mathcal{R}_1^j(w_m)$$

This equation is solved with a Newton algorithm, which consists in constructing a sequence $\{w_m^k\}_{k \geq 1}$ defined by

$$\langle J''(u_{m-1} + w_m^k)(w_m^{k+1} - w_m^k), \delta w \rangle = -\langle J'(u_{m-1} + w_m^k), \delta w \rangle \quad \forall \delta w \in \mathcal{R}_1^j(w) \quad (6.9)$$

REMARK 6.3. Note that contrary to equation (6.7), the tangent operator $h \mapsto J''(u_{m-1} + w_m^k)(h)$ must be recomputed for each direction of the alternated algorithm and for each Newton iteration. Therefore, the present progressive construction of a PGD is more computationally expensive than the construction of a succession of PGD approximations, as presented in section 6.4.1. However, in order to reduce computational costs, a quasi-Newton algorithm could be introduced, e.g. by replacing $J''(u_{m-1} + w_m^k)$ by $J''(u_{m-1})$.

For a given dimension d , we compare the convergence of the present PGD with that of the G-PGD of the last Newton iteration in the strategy presented in section 6.4.1 (see remark 6.2 for the interpretation of this PGD). When $d = 2$, we obtain relatively close convergence curves, as shown in figure 6.22a. But for $d \geq 3$, results are quite different, especially when the decomposition rank grows, as shown by figure 6.22b and 6.22c. Performing updates do not change this qualitative behavior (corresponding curves are not plotted here). As a conclusion, the present approach (progressive construction of a unique PGD) appears less efficient than the first approach (construction of a sequence of PGD approximations of the iterates of an iterative algorithm), in terms of precision and computational costs.

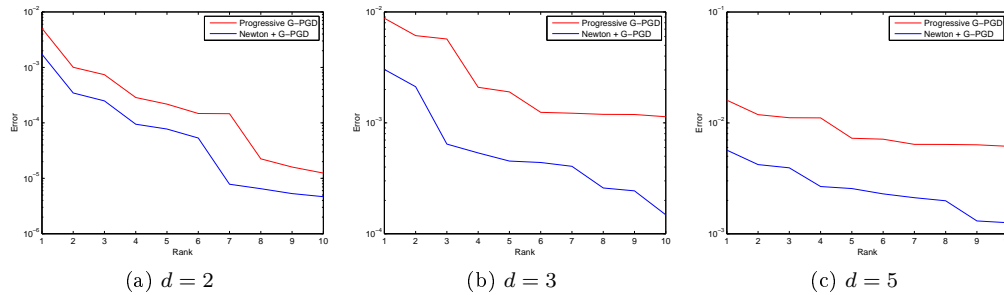


Fig. 6.22: Comparison between the PGD obtained with a progressive construction and the PGD obtained with a Newton algorithm coupled with PGD algorithm for approximating the Newton iterates (convergence of the PGD of the last Newton iterate). Error in residual norm.

6.5. Example 4 : a non linear and non convex problem. We consider the following non linear partial differential equation defined on a hyper-rectangular domain $\Omega = \Omega_1 \times \dots \times \Omega_d$:

$$\begin{cases} -\nabla \cdot (\varepsilon + u^2) \nabla u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \quad (6.10)$$

We choose $\varepsilon = 10^{-1}$. We use piecewise linear finite element approximation spaces $V_k \subset H_0^1(\Omega_k)$, with 1001 elements. We then work in a finite dimensional tensor space $V = V_1 \otimes \dots \otimes V_d \subset H_0^1(\Omega)$ with dimension 10^{3d} . This problem is not associated with the minimization of a convex functional, so that we have no guaranty for convergence of a progressive PGD as defined in section 5.3. Therefore, we only focus on the following numerical strategy. We apply a Newton algorithm for the solution of (6.10) and we use the G-PGD algorithm for the solution of the sequence of associated linear problems. Let us detail this strategy.

