

To be or not to be intrusive? The solution of parametric and stochastic equations — Proper Generalized Decomposition*

Loïc Giraldi^a, Dishu Liu^b,
Hermann G. Matthies^c, and Anthony Nouy^{†a}

^aÉcole Centrale de Nantes, GeM UMR 6183, Nantes, France

^bInstitute of Aerodynamics and Flow Control, German Aerospace
Center (DLR), Brunswick, Germany

^cInstitute of Scientific Computing, Technische Universität Braunschweig,
Brunswick, Germany

30th August 2018

Abstract

A numerical method is proposed to compute a low-rank Galerkin approximation to the solution of a parametric or stochastic equation in a non-intrusive fashion. The considered nonlinear problems are associated with the minimization of a parameterized differentiable convex functional. We first introduce a bilinear parameterization of fixed-rank tensors and employ an alternating minimization scheme for computing the low-rank approximation. In keeping with the idea of non-intrusiveness, at each step of the algorithm the minimizations are carried out with a quasi-Newton method to avoid the computation of the Hessian. The algorithm is made non-intrusive through the use of numerical integration. It only requires the evaluation of residuals at specific parameter values. The algorithm is then applied to two numerical examples.

Keywords: parametric stochastic equation, Galerkin approximation, non-intrusive method, low-rank approximation, alternating minimization algorithm, quasi-Newton method, Proper Generalized Decomposition

Classification: 65K10, 65D30, 65M70, 15A69, 60H35

1 Introduction

We are interested in computing the solution of a stochastic parametric equation. In the literature, methods are said to be non-intrusive when they require simple calls to

*This work was partly supported by the Deutsche Forschungsgemeinschaft (DFG) and by the French National Research Agency (Grant ANR CHORUS MONU-0005)

[†]Corresponding author: Anthony.Nouy@ec-nantes.fr

the deterministic solver to compute samples of the solution. We can cite for instance approaches based on Monte-Carlo, collocation, or L^2 -projection methods [15]. On the other hand, Galerkin-type methods [12] are often considered as intrusive, as the Galerkin conditions lead to a coupled system of equations [8, 12] implying that the original software for the fixed parameter case can not be used and requires modification. However, in [8] it was shown that — in analogy to the *partitioned* solution of coupled problems — it is possible to solve the usual Galerkin equations non-intrusively in the parametric case by making use of the “deterministic” solver, i.e. the solver for a fixed value of the parameters. Recent methods to compute a low-rank approximation [4, 6] to such parametric or stochastic problems also lead to Galerkin-type procedures. Here we want to show that these methods too can be executed in a non-intrusive manner.

We want to represent the parametric solution $u(p)$ by an approximation of the form

$$u(p) \approx \sum_{i=1}^r \lambda_i(p) v_i,$$

where the $v_i \in \mathcal{U}$ are fixed vectors and the $\lambda_i(p)$ are real-valued functions of p , and hopefully the *rank* r is sufficiently small. An obvious advantage of such a decomposition is the reduction of the number of terms for the representation of the solution. We also hope to reduce the computational time for large parametric problems.

The Singular Value Decomposition (SVD) is the best known technique for constructing a low-rank approximation. If the solution u belongs to the tensor product of Hilbert spaces, the best low-rank approximation with respect to the canonical norm is the truncated SVD. Unfortunately, straightforward computation of the SVD requires to know the solution of the equation, and thus is not directly applicable.

An alternative is to use an iterative solver coupled with a low-rank approximation or truncation technique, leading to *approximate iterations*. These methods [9] have already been used [1, 11, 14] in linear problems, and could be extended to the iterative solver presented in [8] in a straightforward manner.

Another technique called Proper Generalized Decomposition (PGD) [4, 6] computes a low-rank approximation of the solution, relying on a Galerkin-type projection. We distinguish a progressive and a direct computation of the approximation. The first one consists in building the approximation in a greedy fashion, with the computation of a rank one approximation at each iteration, while the direct approach directly computes a fixed rank approximation in one go. Such fixed rank approximation can be computed with an alternating minimization algorithm in an optimization context [10].

Since the PGD relies on Galerkin-type projections, this method is usually classified as intrusive. In the present paper, it is shown that low-rank approximations can be computed in a non-intrusive fashion by just evaluating residuals. A low-rank approximation is found by alternating minimization of a convex functional. These minimizations are carried out with a quasi-Newton technique — we choose the quasi-Newton BFGS algorithm here [5, 12] — which avoids the computation of the Hessian. As a consequence, the proposed algorithm only requires evaluations of residuals — the negative gradient of the functional — to compute the low-rank approximation of the solution. The efficiency of the proposed approach is essentially related to the number of residual evaluations, which could be reduced by introducing structured approximations of the parameter-dependent residuals. The aim of the present paper is simply to show the feasibility of computing low-rank Galerkin approximation of the solution in a non-intrusive fashion, based on simple evaluations of the residual for some parameter values. Efficient implementations will be proposed in a future work.

The outline of the paper is as follows. In Section 2 the parametric problem is introduced with a special emphasis on the link between parametric-strong and parametric-

weak formulations. We give necessary conditions for these problems to be well-posed. Section 3 introduces the different ingredients for computing a low-rank approximation of the solution with a basic PGD method in a non-intrusive fashion via numerical integration and the use of a BFGS technique. An improved algorithm is presented in Section 4. In Section 5, the method is illustrated with two numerical examples.

2 Parametric problems

We consider the parametric problem of finding $u(p) \in \mathcal{U}$ such that

$$A(u(p); p) = b(p), \quad p \in \mathcal{P}, \quad (1)$$

where \mathcal{U} is a Hilbert space and \mathcal{P} is a parameter set equipped with a finite measure μ (e.g. probability measure), $A(\cdot; p) : \mathcal{U} \rightarrow \mathcal{U}$ and $b(p) \in \mathcal{U}$. We identify \mathcal{U} with its dual and we denote by $\langle \cdot, \cdot \rangle_{\mathcal{U}}$ the inner product on \mathcal{U} and $\|\cdot\|_{\mathcal{U}}$ the associated norm. A Galerkin approximation of the solution map could be computed in a non intrusive manner as in [8]. In this work, we are interested in finding a low-rank approximation of the solution. To do so, we assume that Problem (1) derives from the minimization of a functional $J(\cdot; p) : \mathcal{U} \ni v \mapsto J(v; p) \in \mathbb{R}$. A suitable framework is established with the following theorem.

Theorem 2.1. *Assume that*

- (a) *The map $v \mapsto J(v; p)$ is strongly convex uniformly in p , meaning that there exists a constant $\alpha > 0$ independent of p such that for all $v, w \in \mathcal{U}$ and for all $t \in [0, 1]$ we have*

$$J(tv + (1-t)w; p) \leq tJ(v; p) + (1-t)J(w; p) - \frac{\alpha}{2}t(1-t)\|v - w\|_{\mathcal{U}}^2,$$

- (b) *$v \mapsto J(v; p)$ is Fréchet differentiable with gradient*

$$\nabla J(v; p) = A(v; p) - b(p),$$

- (c) *$p \mapsto A(0; p) - b(p)$ is square μ -integrable,*

- (d) *$p \mapsto J(v; p)$ is μ -integrable and $v \mapsto A(v; p) - b(p)$ is Lipschitz uniformly in p on bounded sets, meaning that for all bounded sets $\mathcal{S} \subset \mathcal{U}$, there exists a constant $K > 0$ independent of p such that*

$$\|A(v; p) - A(w; p)\|_{\mathcal{U}} \leq K \|v - w\|_{\mathcal{U}}, \quad \forall v, w \in \mathcal{S}.$$

Then a solution of (1) exists and is unique for all parameters p such that we can define a solution map $u : \mathcal{P} \rightarrow \mathcal{U}$. Moreover, u is in $L^2(\mathcal{P}; \mathcal{U})$, u is the unique minimizer in $L^2(\mathcal{P}; \mathcal{U})$ of the functional

$$J_{\mathcal{P}} : u \mapsto \int_{\mathcal{P}} J(u(p); p) \mu(dp), \quad (2)$$

and is equivalently characterized by

$$\int_{\mathcal{P}} \langle A(u(p); p) - b(p), \delta u(p) \rangle_{\mathcal{U}} \mu(dp) = 0, \quad \forall \delta u \in L^2(\mathcal{P}; \mathcal{U}). \quad (3)$$

Proof. See Appendix A. □

It is additionally assumed [8] that an iterative solver for (1) is available,

$$u^{(k+1)}(p) \leftarrow u^{(k)}(p) + P^{-1}(R(u^{(k)}(p); p)),$$

convergent for all fixed values of p , where

$$R(u^{(k)}(p); p) := b(p) - A(u^{(k)}(p); p)$$

is the standard residual of (1). The linear map P is a preconditioner, which may depend on p and on the current iterate $u^{(k)}$; e.g. in Newton's method $P(u^{(k)}; p) = \nabla A(u^{(k)}; p)$ — the Fréchet derivative or gradient of $A(\cdot; p)$. Usually P is such that $P(u^{(k)}; p)(\Delta u^{(k)}) = R(u^{(k)}; p)$ is “easy to solve” for $\Delta u^{(k)} := u^{(k+1)} - u^{(k)}$. In any case, we assume that for all arguments p and u the map $v \mapsto P(v; p)$ is linear in Δu and non-singular. One should stress the fact that P or P^{-1} are never needed explicitly, only their action onto a vector. We assume that the software interface to the solver for (1) is such that one may access the residual $R(u; p)$ without any modification of the software, i.e. *non-intrusively*.

In a standard Galerkin method we introduce a finite dimensional space in $L^2(\mathcal{P}; \mathcal{U})$ where we look for an approximation of the solution map. Its non-intrusive computation was treated in [8]. Here we are interested in computing a low-rank approximation of the solution of the form

$$u \approx u_r = \sum_{i=1}^r \lambda_i \otimes v_i, \quad \lambda_i \in \mathcal{Q}, \quad v_i \in \mathcal{U}, \quad (4)$$

in a non-intrusive manner, where $\mathcal{Q} = L^2(\mathcal{P})$ is the space of square μ -integrable functions equipped with its natural inner product $\langle \cdot, \cdot \rangle_{\mathcal{Q}}$ and associated norm $\| \cdot \|_{\mathcal{Q}}$. The search for a low-rank approximation is justified by the tensor product structure of $L^2(\mathcal{P}; \mathcal{U})$ which is identified isomorphically with the tensor Hilbert space $\mathcal{Q} \otimes \mathcal{U}$ equipped with the induced canonical norm.