Here, we use an incremental version of the Newton algorithm. We denote by $F(u) \in V$ the (discrete) representant in V of the residual $-\nabla \cdot ((\varepsilon + u^2)\nabla u) - f$. Given an initial guess $u_0 = 0$, we construct a sequence $\{u_k\}_{k \geq 0} \subset V$ defined by $u_{k+1} = u_k + \delta_{k+1}$, with $\delta_{k+1} \in V$ such that

$$F'(u_k)\delta_{k+1} = -F(u_k) \tag{6.11}$$

that is $\delta_{k+1} \in V$ is the approximation of the solution of

$$\begin{cases} -\varepsilon \Delta \delta_{k+1} - 2\nabla \cdot (\delta_{k+1} u_k \nabla u_k) - \nabla \cdot (u_k^2 \nabla \delta_{k+1}) = -F(u_k) & \text{in } \Omega \\ u_{k+1} = 0 & \text{on } \partial\Omega \end{cases} \tag{6.12}$$

Note that equation (6.11) involves a nonsymmetric operator $F'(u_k)$. MM-PGD would be a priori a better choice than G-PGD in this context. However, for the present example, we observe similar convergence properties for G-PGD and MM-PGD. Therefore, we only consider the use of G-PGD: at each Newton iteration, we apply a G-PGD algorithm for constructing a rank- m approximation δ_{k+1}^m of δ_{k+1} . We here impose a fixed decomposition rank m . The approximation $u_{k+1} = u_k + \delta_{k+1}^m$ will be a rank- $(k+1)m$ tensor. Note that with this incremental procedure, even for a fixed rank m , if the resulting rank- m approximation δ_{k+1}^m is a good approximation of the exact increment δ_{k+1} , we can expect the convergence of the sequence u_k towards the exact solution u of the initial problem¹². This is illustrated below. In the following, we will impose the decomposition rank m and analyze the influence of m on the convergence of the global Newton algorithm.

REMARK 6.4. *Obviously, in terms of computational costs, there will be an interest in limiting the maximum rank m for the approximations δ_k^m of successive increments δ_k , or even in choosing an adaptive rank m_k so that the precision of $\delta_k^{m_k}$ is of the order of the error of the current Newton iterate.*

First, we analyze the 2-dimensional case. Figure 6.23a gives the convergence of Newton algorithm for different m , and figure 6.23b gives the convergence of the successive G-PGDs of increments δ_k for $m = 5$. We observe that, contrary to section 6.4.1, the Newton algorithm converges to the exact solution (no stagnation), with a convergence rate increasing with m (convergence rate of Newton algorithm is improved when increasing the precision of the approximations of increments). We note that the convergence rate of G-PGD decreases with k . This reflects the fact that the tensor decomposition of increments δ_k is more and more difficult as k increases (it can be understood by analyzing the spectral content of the δ_k).

Figure 6.24a and figure 6.24b illustrate the convergence of Newton algorithm and of the PGDs of successive increments, where a (1)G-PGD algorithm is used (1 update). Again, results are quite different from those of section 6.4.1. We observe a very small influence of updates, up to a rank $m = 5$. As a higher decomposition rank would be unnecessary in this incremental context, we can conclude that updates are not really useful for the present solution strategy. These comments remain valid for higher dimensional problems: in the 5-dimensional case, figures 6.25a and 6.25b illustrate the non updated version of G-PGD, while figures 6.26a and 6.26b illustrate the updated version of G-PGD.

7. Conclusion. We have presented in a general framework a class of tensor-based approximation methods called Proper Generalized Decompositions (PGD) for

¹²This is an important difference with the construction of section 6.4.1, where the approximations u_k were tensors of fixed rank m for all k , thus yielding a stagnation of the global Newton algorithm.

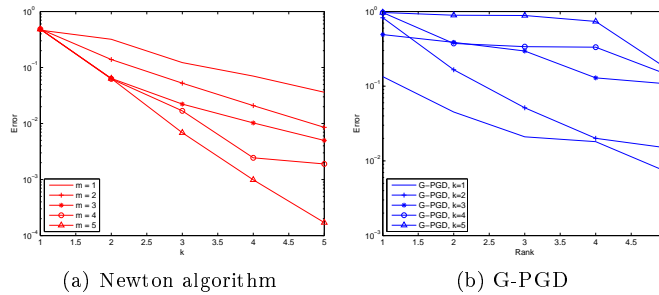


Fig. 6.23: Convergence of Newton algorithm using G-PGD solvers with different maximum ranks m (a) and convergence of the successive G-PGDs (with $m = 5$) for each Newton iteration (b). Error in residual norm. $d = 2$.

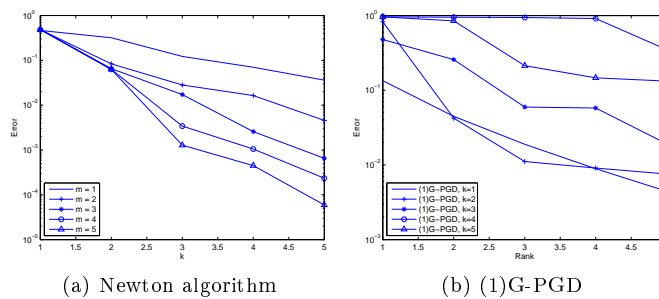


Fig. 6.24: Convergence of Newton algorithm using (1)G-PGD solvers with different maximum ranks m (a) and convergence of the successive (1)G-PGDs (with $m = 5$) for each Newton iteration (b). Error in residual norm. $d = 2$.

the a priori construction of a tensor decomposition of the solution of high-dimensional variational problems. We have introduced alternative definitions of tensor approximations, based on direct or progressive (greedy-type) constructions, the latter ones consisting in computing successive corrections in suitable tensor subsets. In the context of progressive constructions, we have introduced different updating strategies that uses the previously generated information and allows to significantly improve the quality of progressive decompositions. The prototypical case of convex optimization problems has served as a guideline in this general presentation of the PGD. Then, for the solution of more general variational problems, different definitions of progressive PGD have been introduced, corresponding to different definitions of corrections in tensor subsets. In the first one, MR-PGD, successive corrections are defined by a minimization of the residual norm. This definition is generally robust but it is computationally expensive and it may yield to poor convergence properties for bad choices of residual norms. In the second one, G-PGD, the successive corrections are defined using Galerkin projections. It presents a lack of robustness but it is less computationally expensive than minimal residual formulations. In the third one, MM-PGD, corrections are defined using Petrov-Galerkin projections. It tries to overcome the lack

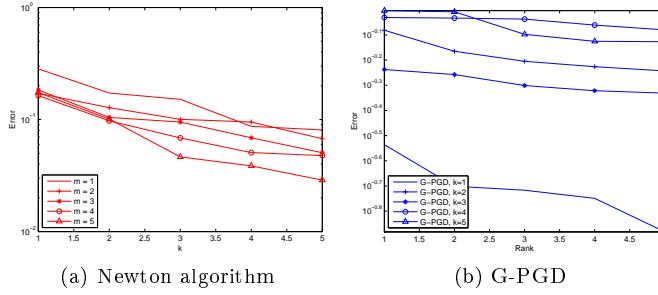


Fig. 6.25: Convergence of Newton algorithm using G-PGD solvers with different maximum ranks m (a) and convergence of the successive G-PGDs (with $m = 5$) for each Newton iteration (b). Error in residual norm. $d = 5$.

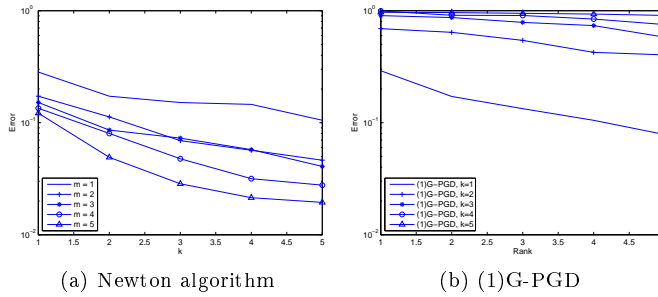


Fig. 6.26: Convergence of Newton algorithm using (1)G-PGD solvers with different maximum ranks m (a) and convergence of the successive (1)G-PGDs (with $m = 5$) for each Newton iteration (b). Error in residual norm. $d = 5$.