3 Basic Proper Generalized Decomposition

3.1 Computation of the approximation

The basic PGD technique consists in using a greedy rank one approximation [6] for computing an approximation of the solution of the form (4). Assume that we already have computed $u_r = \sum_{i=1}^r \lambda_i \otimes v_i$ and we want to find an approximation $u_r + \lambda \otimes v$ of the solution. The couple (λ, v) is computed by solving the minimization problem

$$\min_{(\lambda, v) \in \mathcal{Q} \times \mathcal{U}} J_{\mathcal{P}}(u_r + \lambda \otimes v).$$

The solution is computed using an alternating minimization algorithm. For λ fixed, v is computed solving the Euler-Lagrange equation related to the minimization with respect to v :

$$\int_{\mathcal{P}} \langle A(u_r(p) + \lambda(p)v; p) - b(p), \lambda(p)\delta v \rangle_{\mathcal{U}} \mu(dp) = 0, \quad \forall \delta v \in \mathcal{U},$$

or equivalently

$$\langle R_{\lambda}(v), \delta v \rangle_{\mathcal{U}} = 0, \quad \forall \delta v \in \mathcal{U}, \quad (5)$$

with $R_{\lambda}(v) = \int_{\mathcal{P}} (b(p) - A(u_r(p) + \lambda(p)v; p)) \lambda(p) \mu(dp)$. Similarly, for v fixed, the minimization on λ requires the solution of the nonlinear equation:

$$\int_{\mathcal{P}} \langle A(u_r(p) + \lambda(p)v; p) - b(p), \delta \lambda(p)v \rangle_{\mathcal{U}} \mu(dp) = 0, \quad \forall \delta \lambda \in \mathcal{Q},$$

or equivalently,

$$\langle R_v(\lambda), \delta\lambda \rangle_{\mathcal{Q}} = \int_{\mathcal{P}} R_v(\lambda)(p) \delta\lambda(p) \mu(dp) = 0, \quad \forall \delta\lambda \in \mathcal{Q}, \quad (6)$$

with $R_v(\lambda) : p \mapsto \langle b(p) - A(u_r(p) + \lambda(p)v; p), v \rangle_{\mathcal{U}}$. Problems (5) and (6) are well-defined since they correspond to the Euler-Lagrange equation related to the minimization of the strongly convex functionals $J_\lambda : v \mapsto J_{\mathcal{P}}(u_r + \lambda \otimes v)$ and $J_v : \lambda \mapsto J_{\mathcal{P}}(u_r + \lambda \otimes v)$ respectively. Moreover, the approximations $\{u_r\}_{r \in \mathbb{N}}$ are guaranteed to converge to the solution (see [3]).

The equations (5) and (6) can be solved by any suitable method. For example, Newton's method for (5) is iterating $v_{k+1} \leftarrow v_k - [\nabla R_\lambda(v_k)]^{-1} R_\lambda(v_k)$ until convergence, and correspondingly for (6): $\lambda_{k+1} \leftarrow \lambda_k - [\nabla R_v(\lambda_k)]^{-1} R_v(\lambda_k)$. Newton's method can be seen as a prototype algorithm for solving (5) and (6). The basic PGD algorithm is summarized in Algorithm 1.

Algorithm 1 Basic PGD

```

Initialization  $u_0$ .
 $r \leftarrow 0$ 
while no convergence of  $u_r$  do
  Initialize  $v, \lambda$ 
  while no convergence of  $\lambda \otimes v$  do
     $\lambda \leftarrow \lambda / \|\lambda\|_{\mathcal{Q}}$ 
    Solve Equation (5) for  $v$ 
     $v \leftarrow v / \|v\|_{\mathcal{U}}$ 
    Solve Equation (6) for  $\lambda$ 
  end while
   $u_{r+1} \leftarrow u_r + \lambda \otimes v$ 
   $r \leftarrow r + 1$ 
end while

```

3.2 Non-intrusive implementation

3.2.1 Computation of the projected residuals

To drive to zero the residuals in $R_\lambda(v) = -\nabla J_\lambda(v)$ in (5) and $R_v(\lambda) = -\nabla J_v(\lambda)$ in (6) in the basic PGD Algorithm 1 (e.g. by Newton's method as indicated above in the solutions steps in Algorithm 1), those residuals have to be evaluated. The non-intrusive evaluation will only use the usual residual $R(v; p)$ of (1) as introduced in Section 2.

Let $\{w_z\}$ and $\{p_z\}$ be the weights and points associated with a quadrature formula on \mathcal{P} for the measure μ . The residual $R_\lambda(v)$ in expression (5) then becomes

$$R_\lambda(v) = \int_{\mathcal{P}} \lambda(p) R(u_r(p) + \lambda(p)v; p) \mu(dp) \approx \sum_z w_z \lambda(p_z) R(u_r(p_z) + \lambda(p_z)v; p_z).$$

Similarly, the expression in (6) becomes for all $\delta\lambda \in \mathcal{Q}$

$$\begin{aligned} \langle R_v(\lambda), \delta\lambda \rangle_{\mathcal{Q}} &= \int_{\mathcal{P}} \langle R(u_r(p) + \lambda(p)v; p), v \rangle_{\mathcal{U}} \delta\lambda(p) \mu(dp) \\ &\approx \sum_z w_z \langle R(u_r(p_z) + \lambda(p_z)v; p_z), v \rangle_{\mathcal{U}} \delta\lambda(p_z). \end{aligned}$$

One may observe from these relations that the computation of the residuals in (5) and (6) requires only the evaluation of standard residuals at the quadrature points p_z of the parametric space with state vector $u_r(p_z) + \lambda(p_z)v$, that is

$$R(u_r(p_z) + \lambda(p_z)v; p_z) = b(p_z) - A(u_r(p_z) + \lambda(p_z)v; p_z).$$

3.2.2 Introduction of a quasi-Newton method

If one were to use Newton’s method for solving (5) resp. (6) in Algorithm 1, one would not only have to evaluate residuals, but one would also have to evaluate the Hessians of the functionals (gradients of the residuals). The first Hessian is equal to

$$\nabla^2 J_\lambda(v) = -\nabla R_\lambda(v) = \int_{\mathcal{P}} \lambda(p)^2 \nabla A(u_r(p) + \lambda(p)v; p) \mu(dp) \quad (7)$$

and could be called a “weighted tangent matrix”, and for the other residual we have for all $\delta\lambda_1, \delta\lambda_2 \in \mathcal{Q}$:

$$\begin{aligned} \nabla^2 J_v(\lambda)(\delta\lambda_1, \delta\lambda_2) &= -\langle \nabla R_v(\lambda) \delta\lambda_1, \delta\lambda_2 \rangle_{\mathcal{Q}} \\ &:= \int_{\mathcal{P}} \langle \nabla A(u_r(p) + \lambda(p)v; p) v, v \rangle_{\mathcal{U}} \delta\lambda_1(p) \delta\lambda_2(p) \mu(dp). \end{aligned} \quad (8)$$

Again this would mean accessing the “tangent matrix” $\nabla A(\cdot; p)$, and hence Newton’s method cannot be really carried out non-intrusively. We therefore propose to use a quasi-Newton method, which only requires evaluation of residuals. This can be done in a non-intrusive fashion as demonstrated in Subsection 3.2.1.

In the following, the symbol x can stand for λ (resp. v), y for v (resp. λ) and \mathcal{X} for the space \mathcal{Q} (resp. \mathcal{U}). The inner product on \mathcal{X} is denoted by $\langle \cdot, \cdot \rangle_{\mathcal{X}}$. A quasi-Newton method [5] defines the iterations by

$$x^{(\ell+1)} = x^{(\ell)} + \rho_\ell C_\ell R_y(x^{(\ell)}),$$

where ρ_ℓ is a scalar factor to be defined through a linesearch procedure to be described later, and C_ℓ is an approximation of the inverse of the negative gradient of $R_y(x^{(\ell)})$ computed with the different iterates of the algorithm such that the so called quasi-Newton equation $C_{\ell+1} z_\ell = t_\ell$ is satisfied — see below for z_ℓ and t_ℓ — and the correction of C_ℓ in each iteration is of low rank.

Given that we are minimizing a functional, we use here a BFGS method [13] where at iteration $\ell + 1$, $C_{\ell+1}$ is defined recursively by

$$C_{\ell+1} = C_\ell + \frac{\langle z_\ell, t_\ell \rangle_{\mathcal{X}} + \langle z_\ell, s_\ell \rangle_{\mathcal{X}}}{\langle z_\ell, t_\ell \rangle_{\mathcal{X}}^2} (t_\ell \otimes t_\ell) - \frac{1}{\langle z_\ell, t_\ell \rangle_{\mathcal{X}}} (s_\ell \otimes t_\ell + t_\ell \otimes s_\ell). \quad (9)$$

where $z_\ell = -(R_y(x^{(\ell+1)}) - R_y(x^{(\ell)}))$, $t_\ell = x^{(\ell+1)} - x^{(\ell)}$, $s_\ell = C_\ell z_\ell$ and C_0 is taken as the formal inverse of a convenient preconditioner to be defined later.

It should be noted that the algorithm can be performed in a ‘matrix-free’ formulation, as the matrices C_ℓ are only needed through their action on a vector. Hence they have not to be stored explicitly [13]. The application of C_ℓ to a vector is described recursively by (9), the action of a typical term, e.g. $s_\ell \otimes t_\ell$, on a vector x being given by $\langle t_\ell, x \rangle_{\mathcal{X}} s_\ell$. The choice of C_0 will be described later. In that way only the vectors t_ℓ and s_ℓ plus the scalar factors have to be stored for each update. The application of C_ℓ to a vector thus needs two inner products and a linear combination of three vectors per update. Most often, BFGS is used in a *limited memory* form [13], with the number of updates limited

to L . Once the counter reaches $\ell \geq L$, either all updates are ‘forgotten’ — a restart — or the vectors t_ℓ and s_ℓ plus scalar factors are put in a queue of length L , and when the queue is full the first update is popped out and the last one enqueued; for details see [13].

With the notations $J_v : \lambda \mapsto J_{\mathcal{P}}(u_r + \lambda \otimes v)$ and $J_\lambda : v \mapsto J_{\mathcal{P}}(u_r + \lambda \otimes v)$, this yields the Algorithm 2 for computing the solution of $R_y(x) = 0$.

Algorithm 2 BFGS for computing the solution of $R_y(x) = 0$

Initialization of $x^{(0)}$

$C_0 \leftarrow P_y^{-1}$ (symbolically, inverse of preconditioner P_y)

$\ell \leftarrow 0$

$d_0 \leftarrow C_\ell R_y(x^{(0)})$

while *no convergence* **do**

$\rho_\ell \leftarrow$ coarse root of $\rho \mapsto \sigma(\rho)$

$t_\ell \leftarrow \rho_\ell d_\ell$

$x^{(\ell+1)} \leftarrow x^{(\ell)} + t_\ell$

$d_{\ell+1} \leftarrow C_\ell R_y(x^{(\ell+1)})$

$s_\ell \leftarrow d_\ell - d_{\ell+1}$

 Store update information t_ℓ , s_ℓ and scalar factors

$\ell \leftarrow \ell + 1$

end while

One should bear in mind that this algorithm relies on evaluations of $R_y(x^{(\ell)})$, that is evaluations of standard residuals according to Section 3.2.1, which makes this algorithm non-intrusive. The scalar ρ_ℓ is computed with a coarse linesearch, which picks ρ_ℓ such as to minimize $\rho \mapsto \zeta(\rho) := J_y(x^{(\ell)} + \rho d_\ell)$. At the minimum we will have $\sigma(\rho) := d\zeta(\rho)/d\rho = 0$, which means $\sigma(\rho) = \langle d_\ell, R_y(x^{(\ell)} + \rho d_\ell) \rangle_{\mathcal{X}} = 0$. The linesearch can thus be carried out by finding a zero or root of the one-dimensional equation $\sigma(\rho) = 0$, which involves only evaluation of residuals and hence can be performed non-intrusively. In [13] a variant of *regula falsi* was used for this. The linesearch can be very coarse, it is in effect an ‘insurance policy’ to avoid divergence in early iterations. It can be used with Newton’s method to increase the domain of convergence. One may show (see [5, 13] and the references therein) that as the method converges, one may choose $\rho_\ell = 1$ so that the linesearch does not have to be carried out later in the iteration; for details see [13]. Given that $R_y(x)$ can be evaluated in a non-intrusive fashion with numerical integration, the whole technique is non-intrusive. The BFGS method is summarized in Algorithm 2, and the non-intrusive implementation of the basic PGD method now uses the BFGS algorithm as described in Algorithm 2 for the two tasks:

- Solve equation (5) for v .
- Solve equation (6) for λ .

It remains to specify the matrix $C_0 = P_y^{-1}$. The matrix is only needed when applied to a vector, thus P_y^{-1} is not needed explicitly. The preconditioner is best if it is a good approximation of the Hessian $\nabla^2 J_y$. Relation (7) suggest some very simple choices, e.g. for $x = v$ and $y = \lambda$ when we solve for v (solving equation (5)), we may use the original ‘deterministic’ preconditioner $P = P(v; p)$ described in Section 2 to obtain an approximation for $\nabla^2 J_\lambda(v)$. As λ is normalized, a very crude approximation is $P_\lambda := P(u_r(p_a); p_a)$, where $p_a \in \mathcal{P}$ is a (possibly well chosen) sample. This way the preconditioner is accessible in a non-intrusive fashion.

Remark 3.1. Another possibility is to directly replace $C_0 R_y(x^{(\ell)}) \leftrightarrow C_0 R_\lambda(v^{(\ell)})$ (the only context where C_0 is needed) by

$$P_\lambda^{-1} R_\lambda(v^{(\ell)}) = C_0 R_\lambda(v^{(\ell)}) := \sum_z w_z (\lambda^{(\ell)}(p_z))^2 P^{-1}(u_r(p_z) + \lambda^{(\ell)}(p_z) v^{(\ell)}; p_z) \left(R(u_r(p_z) + \lambda^{(\ell)}(p_z) v^{(\ell)}; p_z) \right),$$

where each evaluation at a sampling point p_z corresponds to one ‘iteration’ of the original deterministic system, a non-intrusive computation.

On the other hand for $x = \lambda$ and $y = v$ when we solve for λ (solving equation (6)), we see from Equation (8) that the action of $\nabla^2 J_v(\lambda)$ is fully diagonalized, it is multiplication by the positive scalar function $p \mapsto \langle \nabla A(u_r(p) + \lambda(p)v; p) v, v \rangle_{\mathcal{U}}$. A very simple choice is replacing that function by a constant $P_v \in \mathbb{R}_+$ which one may take — as v is normalised — inside the convex hull of the spectra of the symmetric positive definite operators $\nabla A(v; p)$, a crude approximation is the constant function

$$p \mapsto P_v := \langle P(u_r(p_a) + \lambda(p_a)v; p_a)v, v \rangle_{\mathcal{U}} \approx \langle \nabla A(u_r(p_a) + \lambda(p_a)v; p_a) v, v \rangle_{\mathcal{U}} > 0,$$

where $p_a \in \mathcal{P}$ is again a random (or well chosen) element. The application to a function $\lambda \in \mathcal{Q}$ is then the multiplication by the constant $P_v^{-1} \in \mathbb{R}_+$ defined by

$$P_v^{-1} = (\langle P(u_r(p_a); p_a)v, v \rangle_{\mathcal{U}})^{-1}.$$

That is certainly a non-intrusive computation.

Remark 3.2. Another possibility is to directly replace $C_0 R_y(x^{(\ell)}) = C_0 R_v(\lambda^{(\ell)})(p)$ (the only context where C_0 is needed) for each p_z of the integration rule (the only points in \mathcal{P} where it is needed) by

$$P_v^{-1} R_v(\lambda^{(\ell)})(p_z) = C_0 R_v(\lambda^{(\ell)})(p_z) := \langle P^{-1}(u_r(p_z) + \lambda^{(\ell)}(p_z) v^{(\ell)}; p_z) \left(R(u_r(p_z) + \lambda^{(\ell)}(p_z) v^{(\ell)}; p_z) \right), v^{(\ell)} \rangle_{\mathcal{U}},$$

where each evaluation at a sampling point p_z corresponds to one ‘iteration’ of the original deterministic system at parameter value p_z with starting point $u_r(p_z) + \lambda^{(\ell)}(p_z) v^{(\ell)}$, that is a non-intrusive computation.

4 Improved PGD algorithm

In the following, we consider that the cost of the evaluations of $\{u_r(p_z)\}$ is negligible compared to the cost of the evaluations of $\{A(u_r(p_z); p_z) - b(p_z)\}$. This hypothesis suggests that the cost of the optimization of all the $(\lambda_i)_{i=1}^r$, or of all the $(v_i)_{i=1}^r$, should be almost independent of the rank r . We thus propose an improved strategy for computing an approximation of the solution.

4.1 Low-rank approximation of the solution

The set of canonical tensors \mathcal{C}_r of rank at most r , defined by

$$\mathcal{C}_r = \left\{ \sum_{i=1}^r \lambda_i \otimes v_i; \lambda_i \in \mathcal{Q}, v_i \in \mathcal{U} \right\} \subset \mathcal{Q} \otimes \mathcal{U},$$

is weakly closed, and the best approximation of a tensor in \mathcal{C}_r with respect to the canonical norm is given by the truncated singular value decomposition (SVD).

A direct low-rank approximation $u_r \in \mathcal{C}_r$ of the solution u is defined by

$$\min_{v \in \mathcal{C}_r} J_{\mathcal{P}}(v), \quad \text{with} \quad J_{\mathcal{P}}(v) = \int_{\mathcal{P}} J(v(p); p) \mu(dp), \quad (10)$$

where J is defined in Theorem 2.1.

The set \mathcal{C}_r is not a vector space, nor a convex set, so that the computation of the solution to (10) requires specific algorithms. We introduce a parameterization $F_r : \mathcal{Q}^r \times \mathcal{U}^r \rightarrow \mathcal{Q} \otimes \mathcal{U}$ such that $F_r(\mathcal{Q}^r, \mathcal{U}^r) = \mathcal{C}_r$. Let $\boldsymbol{\lambda} = (\lambda_i)_{1 \leq i \leq r} \in \mathcal{Q}^r$ and $\mathbf{v} = (v_i)_{1 \leq i \leq r} \in \mathcal{U}^r$. The map F_r is defined by

$$F_r(\boldsymbol{\lambda}, \mathbf{v}) = \sum_{i=1}^r \lambda_i \otimes v_i. \quad (11)$$

Thanks to this parameterization, the problem (10) consists in solving

$$\min_{\boldsymbol{\lambda} \in \mathcal{Q}^r, \mathbf{v} \in \mathcal{U}^r} J_{\mathcal{P}} \circ F_r(\boldsymbol{\lambda}, \mathbf{v}), \quad (12)$$

Lemma 4.1. *The map F_r is bilinear and continuous, such that F_r and its partial maps are Fréchet differentiable.*

Proof. The continuity of F_r comes from the continuity of the tensor product $\otimes : \mathcal{Q} \times \mathcal{U} \mapsto \mathcal{Q} \otimes \mathcal{U}$ with respect to the norm $\|\cdot\|_{\mathcal{Q} \otimes \mathcal{U}}$. As a consequence, F_r and its partial maps are Fréchet differentiable. \square

Remark 4.2. *The representation (11) is not unique. For $T \in GL_r(\mathbb{R})$, we denote $T\boldsymbol{\lambda} = \{\sum_{j=1}^r T_{ij}\lambda_j\}_{i=1}^r \in \mathcal{Q}^r$ and $T\mathbf{v} = \{\sum_{j=1}^r T_{ij}v_j\}_{i=1}^r \in \mathcal{U}^r$. For all $(\boldsymbol{\lambda}, \mathbf{v}, T) \in \mathcal{Q}^r \times \mathcal{U}^r \times GL_r(\mathbb{R})$, we have $F_r(\boldsymbol{\lambda}, \mathbf{v}) = F_r(T\boldsymbol{\lambda}, T^{-1}\mathbf{v})$. The principal consequence is that there exists an infinite number of solutions to the problem (10), and that these solutions are not isolated. Hence, we can not directly apply a Newton method since the Hessian will become ill-conditioned near a critical point.*

4.2 Adaptive alternating minimization algorithm

We solve the problem (12) with an alternating minimization algorithm, which means that we alternatively solve the problems

$$\min_{\mathbf{v} \in \mathcal{U}^r} \int_{\mathcal{P}} J(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) \mu(dp) \quad \text{and} \quad \min_{\boldsymbol{\lambda} \in \mathcal{Q}^r} \int_{\mathcal{P}} J(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) \mu(dp)$$

until convergence of $F_r(\boldsymbol{\lambda}, \mathbf{v})$. The existence and the characterization of the solutions of these problems are given in the following theorem.

Theorem 4.3. *Under the assumptions of Theorem 2.1, if $\boldsymbol{\lambda}$ is a set of linearly independent functions, there exists a unique solution $\mathbf{v} \in \mathcal{U}^r$ to the minimization problem*

$$\min_{\mathbf{v} \in \mathcal{U}^r} \int_{\mathcal{P}} J(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) \mu(dp),$$

characterized by the equation

$$\int_{\mathcal{P}} \langle R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p), F_r(\boldsymbol{\lambda}, \delta\mathbf{v})(p) \rangle_{\mathcal{U}} \mu(dp) = 0, \quad \forall \delta\mathbf{v} \in \mathcal{U}^r. \quad (13)$$

Similarly, if \mathbf{v} is a set of linearly independent vectors, there exists a unique solution $\boldsymbol{\lambda} \in \mathcal{Q}^r$ to the minimization problem

$$\min_{\boldsymbol{\lambda} \in \mathcal{Q}^r} \int_{\mathcal{P}} J(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) \mu(dp),$$

characterized by the equation

$$\int_{\mathcal{P}} \langle R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p), F_r(\delta \boldsymbol{\lambda}, \mathbf{v})(p) \rangle_{\mathcal{U}} \mu(dp) = 0, \quad \forall \delta \boldsymbol{\lambda} \in \mathcal{Q}^r. \quad (14)$$