of robustness of G-PGD by improving convergence with respect to a specified metric. Several numerical examples have shown the interest (and sometimes the need) of performing updates while using progressive PGDs. We have also introduced a particular class of operators, called \mathcal{S} -tangent, for which we can ensure the monotonic decrease of some residual error for the G-PGD. In this context, G-PGD appears as robust as MR-PGD. \mathcal{S} -tangency covers a particular class of applications, and it especially appears as a natural property in the convergence analysis of G-PGD's, independently of classical properties of PDEs like ellipticity or symmetry. However, at that time, many questions remain open concerning this notion of \mathcal{S} -tangency, in particular concerning the characterization of \mathcal{S} -tangent bilinear forms for infinite dimensional tensor spaces.

We have also presented how the PGD can be coupled with classical iterative methods where it is used as a solver of successive linear problems. Compared to classical tensor approximation methods, the PGD allows the use of a larger class of iterative methods. For nonlinear problems, different solution strategies have been introduced and compared, that is strategies based on the construction of a unique PGD of the solution (with direct or progressive algorithms) or strategies based on the use of PGD as linear solvers in classical nonlinear iterative algorithms. Although

both strategies seem to yield nice convergence properties for the tested numerical applications, the latter one appears to be the most efficient from a computational point of view.

Acknowledgments. This work is supported by the French National Research Agency (grants ANR-2010-COSI-006 and ANR-2010-BLAN-0904) and by GNR MoMaS (ANDRA, BRGM, CEA, EdF, IRSN, PACEN-CNRS).

REFERENCES

- [1] A. AMMAR, F. CHINESTA, AND A. FALCO, *On the convergence of a greedy rank-one update algorithm for a class of linear systems*, Archives of Computational Methods in Engineering, 17 (2010), pp. 473–486.
- [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 modelling of complex fluids*, Journal of Non-Newtonian Fluid Mechanics, 139 (2006), pp. 153–176.
- [3] J. BALLANI AND L. GRASEDYCK, *A projection method to solve linear systems in tensor format*, Technical Report 309, RWTH Aachen, 2010.
- [4] D. S. BERNSTEIN, *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*, Princeton University Press, 2009.
- [5] C. LE BRIS, T. LELIEVRE, AND Y. MADAY, *Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations*, Constructive Approximation, 30 (2009), pp. 621–651.
- [6] E. CANCES, V. EHRLACHER, AND T. LELIEVRE, *Convergence of a greedy algorithm for high-dimensional convex nonlinear problems*, arXiv, arXiv (2010).
- [7] J. CHEN AND Y. SAAD, *On The Tensor SVD And The Optimal Low Rank Orthogonal Approximation Of Tensors*, Siam Journal On Matrix Analysis And Applications, 30 (2008), pp. 1709–1734.
- [8] F. CHINESTA, A. AMMAR, AND E. CUETO, *Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models*, Archives of Computational Methods in Engineering, 17 (2010), pp. 327–350.
- [9] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *On the best rank-1 and rank-(R_1, R_2, \dots, R_N) approximation of higher-order tensors*, Siam Journal On Matrix Analysis And Applications, 21 (2000), pp. 1324–1342.
- [10] 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 (2008), pp. 1084–1127.
- [11] A. FALCO, *Algorithms and numerical methods for high dimensional financial market models*, Revista de Economia Financiera, 20 (2010), pp. 51–68.
- [12] A. FALCO AND W. HACKBUSCH, *On minimal subspaces in tensor representations*, Foundations of Computational Mathematics, (submitted).
- [13] A. FALCO AND A. NOUY, *A Proper Generalized Decomposition for the solution of elliptic problems in abstract form by using a functional Eckart-Young approach*, Journal of Mathematical Analysis and Applications, 376 (2011), pp. 469–480.
- [14] ———, *Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces*, arXiv:1106.4424v1, ((submitted)).
- [15] L. FIGUEROA AND E. SÜLI, *Greedy approximation of high-dimensional ornstein-uhlenbeck operators with unbounded drift*, arXiv:1103.0726, (2011).
- [16] L. GRASEDYCK, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2029–2054.
- [17] W. HACKBUSCH, *Minimal subspaces*, in Working Seminar on Joint German-Russian DFG-RFBR project, MPI, Leipzig, May 4-5, 2010, 2010.
- [18] W. HACKBUSCH, B. KHOROMSKIJ, AND E. TYRTYSHNIKOV, *Approximate iterations for structured matrices*, Numerische Mathematik, 109 (2008), pp. 365–383. 10.1007/s00211-008-0143-0.
- [19] W. HACKBUSCH, B. N. KHOROMSKIJ, S. A. SAUTER, AND E. E. TYRTYSHNIKOV, *Use of tensor formats in elliptic eigenvalue problems*, Tech. Report Research Report No.2010-78, Max Planck Institute for Mathematics in the Sciences, 2008.
- [20] W. HACKBUSCH AND S. KUHN, *A New Scheme for the Tensor Representation*, JOURNAL OF FOURIER ANALYSIS AND APPLICATIONS, 15 (2009), pp. 706–722.