Proof. See Appendix B. □

We equip the product space \mathcal{U}^r with the natural inner product $\langle \cdot, \cdot \rangle_{\mathcal{U}^r}$ defined by

$$\langle \mathbf{w}, \mathbf{v} \rangle_{\mathcal{U}^r} = \sum_{i=1}^r \langle w_i, v_i \rangle_{\mathcal{U}}, \quad \forall \mathbf{w} = (w_i)_{i=1}^r \in \mathcal{U}^r, \quad \forall \mathbf{v} = (v_i)_{i=1}^r \in \mathcal{U}^r,$$

and we equip the product space \mathcal{Q}^r with the natural inner product $\langle \cdot, \cdot \rangle_{\mathcal{Q}^r}$ defined by

$$\langle \boldsymbol{\lambda}, \boldsymbol{\gamma} \rangle_{\mathcal{Q}^r} = \sum_{i=1}^r \langle \lambda_i, \gamma_i \rangle_{\mathcal{Q}}, \quad \forall \boldsymbol{\lambda} = (\lambda_i)_{i=1}^r \in \mathcal{Q}^r, \quad \forall \boldsymbol{\gamma} = (\gamma_i)_{i=1}^r \in \mathcal{Q}^r.$$

We have to solve the nonlinear Equations (13) and (14) that are related to the minimization of some functionals. Given that

$$\begin{aligned} & \int_{\mathcal{P}} \langle R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p), F_r(\boldsymbol{\lambda}, \delta \mathbf{v})(p) \rangle_{\mathcal{U}} \mu(dp) \\ &= \sum_{i=1}^r \left\langle \int_{\mathcal{P}} R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) \lambda_i(p) \mu(dp), \delta v_i \right\rangle_{\mathcal{U}} = \langle R_{\boldsymbol{\lambda}}(\mathbf{v}), \delta \mathbf{v} \rangle_{\mathcal{U}^r}, \end{aligned}$$

finding the solution to Equation (13) is equivalent to finding \mathbf{v} , solution to

$$\langle R_{\boldsymbol{\lambda}}(\mathbf{v}), \delta \mathbf{v} \rangle_{\mathcal{U}^r} = 0, \quad \forall \delta \mathbf{v} \in \mathcal{U}^r, \quad (15)$$

with $R_{\boldsymbol{\lambda}}(\mathbf{v}) = (R_{\lambda_i}(\mathbf{v}))_{i=1}^r \in \mathcal{U}^r$ and

$$R_{\lambda_i}(\mathbf{v}) = \int_{\mathcal{P}} R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) \lambda_i(p) \mu(dp), \quad i \in \{1, \dots, r\}.$$

It is only necessary to compute $R_{\lambda_i}(\mathbf{v})(p)$ at the integration points p_z . Once the residuum $R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p_z); p_z)$ has been evaluated it can be used for all $i \in \{1, \dots, r\}$.

Similarly, using that

$$\begin{aligned} & \int_{\mathcal{P}} \langle R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p), F_r(\delta \boldsymbol{\lambda}, \mathbf{v})(p) \rangle_{\mathcal{U}} \mu(dp) \\ &= \sum_{i=1}^r \int_{\mathcal{P}} \langle R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p), v_i \rangle_{\mathcal{U}} \delta \lambda_i(p) \mu(dp) = \langle R_{\mathbf{v}}(\boldsymbol{\lambda}), \delta \boldsymbol{\lambda} \rangle_{\mathcal{Q}^r}, \end{aligned}$$

finding the solution to Equation (14) is equivalent to finding $\boldsymbol{\lambda}$, solution to

$$\langle R_{\mathbf{v}}(\boldsymbol{\lambda}), \delta \boldsymbol{\lambda} \rangle_{\mathcal{Q}^r} = 0, \quad \forall \delta \boldsymbol{\lambda} \in \mathcal{Q}^r, \quad (16)$$

with $R_{\mathbf{v}}(\boldsymbol{\lambda}) = (R_{v_i}(\boldsymbol{\lambda}))_{i=1}^r \in \mathcal{Q}^r$ and

$$R_{v_i}(\boldsymbol{\lambda}) : p \mapsto \langle R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p), v_i \rangle_{\mathcal{U}}, \quad i \in \{1, \dots, r\},$$

approximated again by standard residuum evaluations; and $R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p_z); p_z)$ has to be evaluated only once for all $i \in \{1, \dots, r\}$.

We can thus use once again a BFGS method to solve Problems (15) and (16). Moreover, the algorithm can be made non-intrusive using numerical integration given that

$$\langle R_{v_i}(\boldsymbol{\lambda}), \delta \lambda_i \rangle_{\mathcal{Q}} \approx \sum_z w_z \langle R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p_z); p_z), v_i \delta \lambda_i(p_z) \rangle_{\mathcal{U}}$$

and

$$R_{\lambda_i}(\mathbf{v}) \approx \sum_z w_z R(F_r(\boldsymbol{\lambda}, \mathbf{v})(p_z); p_z) \lambda_i(p_z).$$

With $u_r = F_r(\boldsymbol{\lambda}, \mathbf{v})$, we insist on the fact that a BFGS technique will require the evaluations of the residual $R(u_r(p_z); p_z) = b(p_z) - A(u_r(p_z); p_z)$.

In order to avoid any degeneracy and obtain well-conditioned problems, we introduce two orthogonalization steps. We denote by $orth : \mathcal{X}^r \rightarrow \mathcal{X}^r$, $\mathcal{X} = \mathcal{Q}$ or \mathcal{U} , an operator such that with $\mathbf{x}' = orth(\mathbf{x})$, we have $\text{span } \mathbf{x} \subset \text{span } \mathbf{x}'$, and $\mathbf{x}' = (x'_i)_{i=1}^r$ is an orthonormal set. Such a set can be obtained by taking the first r left singular vectors of \mathbf{x} considered as a tensor in $\mathcal{X} \otimes \mathbb{R}^r$ for instance.

Finally, the rank is adapted by choosing a good initial guess at each step. Except for the rank one approximation, the initial guess for the computation of the rank r approximation is chosen to be the rank $r - 1$ approximation of the solution computed at the previous iteration plus a rank one term. The whole approach is summarized in Algorithm 3.

Algorithm 3 Non-intrusive implementation of the improved PGD

```

Initialization of  $u_0$ .
 $r \leftarrow 1$ 
while no convergence of  $u_r$  do
  Initialize  $v_r, \lambda_r$ 
   $\boldsymbol{\lambda} \leftarrow (\lambda_i)_{i=1}^r$ 
   $\mathbf{v} \leftarrow (v_i)_{i=1}^r$ 
  while no convergence of  $F_r(\boldsymbol{\lambda}, \mathbf{v})$  do
     $\boldsymbol{\lambda} \leftarrow orth(\boldsymbol{\lambda})$ 
    Solve Equation (15) using Algorithm 2
     $\mathbf{v} \leftarrow orth(\mathbf{v})$ 
    Solve Equation (16) using Algorithm 2
  end while
   $u_r \leftarrow F_r(\boldsymbol{\lambda}, \mathbf{v})$ 
   $r \leftarrow r + 1$ 
end while

```

The operator C_0 has now to be defined. Again, we propose a priori good approximations C_0^{-1} of the Hessian of the functional in order to improve the performance of the BFGS method. We observe that the Hessian $H_{\boldsymbol{\lambda}}(\mathbf{v})$ of $J_{\mathcal{P}} \circ F_r(\boldsymbol{\lambda}, \mathbf{v})$ where the derivative is taken with respect to \mathbf{v} is

$$\langle H_{\boldsymbol{\lambda}}(\mathbf{v}) \delta \mathbf{v}, \delta \mathbf{v}' \rangle_{\mathcal{U}^r} = \sum_{i=1}^r \sum_{j=1}^r \left\langle \left(\int_{\mathcal{P}} \lambda_i(p) \lambda_j(p) \nabla A(u_r(p); p) \mu(dp) \right) \delta v_i, \delta v'_j \right\rangle_{\mathcal{U}}.$$

Given that $\boldsymbol{\lambda}$ is a family of orthonormal vectors, this suggests that C_0^{-1} could be approximated by a block-diagonal version of the preconditioner proposed in Section 3.2.2, each block being defined by $P(u_r(p_a); p_a)$.

Remark 4.4. Similarly, the Hessian $H_v(\boldsymbol{\lambda})$ of $J_{\mathcal{P}} \circ F_r(\boldsymbol{\lambda}, \mathbf{v})$ where the derivative is taken with respect to $\boldsymbol{\lambda}$ is

$$\langle H_v(\boldsymbol{\lambda})\delta\boldsymbol{\lambda}, \delta\boldsymbol{\lambda}' \rangle_{\mathcal{Q}^r} = \sum_{i=1}^r \sum_{j=1}^r \int_{\mathcal{P}} \langle \nabla A(u_r(p); p)v_i, v_j \rangle_{\mathcal{U}} \delta\lambda_i(p) \delta\lambda'_j \mu(dp).$$

Again, a simple approximation of the Hessian is a block diagonal version of the preconditioner proposed in Section 3.2.2, the i^{th} block being defined by $\alpha_i \text{Id}_{\mathcal{U}}$ with $\alpha_i = \langle P(u_r(p_a); p_a)v_i, v_i \rangle_{\mathcal{U}}$. Note that all proposed approximations of the Hessian require the computation of only one preconditioner at the parameter p_a .

4.3 Finite dimensional case - Algebraic form

We assume that \mathcal{U} is a finite dimensional vector space. Let $\{e_i\}_{i=1}^n$ be a basis of \mathcal{U} . We denote by $\mathcal{Q}_m = \text{span}\{\psi_j\}_{j=1}^m \subset \mathcal{Q}$, where $\{\psi_j\}_{j=1}^m$ is a basis of \mathcal{Q}_m . We introduce thus a finite dimensional space $\mathcal{Q}_m \otimes \mathcal{U} \subset \mathcal{Q} \otimes \mathcal{U}$ for approximating the solution.

A tensor $u \in \mathcal{Q}_m \otimes \mathcal{U}$ can thus be written

$$u = \sum_{i=1}^m \sum_{j=1}^n u_{ij} \psi_i \otimes e_j,$$

and a low-rank tensor is given as

$$u_r = F_r(\boldsymbol{\lambda}, \mathbf{v}) = \sum_{k=1}^r \lambda_k \otimes v_k = \sum_{i=1}^m \sum_{j=1}^n \left(\sum_{k=1}^r \lambda_{ik} v_{jk} \right) \psi_i \otimes e_j.$$

We denote by $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times r}$, $\mathbf{V} \in \mathbb{R}^{n \times r}$ and $\mathbf{F}_r(\boldsymbol{\Lambda}, \mathbf{V}) \in \mathbb{R}^{m \times n}$ matrices such that

$$\boldsymbol{\Lambda}_{ik} = \lambda_{ik}, \quad \mathbf{V}_{jk} = v_{jk} \quad \text{and} \quad \mathbf{F}_r(\boldsymbol{\Lambda}, \mathbf{V}) = \boldsymbol{\Lambda} \mathbf{V}^T.$$

With these notations, a low-rank tensor can be expressed under the different forms

$$u_r = F_r(\boldsymbol{\lambda}, \mathbf{v}) = \sum_{k=1}^r \lambda_k \otimes v_k = \sum_{i=1}^m \sum_{j=1}^n \mathbf{F}_r(\boldsymbol{\Lambda}, \mathbf{V})_{ij} \psi_i \otimes e_j,$$

thus allowing the identification of F_r with $\mathbf{F}_r : \mathbb{R}^{m \times r} \times \mathbb{R}^{n \times r} \rightarrow \mathbb{R}^{m \times n}$. Denoting by $\mathbf{J}_{\mathcal{P}} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ the functional such that $J_{\mathcal{P}} \circ F_r(\boldsymbol{\lambda}, \mathbf{v}) = \mathbf{J}_{\mathcal{P}} \circ \mathbf{F}_r(\boldsymbol{\Lambda}, \mathbf{V})$, we can consider the minimization problem, equivalent to Problem (12), defined by

$$\min_{\boldsymbol{\Lambda} \in \mathbb{R}^{m \times r}, \mathbf{V} \in \mathbb{R}^{n \times r}} \mathbf{J}_{\mathcal{P}} \circ \mathbf{F}_r(\boldsymbol{\Lambda}, \mathbf{V}),$$

and directly use all the algorithms described in Section 3 and 4 in the general setting.

We denote by $\boldsymbol{\psi}_z$ the vector of evaluations of the basis functions of $\mathcal{Q}_m \subset \mathcal{Q} = L^2(\mathcal{P})$ at the parameter value $p_z \in \mathcal{P}$, defined by $\boldsymbol{\psi}_z = (\psi_i(p_z))_{i=1}^m \in \mathbb{R}^m$.

We set $u_r(p_z) = F_r(\boldsymbol{\lambda}, \mathbf{v})(p_z)$ and denote by $\mathbf{R}(u_r(p_z); p_z) \in \mathbb{R}^n$ the vector defined by

$$(\mathbf{R}(u_r(p_z); p_z))_j = \langle \mathbf{R}(u_r(p_z); p_z), e_j \rangle_{\mathcal{U}} = \langle b(p_z) - A(u_r(p_z); p_z), e_j \rangle_{\mathcal{U}}, \quad \forall j \in \{1, \dots, n\}.$$

We deduce the algebraic form of (15) and (16), that is

$$\begin{aligned} \langle R_{\boldsymbol{\lambda}}(\mathbf{v}), \delta\mathbf{v} \rangle_{\mathcal{U}^r} &\approx \sum_z w_z \boldsymbol{\psi}_z^T \boldsymbol{\Lambda} \delta\mathbf{V}^T \mathbf{R}(u_r(p_z); p_z), \\ &= \left\langle \left(\sum_z w_z \mathbf{R}(u_r(p_z); p_z) \boldsymbol{\psi}_z^T \right) \boldsymbol{\Lambda}, \delta\mathbf{V} \right\rangle_{\mathbb{R}^{n \times r}}, \end{aligned}$$

and

$$\begin{aligned} \langle R_v(\boldsymbol{\lambda}), \delta \boldsymbol{\lambda} \rangle_{\mathcal{Q}^r} &\approx \sum_z w_z \boldsymbol{\psi}_z^T \delta \boldsymbol{\Lambda} \mathbf{V}^T \mathbf{R}(u_r(p_z); p_z), \\ &= \left\langle \left(\sum_z w_z \boldsymbol{\psi}_z \mathbf{R}(u_r(p_z); p_z)^T \right) \mathbf{V}, \delta \boldsymbol{\Lambda} \right\rangle_{\mathbb{R}^{m \times r}} \end{aligned}$$

where $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbb{R}^{s \times t}} = \text{trace}(\mathbf{X}\mathbf{Y}^T)$ is the canonical inner product in the matrix space $\mathbb{R}^{s \times t}$. This clearly shows that $R_\lambda(\mathbf{v})$ and $R_v(\boldsymbol{\lambda})$ can be evaluated in a non-intrusive fashion thanks to simple evaluations of the original residual $R(u_r(p_z); p_z) = b(p_z) - A(u_r(p_z); p_z)$ defined in Section 2 and hence is a non-intrusive computation.

5 Numerical examples

5.1 Electronic network

In this section, we use the example introduced in [8]. It is a simple electronic network. The original equation to be solved is

$$Bu(p) + (p_1 + 2)(u(p)^T u(p))u(p) = (p_2 + 25)f,$$

with

$$B = \frac{1}{R} \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & -1 & 4 \end{pmatrix}, \quad f = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad R = 100,$$

where the matrix B represents the network from Figure 1, and $p = (p_1, p_2)$ where p_1 and p_2 are uniform random variables on $[-1, 1]$. The matrix has this simple form as we have chosen all resistors equal. The problem is related to the minimization of the functional

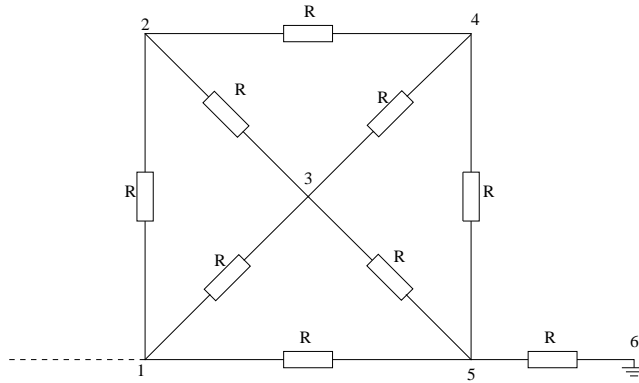


Figure 1: Electronic network.

defined by

$$J(v; p) = \frac{1}{2} v^T B v + \frac{1}{4} (p_1 + 2)(v^T v)^2 - (p_2 + 25)v^T f.$$

The residual $R(u(p); p) = -\nabla J(u(p); p)$ is thus given by

$$R(u(p); p) = (p_2 + 25)f - \left(Bu(p) + (p_1 + 2)(u(p)^T u(p))u(p) \right).$$

Proposition 5.1. *The assumptions of Theorem 2.1 are satisfied.*

Proof. See Appendix C. □

The basis of the finite dimensional stochastic space $\mathcal{Q}_m \subset \mathcal{Q}$ used for the approximation is chosen to be the multidimensional Legendre polynomials $\{\psi_j\}_{j=1}^m$ of total degree d which are orthogonal for the measure μ . For the quadrature rule, we have chosen a full tensorization of unidimensional Gauss-Legendre quadrature with $(d + 1)$ points, such that the total number of quadrature points is $(d + 1)^2$.

We can now directly apply the basic PGD procedure shown in Algorithm 1 and the improved algorithm described in Algorithm 3 in order to find a low-rank approximation of the solution map u . The convergence of the algorithms is controlled by a stagnation criterion.

Concerning the basic PGD, the stagnation criterion is set to 10^{-2} with a maximum of 10 iterations for each alternating minimization algorithm. The tolerance of the BFGS method is set to 10^{-10} . For Problem (6), the initialization of the preconditioner for the BFGS algorithm is the identity, and B for Problem (5). Moreover λ is initialized with a vector full of ones and v with a vector full of 10^{-8} . Being close to 0 is beneficial for later iterations, as the corrections will only slightly improve the approximation. However we observed that initializing v to 0 may induce that the next problem on λ becomes ill-conditioned due to the equality $\lambda \otimes 0 = 0$ for all λ . For the improved PGD, the stagnation criterion is set to $\max(10^{-(r+1)}, 10^{-8})$ with a maximum of 20 iterations for the alternating minimization algorithm. Both algorithms are initialized with $u_0 = 0$.

The relative error is measured with respect to the norm

$$\begin{aligned} \varepsilon(u_r) &= \frac{\|u - u_r\|_{\mathcal{Q} \otimes \mathcal{U}}}{\|u\|_{\mathcal{Q} \otimes \mathcal{U}}} = \sqrt{\frac{\int_{\mathcal{P}} \|u(p) - u_r(p)\|_{\mathcal{U}}^2 \mu(dp)}{\int_{\mathcal{P}} \|u(p)\|_{\mathcal{U}}^2 \mu(dp)}} \\ &\approx \sqrt{\frac{\sum_z w_z \|u(p_z) - u_r(p_z)\|_{\mathcal{U}}^2}{\sum_z w_z \|u(p_z)\|_{\mathcal{U}}^2}}, \end{aligned} \quad (17)$$

with u the exact solution, u_r the low-rank approximation and using a fully tensorized Gauss-Legendre quadrature with a number of points $20^2 = 400$. The deterministic solutions $\{u(p_z)\}$ are computed using a modified Newton algorithm where the tangent matrix is chosen to be the linear part B of the Hessian of the functional J . The low-rank approximations are also compared to the full-rank Galerkin approximation computed with the block-Jacobi algorithm introduced in [8], with a stagnation criterion of 10^{-10} . The comparison is made in Table 1 for total degrees $d = 2, 3, 4, 5$ and ranks $1, 2, 3, 4, 5$ for the approximations.

We can observe from Table 1 that the low-rank approximation gives a good approximation of the solution, even with a rank one approximation. Moreover, for a rank greater than 3, the low-rank approximation always gives results as good as the one of the full-rank Galerkin approximation. Besides, in this example, we can see that the greedy approximation gives satisfying results, even if the result is not optimal compared to the approximation resulting from a direct optimization in low-rank subsets.

For the rest of this section, we focus on $d = 5$ and we measure the efficiency of the different algorithms by counting the number of calls to the residual $R(u_r(p_z); p_z) = b(p_z) - A(u(p_z); p_z)$. The results are reported in Table 2.

Both algorithms are similar at the beginning until $r = 2$. When $r = 3$, Algorithm 3 becomes more efficient for computing the low-rank approximation. However, if we compare with the block-Jacobi solver, the latter one only requires 540 calls to the residual. This suggests that the classical algorithms for computing the low-rank approximation of the solution of nonlinear equations must be reconsidered in terms of efficiency and intrusivity and different approaches must be proposed.

	$d = 2$	$d = 3$	$d = 4$	$d = 5$
Block-Jacobi solver [8]				
	5.14×10^{-5}	3.31×10^{-6}	2.31×10^{-7}	1.70×10^{-8}
Basic PGD (Algorithm 1)				
$r = 1$	2.34×10^{-3}	2.34×10^{-3}	2.34×10^{-3}	2.34×10^{-3}
$r = 2$	9.67×10^{-5}	8.22×10^{-5}	8.22×10^{-5}	8.22×10^{-5}
$r = 3$	5.14×10^{-5}	3.39×10^{-6}	8.03×10^{-7}	7.78×10^{-7}
$r = 4$	5.14×10^{-5}	3.31×10^{-6}	2.34×10^{-7}	3.63×10^{-8}
$r = 5$	5.14×10^{-5}	3.31×10^{-6}	2.31×10^{-7}	1.71×10^{-8}
Improved PGD (Algorithm 3)				
$r = 1$	2.34×10^{-3}	2.34×10^{-3}	2.34×10^{-3}	2.34×10^{-3}
$r = 2$	5.14×10^{-5}	3.31×10^{-6}	2.85×10^{-7}	1.95×10^{-7}
$r = 3$	5.14×10^{-5}	3.31×10^{-6}	2.31×10^{-7}	1.79×10^{-8}
$r = 4$	5.14×10^{-5}	3.31×10^{-6}	2.31×10^{-7}	1.79×10^{-8}
$r = 5$	5.14×10^{-5}	3.31×10^{-6}	2.31×10^{-7}	1.76×10^{-8}

Table 1: Relative error for the approximation resulting from the block-Jacobi solver, the basic PGD and the improved algorithm for different total degrees d and different r .