- [21] R. A. HORN AND C. R. JOHNSON, *Topics in matrix analysis*, Cambridge University Press, 1991.
- [22] T. ICHINOSE, *Spectral properties of tensor products of linear operators. i*, Transactions of the American Mathematical Society, 235 (1978), pp. 75–113. ArticleType: research-article / Full publication date: Jan., 1978 / Copyright © 1978 American Mathematical Society.
- [23] ———, *Spectral properties of tensor products of linear operators. II: the approximate point spectrum and kato essential spectrum*, Transactions of the American Mathematical Society, 237 (1978), pp. 223–254. ArticleType: research-article / Full publication date: Mar., 1978 / Copyright © 1978 American Mathematical Society.
- [24] B.N. KHOROMSKIJ AND C. SCHWAB, *Tensor-structured galerkin approximation of parametric and stochastic elliptic pdes*, Tech. Report Research Report No. 2010-04, ETH, 2010.
- [25] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.
- [26] C. S. KUBRUSLY AND N. LEVAN, *Preservation of tensor sum and tensor product*, Acta Mathematica Universitatis Comenianae, LXXX (2011), pp. 133–142.
- [27] P. LADEVÈZE, *Nonlinear Computational Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation*, Springer Verlag, 1999.
- [28] P. LADEVÈZE, J.C. PASSIEUX, AND D. NÉRON, *The LATIN multiscale computational method and the Proper Generalized Decomposition*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 1287–1296.
- [29] H. G. MATTHIES AND E. ZANDER, *Solving stochastic systems with low-rank tensor compression*, Linear Algebra and its Applications, pp. –.
- [30] A. NOUY, *A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 4521–4537.
- [31] ———, *Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 4718–4736.
- [32] ———, *A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 1603–1626.
- [33] ———, *Proper Generalized Decompositions and separated representations for the numerical solution of high dimensional stochastic problems*, Archives of Computational Methods in Engineering, 17 (2010), pp. 403–434.
- [34] A. NOUY AND O.P. LE MAÎTRE, *Generalized spectral decomposition method for stochastic non linear problems*, Journal of Computational Physics, 228 (2009), pp. 202–235.
- [35] I. OSELEDETS AND E. TYRTYSHNIKOV, *TT-cross approximation for multidimensional arrays*, Linear Algebra And Its Applications, 432 (2010), pp. 70–88.
- [36] V.N. TEMLYAKOV, *Greedy approximation*, Acta Numerica, 17 (2008), pp. 235–409.
- [37] A. USCHMAJEV, *Well-posedness of convex maximization problems on Stiefel manifolds and orthogonal tensor product approximations*, Numerische Mathematik, 115 (2010), pp. 309–331.
- [38] OSELEDETS I. V. AND TYRTYSHNIKOV E. E., *Breaking the curse of dimensionality, or how to use svd in many dimensions*, SIAM J. Sci. Comp., 31 (2009), pp. 3744–3759.