	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$
Basic PGD (Algorithm 1)					
Relative error	2.34×10^{-3}	8.22×10^{-5}	7.78×10^{-7}	3.63×10^{-8}	1.71×10^{-8}
Residual calls	1044	2160	3096	3816	4464
Improved algorithm (Algorithm 3)					
Relative error	2.34×10^{-3}	1.95×10^{-7}	1.79×10^{-8}	1.79×10^{-8}	1.79×10^{-8}
Residual calls	1044	2304	2700	2844	3024

Table 2: Number of calls to the residual and corresponding relative error for different ranks r for the basic PGD and the improved algorithm.

5.2 Obstacle problem

We consider the obstacle problem introduced in [3]. A rope is clamped at its extremities over an obstacle modeled by a function g , and a force f is applied to the rope. Noting $\Omega = (0, 1)$, the vertical displacement of the rope is modeled by the function $v : \Omega \times \mathcal{P} \rightarrow \mathbb{R}$. The force is set to be constant $f \equiv 1$, and the obstacle is defined by

$$g(p; x) = p[\sin(3\pi x)]_+ + (p - 1)[\sin(3\pi x)]_-, \quad \forall (x, p) \in \Omega \times \mathcal{P},$$

where p is uniformly distributed in $\mathcal{P} = (0, 1)$. The non penetration condition (i.e. $v \geq g$) is taken into account with a penalty formulation, with the penalty coefficient $\rho = 10^3$. The reference solution u can be found by solving

$$\min_{v \in L^2(\mathcal{P}) \otimes H_0^1(\Omega)} J_{\mathcal{P}}(v),$$

with

$$J_{\mathcal{P}}(v) = \int_{\mathcal{P}} J(v(p); p) \mu(dp)$$

and

$$J(v; p) = \int_{\Omega} \frac{1}{2} \left(\frac{\partial}{\partial x} v(x) \right)^2 - f(p; x)v(x) + \frac{\rho}{2} [v(x) - g(p; x)]_+^2 dx.$$

Following the proofs in [3], we can show that the assumptions of Theorem 2.1 are satisfied. The domain Ω is discretized with 40 P1 finite elements, while we use piecewise

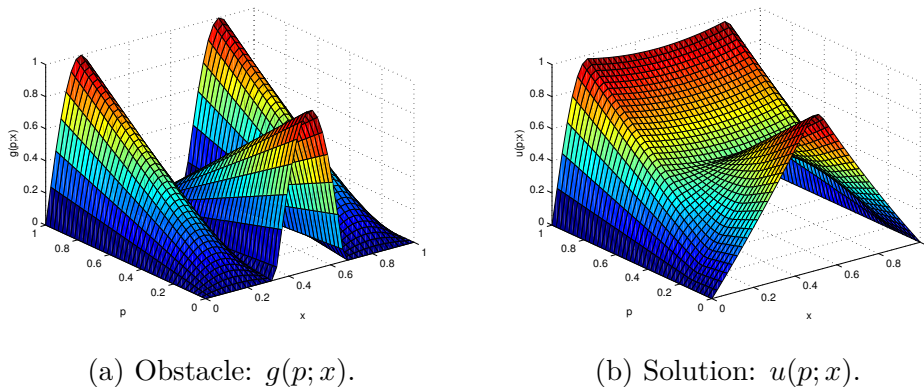


Figure 2: Obstacle and solution as functions of x and p [3].

polynomials of degree 1 on \mathcal{P} . The reference solution denoted by u_{L^2} is computed via a L^2 -projection, where a BFGS method has been applied at each quadrature point.

We use the relative error introduced in Equation (17) as an error estimate, where the solution u has been replaced with its L^2 -projection u_{L^2} . In Figure 3, the relative error with respect to the rank is shown. Three approximations are illustrated, the truncated SVD of the L^2 -projection, the basic PGD (Algorithm 1), and the improved algorithm (Algorithm 3). We used the same parameters as in Section 5.1 for the different algorithms, except for C_0 which is now constructed based on $(\nabla J(u_r(0); 0))^{-1}$. It is directly used in the basic PGD case, and a block-diagonal version is constructed for the improved PGD algorithm.

The basic PGD was the technique considered in Cancès et al. [3], asserting in Section 6.2 that “*this procedure is intrusive in general*”, while we have shown in this work that

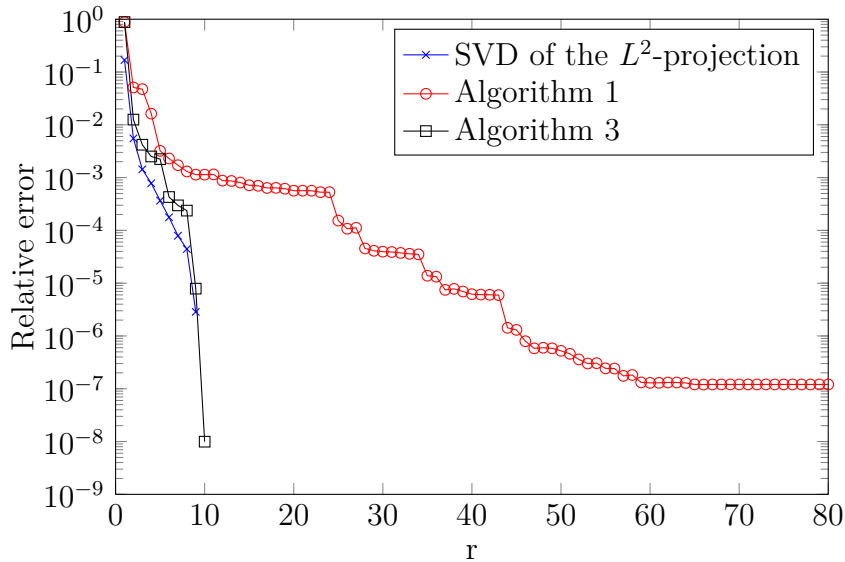


Figure 3: Relative error with respect to the rank of the approximation for different algorithms.

we can use a Galerkin approach in a non-intrusive fashion for constructing a low-rank approximation of the solution. Concerning the stopping criterion, we used the same as the one used in Section 5.1.

The SVD supplies the best rank r approximation of the reference solution with respect to the canonical norm $\|\cdot\|_{Q\otimes U}$. The basic PGD seems to slowly converge toward the solution of the problem with respect to the canonical norm, while the improved PGD has a similar convergence as the truncated SVD, and finally yields a relative error of 10^{-8} with a rank 10 approximation.

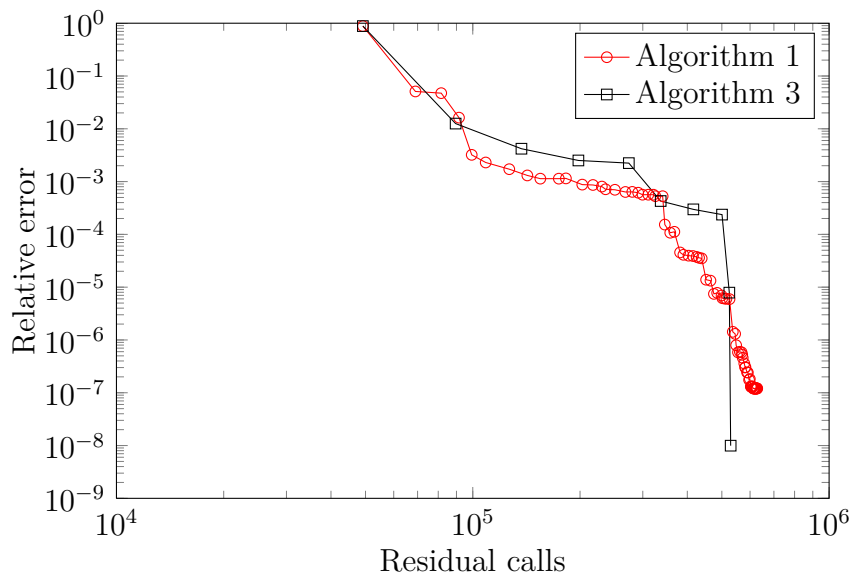


Figure 4: Relative error with respect to the number of calls to the residual for different algorithms.

The number of calls to the residual is shown in Figure 4 for Algorithms 1 and 3. We observe similar speeds of convergence for both algorithms, while the intrusive implementation of Algorithm 3 is usually slower when J is quadratic as it has been observed in [7]. Note that in this case, the basic PGD requires the construction of

the computation of 80 deterministic Hessians, to be compared to the 10 deterministic Hessians required by the improved PGD. Moreover, the L^2 -projection of the solution only requires 3304 calls to the residual, which is much less than the number of calls required for the rank one approximation of the solution.

This example illustrates that accurate low rank approximation of the solution of nonlinear problems can be directly obtained in a non-intrusive fashion. However, the proposed construction is clearly not efficient regarding the computational complexity and new algorithms that are adapted to this non-intrusive setting are clearly required.

6 Conclusion

In this work, algorithms for the non-intrusive computation of low-rank approximations of the solution of a nonlinear stochastic-parametric problem associated with the minimization of a convex functional have been proposed. The proposed approach relies on an alternating minimization algorithm and the BFGS method for minimizing the partial maps. The techniques are implemented in a non-intrusive way thanks to the use of numerical integration, and only require the evaluations of the standard residuum at some parameter values. The method has finally been applied to two model examples.

The goal of this paper was to revisit standard techniques for computing a low-rank approximation of the solution of a parametric equation. The novelty lies in the non-intrusive implementation of these algorithms. It results that the performance of the classical methods for computing the low-rank approximation is modified due to the expensive evaluation of samples (evaluations) of the residual for the numerical integration.

In order to reduce computational complexity, the number of iterations of the solver should be reduced to the minimum using more efficient algorithms than an alternating minimization method plus a BFGS algorithm. Indeed, one iteration of the solver corresponds to one integration of the residual. Thus, at each iteration and for each quadrature point, one sample of the residual must be evaluated.

Also, an adapted integration method could be used in order to further reduce the number of evaluations of the residual for high dimensional parametric problems. A first solution is to use an adaptive sparse grid quadrature technique [2] in order to reduce the number of quadrature points. Another solution is to build an approximation of the residual at each iteration. The costly integration of the residual could then be replaced by a cheaper approximation of this residual exploiting structured approximation techniques.

This work is a first step toward Galerkin-based methods for the low-rank approximation of the solution of parametric nonlinear equations in a non-intrusive manner, offering new opportunities for the development of efficient non-intrusive solvers.

A Proof of Theorem 2.1

Existence and uniqueness of the solution. Given that $v \mapsto J(v; p)$ is strongly convex for all p and continuous, there exists a unique solution to (1). As a consequence, we can define the solution map by $u : p \mapsto u(p) = \arg \min_{v \in \mathcal{U}} J(v; p)$.

Regularity of the solution. $v \mapsto J(v; p)$ is Fréchet differentiable and strongly convex uniformly in p . As a consequence, there exists $\alpha > 0$ independent of p such that

$$\langle \nabla J(u(p); p) - \nabla J(v; p), u(p) - v \rangle_{\mathcal{U}} \geq \alpha \|u(p) - v\|_{\mathcal{U}}^2, \quad \forall p \in \mathcal{P}, \forall v \in \mathcal{U}.$$

Given that $\nabla J(u(p); p) = 0$, and taking $v = 0$ we obtain

$$-\langle \nabla J(0; p), u(p) \rangle_{\mathcal{U}} = -\langle A(0; p) - b(p), u(p) \rangle_{\mathcal{U}} \geq \alpha \|u(p)\|_{\mathcal{U}}^2.$$

The Cauchy-Schwarz inequality gives

$$\|u(p)\|_{\mathcal{U}}^2 \leq \frac{1}{\alpha} \|A(0; p) - b(p)\|_{\mathcal{U}} \|u(p)\|_{\mathcal{U}},$$

then

$$\|u(p)\|_{\mathcal{U}}^2 \leq \frac{1}{\alpha^2} \|A(0; p) - b(p)\|_{\mathcal{U}}^2$$

which yields $u \in L^2(\mathcal{P}; \mathcal{U})$ using (c). In the following, the set $L^2(\mathcal{P}; \mathcal{U})$ is identified to the Hilbert space $\mathcal{Q} \otimes \mathcal{U}$ equipped with the induced product norm.

Characterization of the solution. Given that $p \mapsto J(v; p)$ is μ -integrable, we can define the functional $J_{\mathcal{P}} : \mathcal{Q} \otimes \mathcal{U} \rightarrow \mathbb{R}$ such that

$$J_{\mathcal{P}}(u) = \int_{\mathcal{P}} J(u(p); p) \mu(dp).$$

Lemma A.1. $J_{\mathcal{P}}$ is strongly convex.

Proof. According to the assumptions of Theorem 2.1, $v \mapsto J(v; p)$ is strongly convex uniformly in p . We conclude that, for all v, w in $\mathcal{Q} \otimes \mathcal{U}$,

$$\begin{aligned} J_{\mathcal{P}}(tv + (1-t)w) &= \int_{\mathcal{P}} J(tv(p) + (1-t)w(p); p) \mu(dp) \\ &\leq \int_{\mathcal{P}} \left(tJ(v(p); p) + (1-t)J(w(p); p) - \frac{\alpha}{2} t(1-t) \|v(p) - w(p)\|_{\mathcal{U}}^2 \right) \mu(dp) \\ &\leq tJ_{\mathcal{P}}(v) + (1-t)J_{\mathcal{P}}(w) - \frac{\alpha}{2} t(1-t) \|v - w\|_{L^2(\mathcal{P}; \mathcal{U})}^2, \end{aligned}$$

and $J_{\mathcal{P}}$ is strongly convex. □

Lemma A.2. $J_{\mathcal{P}}$ is Gâteaux differentiable with Gâteaux derivative $\delta J_{\mathcal{P}}(u)(\delta u)$ at $u \in L^2(\mathcal{P}; \mathcal{U}) = \mathcal{Q} \otimes \mathcal{U}$ in the direction $\delta u \in \mathcal{Q} \otimes \mathcal{U}$ given by

$$\delta J_{\mathcal{P}}(u)(\delta u) = \int_{\mathcal{P}} \langle \nabla J(u(p); p), \delta u(p) \rangle_{\mathcal{U}} \mu(dp).$$

Proof. Let $t \in (-1, 1) \setminus \{0\}$ and $f_t(p)$ being defined by

$$f_t(p) = \frac{1}{t} (J(u(p) + t\delta u(p); p) - J(u(p); p)).$$

Thanks to the fundamental theorem of calculus and Cauchy-Schwarz inequality we have

$$\begin{aligned} f_t(p) &= \int_0^1 \langle \nabla J(u(p) + \beta t \delta u(p); p), \delta u(p) \rangle_{\mathcal{U}} d\beta \\ &\leq \int_0^1 \|\nabla J(u(p) + \beta t \delta u(p); p)\|_{\mathcal{U}} \|\delta u(p)\|_{\mathcal{U}} d\beta. \end{aligned}$$

Given that $v \mapsto \nabla J(v; p) = A(v; p) - b(p)$ is Lipschitz on bounded set \mathcal{S} uniformly in p , with \mathcal{S} being the ball centered in 0 of radius $\|u(p) + \delta u(p)\|_{\mathcal{U}}$, there exists $K > 0$ such that

$$\begin{aligned} \|\nabla J(u(p) + \beta t \delta u(p); p)\|_{\mathcal{U}} - \|\nabla J(0; p)\|_{\mathcal{U}} &\leq \|\nabla J(u(p) + \beta t \delta u(p); p) - \nabla J(0; p)\|_{\mathcal{U}} \\ &\leq K \|u(p) + \beta t \delta u(p)\|_{\mathcal{U}} \\ &\leq K (\|u(p)\|_{\mathcal{U}} + \|\delta u(p)\|_{\mathcal{U}}), \end{aligned}$$

and finally

$$\begin{aligned} f_t(p) &\leq \int_0^1 (K(\|u(p)\|_{\mathcal{U}} + \|\delta u(p)\|_{\mathcal{U}}) + \|\nabla J(0; p)\|_{\mathcal{U}}) \|\delta u(p)\|_{\mathcal{U}} d\beta \\ &\leq (K(\|u(p)\|_{\mathcal{U}} + \|\delta u(p)\|_{\mathcal{U}}) + \|A(0; p) - b(p)\|_{\mathcal{U}}) \|\delta u(p)\|_{\mathcal{U}} = g(p). \end{aligned}$$

With u , δu and $p \mapsto A(0; p) - b(p)$ being in $L^2(\mathcal{P}; \mathcal{U})$, g is in $L^1(\mathcal{P}; \mathcal{U})$. We can thus applied the dominated convergence theorem and state that the limit $\delta J_{\mathcal{P}}(u)(\delta u) = \lim_{t \rightarrow 0} \int_{\mathcal{P}} f_t(p) \mu(dp)$ exists and that

$$\delta J_{\mathcal{P}}(u)(\delta u) = \int_{\mathcal{P}} \langle \nabla J(u(p); p), \delta u(p) \rangle_{\mathcal{U}} \mu(dp).$$

□

Lemma A.3. $J_{\mathcal{P}}$ is Fréchet differentiable and, for all u , δu in $L^2(\mathcal{P}; \mathcal{U})$, we have

$$\langle \nabla J_{\mathcal{P}}(u), \delta u \rangle_{L^2(\mathcal{P}; \mathcal{U})} = \int_{\mathcal{P}} \langle \nabla J(u(p); p), \delta u(p) \rangle_{\mathcal{U}} \mu(dp).$$

Proof. The application $\delta J_{\mathcal{P}} : L^2(\mathcal{P}; \mathcal{U}) \ni u \mapsto \delta J_{\mathcal{P}}(u) \in L^2(\mathcal{P}; \mathcal{U})^*$ is linear. Let u and δu in $L^2(\mathcal{P}; \mathcal{U})$. Let $\mathcal{S} \subset \mathcal{Q} \otimes \mathcal{U}$, be a bounded set containing u and $u + \delta u$. Let $v \in L^2(\mathcal{P}; \mathcal{U})$. There exists $K > 0$ such that

$$\begin{aligned} |(\delta J_{\mathcal{P}}(u + \delta u) - \delta J_{\mathcal{P}}(u))(v)| &= \left| \int_{\mathcal{P}} \langle \nabla J(u(p) + \delta u(p); p) - \nabla J(u(p); p), v(p) \rangle_{\mathcal{U}} \mu(dp) \right| \\ &\leq \int_{\mathcal{P}} \|\nabla J(u(p) + \delta u(p); p) - \nabla J(u(p); p)\|_{\mathcal{U}} \|v(p)\|_{\mathcal{U}} \mu(dp) \\ &\leq \int_{\mathcal{P}} K \|\delta u(p)\|_{\mathcal{U}} \|v(p)\|_{\mathcal{U}} \mu(dp) \\ &\leq K \|\delta u\|_{L^2(\mathcal{P}; \mathcal{U})} \|v\|_{L^2(\mathcal{P}; \mathcal{U})}. \end{aligned}$$

It follows that $\delta J_{\mathcal{P}} : L^2(\mathcal{P}; \mathcal{U}) \rightarrow L^2(\mathcal{P}; \mathcal{U})^*$ is continuous. As a consequence, $J_{\mathcal{P}}$ is Fréchet differentiable with Fréchet derivative

$$\langle \nabla J_{\mathcal{P}}(u), \delta u \rangle_{L^2(\mathcal{P}; \mathcal{U})} = \delta J_{\mathcal{P}}(u)(\delta u) = \int_{\mathcal{P}} \langle \nabla J(u(p); p), \delta u(p) \rangle_{\mathcal{U}} \mu(dp).$$

□

According to Lemma A.1, there exists a unique minimizer to $J_{\mathcal{P}}$. The combination of Lemmas A.1 and A.3 ensures that the minimizer is equivalently characterized by

$$\langle \nabla J_{\mathcal{P}}(u), \delta u \rangle_{L^2(\mathcal{P}; \mathcal{U})} = \int_{\mathcal{P}} \langle A(u(p); p) - b(p), \delta u(p) \rangle_{\mathcal{U}} \mu(dp) = 0, \quad \forall \delta u \in L^2(\mathcal{P}; \mathcal{U}).$$

Given that the map $u : p \mapsto \arg \min_{v \in \mathcal{U}} J(v; p)$ is in $L^2(\mathcal{P}; \mathcal{U})$ and satisfies $A(u(p); p) - b(p) = 0$ for all p , it is the unique minimizer of $J_{\mathcal{P}}$.

B Proof of Theorem 4.3

Existence and uniqueness of the solutions. According to Lemma A.1, $J_{\mathcal{P}}$ is strongly convex. Let $J_{\mathcal{P}}^v$ and $J_{\mathcal{P}}^\lambda$ denote $\lambda \mapsto J_{\mathcal{P}}^v(\lambda) = J_{\mathcal{P}} \circ F_r(\lambda, \mathbf{v})$ and $\mathbf{v} \mapsto J_{\mathcal{P}}^\lambda(\mathbf{v}) = J_{\mathcal{P}} \circ F_r(\lambda, \mathbf{v})$. For $t \in (0, 1)$, given that F_r is bilinear (Lemma 4.1) we have

$$\begin{aligned} J_{\mathcal{P}}^\lambda(t\mathbf{v} + (1-t)\mathbf{w}) &= J_{\mathcal{P}}(tF_r(\lambda, \mathbf{v}) + (1-t)F_r(\lambda, \mathbf{w})) \\ &\leq tJ_{\mathcal{P}}^\lambda(\mathbf{v}) + (1-t)J_{\mathcal{P}}^\lambda(\mathbf{w}) - \frac{\alpha}{2}t(1-t) \|F_r(\lambda, \mathbf{v} - \mathbf{w})\|_{\mathcal{Q} \otimes \mathcal{U}}^2. \end{aligned}$$

We have

$$\begin{aligned}
\|F_r(\boldsymbol{\lambda}, \mathbf{v} - \mathbf{w})\|_{\mathcal{Q} \otimes \mathcal{U}}^2 &= \sum_{i=1}^r \sum_{j=1}^r \langle \lambda_i, \lambda_j \rangle_{\mathcal{Q}} \langle v_i - w_i, v_j - w_j \rangle_{\mathcal{U}} \\
&= \sum_{i=1}^r \left\langle v_i - w_i, \sum_{j=1}^r \langle \lambda_i, \lambda_j \rangle_{\mathcal{Q}} (v_j - w_j) \right\rangle_{\mathcal{U}} \\
&= \langle \mathbf{v} - \mathbf{w}, G_{\boldsymbol{\lambda}}(\mathbf{v} - \mathbf{w}) \rangle_{\mathcal{U}^r},
\end{aligned}$$

$G_{\boldsymbol{\lambda}} \in \mathbb{R}^{r \times r}$ being the Gram matrix associated to $\boldsymbol{\lambda}$ and its application to $\mathbf{v} - \mathbf{w}$ being detailed in Section 4.1. If $\boldsymbol{\lambda}$ is a set of linearly independent functions, then $G_{\boldsymbol{\lambda}}$ is symmetric positive definite. It defines then an induced norm such that $\|F_r(\boldsymbol{\lambda}, \mathbf{v} - \mathbf{w})\|_{\mathcal{Q} \otimes \mathcal{U}}^2 = \|\mathbf{v} - \mathbf{w}\|_{G_{\boldsymbol{\lambda}}}^2$ yielding that $J_{\mathcal{P}}^{\boldsymbol{\lambda}}$ is strongly convex. Note that when $\boldsymbol{\lambda}$ is an orthonormal set, we have $\|F_r(\boldsymbol{\lambda}, \mathbf{v} - \mathbf{w})\|_{\mathcal{Q} \otimes \mathcal{U}}^2 = \|\mathbf{v} - \mathbf{w}\|_{\mathcal{U}^r}^2$ and $J_{\mathcal{P}}^{\boldsymbol{\lambda}}$ has the same convexity constant than $J(\cdot; p)$. Similarly, if \mathbf{v} is a set of linearly independent vectors, $J_{\mathcal{P}}^{\mathbf{v}}$ is strongly convex. As a consequence, there exists a unique solution to the minimization problems

$$\min_{\boldsymbol{\lambda} \in \mathcal{Q}^r} J_{\mathcal{P}}^{\mathbf{v}}(\boldsymbol{\lambda}) \quad \text{and} \quad \min_{\mathbf{v} \in \mathcal{U}^r} J_{\mathcal{P}}^{\boldsymbol{\lambda}}(\mathbf{v}).$$

Characterization of the solutions. $J_{\mathcal{P}}$ and F_r being Fréchet differentiable, $J_{\mathcal{P}}^{\mathbf{v}}$ and $J_{\mathcal{P}}^{\boldsymbol{\lambda}}$ are Fréchet differentiable. Given that $J_{\mathcal{P}}^{\mathbf{v}}$ and $J_{\mathcal{P}}^{\boldsymbol{\lambda}}$ are strongly convex, we know that the solution to the minimization problems are uniquely characterized by the equations

$$\langle \nabla J_{\mathcal{P}}^{\mathbf{v}}(\boldsymbol{\lambda}), \delta \boldsymbol{\lambda} \rangle_{\mathcal{Q}^r} = 0, \quad \forall \delta \boldsymbol{\lambda} \in \mathcal{Q}^r \quad \text{and} \quad \langle \nabla J_{\mathcal{P}}^{\boldsymbol{\lambda}}(\mathbf{v}), \delta \mathbf{v} \rangle_{\mathcal{U}^r} = 0, \quad \forall \delta \mathbf{v} \in \mathcal{U}^r.$$

In Appendix A, we established that

$$\langle \nabla J_{\mathcal{P}}(u), \delta u \rangle_{L^2(\mathcal{P}; \mathcal{U})} = \int_{\mathcal{P}} \langle A(u(p); p) - b(p), \delta u(p) \rangle_{\mathcal{U}} \mu(\mathrm{d}p), \quad \forall u, \delta u \in \mathcal{Q} \otimes \mathcal{U}.$$

Using the chain rule, we find that

$$\begin{aligned}
\langle \nabla J_{\mathcal{P}}^{\mathbf{v}}(\boldsymbol{\lambda}), \delta \boldsymbol{\lambda} \rangle_{\mathcal{Q}^r} &= \int_{\mathcal{P}} \langle A(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) - b(p), F_r(\delta \boldsymbol{\lambda}, \mathbf{v})(p) \rangle_{\mathcal{U}} \mu(\mathrm{d}p) \\
\text{and} \quad \langle \nabla J_{\mathcal{P}}^{\boldsymbol{\lambda}}(\mathbf{v}), \delta \mathbf{v} \rangle_{\mathcal{U}^r} &= \int_{\mathcal{P}} \langle A(F_r(\boldsymbol{\lambda}, \mathbf{v})(p); p) - b(p), F_r(\boldsymbol{\lambda}, \delta \mathbf{v})(p) \rangle_{\mathcal{U}} \mu(\mathrm{d}p).
\end{aligned}$$

C Proof of Proposition 5.1

- (a) Since B is symmetric positive definite, the functional J is given by $J(v; p) = \frac{1}{2}v^T B v + \frac{1}{4}(p_1 + 2)(v^T v)^2 - (p_2 + 25)v^T f$.
- (b) $v \mapsto J(v; p)$ is clearly Fréchet differentiable. Moreover, the Hessian of $v \mapsto J(v; p)$ is given by $v \mapsto H(v; p) = B + (p_1 + 2)(v^T v I + v \otimes v)$. We have $\delta v^T H(v; p) \delta v \geq \delta v^T B \delta v \geq \alpha \|\delta v\|_{\mathcal{U}}^2$, α being the smallest eigenvalue of B which is independent of p .
- (c) $p \mapsto -(p_2 + 25)f$ is integrable on \mathcal{P} .
- (d) $p \mapsto J(v; p)$ is clearly integrable on \mathcal{P} . Concerning the Lipschitz continuity property, given a bounded set $\mathcal{S} \subset \mathcal{U}$, and given that $p_1 \leq 1$, we have

$$\begin{aligned}
\|\nabla J(v; p) - \nabla J(w; p)\|_{\mathcal{U}} &= \left\| B(v - w) + (p_1 + 2)(\|v\|_{\mathcal{U}}^2 v - \|w\|_{\mathcal{U}}^2 w) \right\|_{\mathcal{U}} \\
&\leq \|B\| \|v - w\|_{\mathcal{U}} + 3 \left\| \|v\|_{\mathcal{U}}^2 v - \|w\|_{\mathcal{U}}^2 w \right\|_{\mathcal{U}},
\end{aligned}$$

for all $v, w \in \mathcal{S}$, where $\|B\|$ is the operator norm. We introduce the operator $C = (v \otimes v + v \otimes w + w \otimes v + w \otimes w)$. It is bounded since \mathcal{S} is bounded and we can define $\|C\|$. We notice then

$$\begin{aligned} \left\| \|v\|_{\mathcal{U}}^2 v - \|w\|_{\mathcal{U}}^2 w \right\|_{\mathcal{U}} &= \left\| C(v - w) + \|v\|_{\mathcal{U}}^2 (v - w) - (\|v\|_{\mathcal{U}} - \|w\|_{\mathcal{U}})(\|v\|_{\mathcal{U}} + \|w\|_{\mathcal{U}})v \right\|_{\mathcal{U}}, \\ &\leq \left(\|C\| + \|v\|_{\mathcal{U}}^2 \right) \|v - w\|_{\mathcal{U}} + |\|v\|_{\mathcal{U}} - \|w\|_{\mathcal{U}}| (\|v\|_{\mathcal{U}} + \|w\|_{\mathcal{U}}) \|v\|_{\mathcal{U}}, \\ &\leq \left(\|C\| + \|v\|_{\mathcal{U}}^2 + (\|v\|_{\mathcal{U}} + \|w\|_{\mathcal{U}}) \|v\|_{\mathcal{U}} \right) \|v - w\|_{\mathcal{U}}, \end{aligned}$$

using the inequality $|\|v\|_{\mathcal{U}} - \|w\|_{\mathcal{U}}| \leq \|v - w\|_{\mathcal{U}}$. For a given bounded set $\mathcal{S} \subset \mathcal{U}$, we denote by $D = \sup_{v \in \mathcal{S}} \|v\|_{\mathcal{U}}$. With $K = \|B\| + 3\|C\| + 9D^2 > 0$, we finally deduce

$$\|\nabla J(v; p) - \nabla J(w; p)\|_{\mathcal{U}} \leq K \|v - w\|_{\mathcal{U}}, \quad \forall v, w \in \mathcal{S},$$

with K independent of p .

References

- [1] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra with Applications*, 20(1):27–43, Jan. 2013.
- [2] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147, June 2004.
- [3] E. Cancès, V. Ehrlicher, and T. Lelièvre. Convergence of a greedy algorithm for high-dimensional convex nonlinear problems. *Mathematical Models and Methods in Applied Sciences*, 21(12):2433–2467, Dec. 2011.
- [4] F. Chinesta, P. Ladeveze, and E. Cueto. A Short Review on Model Order Reduction Based on Proper Generalized Decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, Oct. 2011.
- [5] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-hall edition, 1983.
- [6] A. Falcó and A. Nouy. Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces. *Numerische Mathematik*, 121(3):503–530, Dec. 2012.
- [7] L. Giraldi. *Contributions aux méthodes de calcul basées sur l'approximation de tenseurs et applications en mécanique numérique*. PhD thesis, Nantes, 2012.
- [8] L. Giraldi, A. Litvinenko, D. Liu, H. G. Matthies, and A. Nouy. To be or not to be intrusive? The solution of parametric and stochastic equations - the "plain vanilla" Galerkin case. arXiv:1309.1617 [math.NA], Sept. 2013.
- [9] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtyshnikov. Approximate iterations for structured matrices. *Numerische Mathematik*, 109(3):365–383, Feb. 2008.
- [10] T. G. Kolda and B. W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, Aug. 2009.
- [11] D. Kressner and C. Tobler. Low-Rank Tensor Krylov Subspace Methods for Parametrized Linear Systems. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1288–1316, Oct. 2011.

- [12] H. G. Matthies and A. Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 194(12-16):1295–1331, Apr. 2005.
- [13] H. G. Matthies and G. Strang. The solution of nonlinear finite element equations. *International Journal for Numerical Methods in Engineering*, 14(11):1613–1626, Jan. 1979.
- [14] H. G. Matthies and E. Zander. Solving stochastic systems with low-rank tensor compression. *Linear Algebra and its Applications*, 436(10):3819–3838, May 2012.
- [15] A. Nouy. Recent Developments in Spectral Stochastic Methods for the Numerical Solution of Stochastic Partial Differential Equations. *Archives of Computational Methods in Engineering*, 16(3):251–285, May 2009.