Advances in Computational Mathematics (2024) 50:19
https://doi.org/10.1007/s10444-024-10114-x

A probabilistic reduced basis method for parameter-dependent problems

Marie Billaud-Friess[1] · Arthur Macherey[1,2] · Anthony Nouy[1] · Clémentine Prieur[2]

Received: 19 April 2023 / Accepted: 15 January 2024 / Published online: 13 March 2024

Abstract

Probabilistic variants of model order reduction (MOR) methods have recently emerged for improving stability and computational performance of classical approaches. In this paper, we propose a probabilistic reduced basis method (RBM) for the approximation of a family of parameter-dependent functions. It relies on a probabilistic greedy algorithm with an error indicator that can be written as an expectation of some parameter-dependent random variable. Practical algorithms relying on Monte Carlo estimates of this error indicator are discussed. In particular, when using probably approximately correct (PAC) bandit algorithm, the resulting procedure is proven to be a weak-greedy algorithm with high probability. Intended applications concern the approximation of a parameter-dependent family of functions for which we only have access to (noisy) pointwise evaluations. As a particular application, we consider the approximation of solution manifolds of linear parameter-dependent partial differential equations with a probabilistic interpretation through the Feynman-Kac formula.

Keywords Reduced basis method · Probabilistic greedy algorithm · Parameter-dependent partial differential equation · Feynman-Kac formula

Mathematics Subject Classification (2010) 65N75 · 65D15

Communicated by: Tobias Breiten

✉ Marie Billaud-Friess
marie.billaud-friess@ec-nantes.fr

Anthony Nouy
anthony.nouy@ec-nantes.fr

Clémentine Prieur
clementine.prieur@univ-grenoble-alpes.fr

[1] Centrale Nantes, Nantes Université, LMJL UMR CNRS 6629, Nantes, France

[2] University Grenoble Alpes, CNRS, Inria, Grenoble INP*, LJK, 38000 Grenoble, France

1 Introduction

This article focuses on the approximation of a family of functions $\mathcal{M} = \{u(\xi) : \xi \in \Xi\}$ indexed by a parameter ξ, each function $u(\xi)$ being an element of some high-dimensional vector space V. The functions $u(\xi)$ can be known a priori, or implicitly given through parameter-dependent equations. In multi-query contexts such as optimization, control, or uncertainty quantification, one is interested in computing $u(\xi)$ for many instances of the parameter. For complex numerical models, this can be computationally intractable. Model order reduction (MOR) methods aim at providing an approximation $u_n(\xi)$ of $u(\xi)$ which can be evaluated efficiently for any ξ in the parameter set Ξ. For linear approximation methods, an approximation $u_n(\xi)$ is obtained by means of a projection onto a low-dimensional subspace V_n which is chosen to approximate at best $\mathcal{M}$, uniformly over Ξ for empirical interpolation method (EIM) or reduced basis method (RBM), or in a mean-square sense for proper orthogonal decomposition (POD) or proper generalized decomposition (PGD) methods (see, e.g., the survey [1]).

Probabilistic variants of MOR methods have been recently proposed for improving stability and computational performance of classical MOR methods. In [2], the authors introduced a probabilistic greedy algorithm for the construction of reduced spaces V_n, which uses different training sets in Ξ with moderate cardinality, randomly chosen at each iteration, that allows a sparse exploration of a possibly high-dimensional parameter set. In [3], the authors derive a similar probabilistic EIM using sequential sampling in Ξ, which provides an interpolation with a prescribed precision with high probability. Let us also mention that a control variate method using a reduced basis paradigm has been proposed in [4] for Monte Carlo (MC) estimation of the expectation of a collection of random variables $u(\xi)$ in a space V of second-order random variables. A greedy algorithm is introduced to select a subspace V_n of random variables, that relies on a statistical estimation of the projection error. This algorithm has been analyzed in [5] and proven to be a weak-greedy algorithm with high probability. Probabilistic approaches have also been introduced for providing efficient and numerically stable error estimates for reduced order models [6–9]. In [10–14], random sketching methods have been systematically used in different tasks of projection-based model order reduction, including the construction of reduced spaces or libraries of reduced spaces, the projection onto these spaces, the error estimation, and preconditioning.

Here, we consider the problem of computing an approximation u_n of u within a reduced basis framework. The reduced basis method performs in two steps, *offline* and *online*. During the offline stage, a reduced space V_n is generated from snapshots $u(\xi_i)$ at parameter values ξ_i greedily selected by maximizing over Ξ (or some subset of Ξ) an error indicator $\Delta(u_{n-1}(\xi), \xi)$ which provides a measure of the discrepancy between $u(\xi)$ and $u_{n-1}(\xi)$. Then, during the online step, $u_n(\xi)$ is obtained by some projection onto V_n.

In this paper, we propose a probabilistic greedy algorithm for which $\Delta(u_n(\xi), \xi)$ is the square error norm $\|u(\xi) - u_n(\xi)\|_V^2$, expressed as the expectation of some parameter-dependent random variable $Z_n(\xi)$,

$$\Delta(u_n(\xi), \xi) = \mathbb{E}(Z_n(\xi)). \tag{1}$$

For maximizing $\mathbb{E}(Z_n(\xi))$, we rely on MC estimates. We consider either a naive MC approach with a fixed number of samples or a PAC (probably approximately correct) bandit algorithm proposed by the authors in [15] based on adaptive sampling. The algorithm only requires a limited number of samples by preferably sampling random variables associated with a probable maximizer ξ. It is particularly suitable for applications where the random variable $Z_n(\xi)$ is costly to sample. Under suitable assumptions on the distribution of $Z_n(\xi)$, it provides a PAC maximizer in relative precision, meaning that with high probability the parameter ξ is a quasi-optimal solution of the optimization problem. We prove in this work that the resulting greedy algorithm is a weak-greedy algorithm with high probability.

Intended applications concern the approximation of a parameter-dependent family of functions $u(\xi)$ defined on a bounded domain D for which we have access to (possibly noisy) pointwise evaluations $u(\xi)(x) := u(x, \xi)$ for any $x \in D$. The proposed probabilistic greedy algorithm can be used to generate a sequence of spaces V_n and corresponding interpolations u_n of u onto V_n. Assuming $u(\xi) \in L^2(D)$ and we have a direct access to pointwise evaluations, the square error norm $\|u(\xi) - u_n(\xi)\|^2_{L^2(D)}$ used to select the parameter ξ can be estimated from samples of $Z_n(\xi) = |D||u(Y, \xi) - u_n(Y, \xi)|^2$ with Y a uniform random variable over D. It results in a probabilistic EIM in the spirit of [3]. In a fully discrete setting where Ξ and D are finite sets, u can be identified with a matrix and the proposed algorithm is a probabilistic version of adaptive cross approximation for low-rank matrix approximation [16, 17], with a particular column-selection strategy. Another context is the solution of a linear parameter-dependent partial differential equation (PDE) defined on a bounded domain D and whose solution $u(\xi)$ admits a probabilistic representation through the Feynman-Kac formula. This allows to express a pointwise evaluation $u(x, \xi)$ as the expectation of a functional of some stochastic process. The problem being linear, the error $u(\xi) - u_n(\xi)$ also admits a Feynman-Kac representation, which again allows to express the square error norm $\Delta(u_n(\xi), \xi) = \|u(\xi) - u_n(\xi)\|^2_{L^2(D)}$ as the expectation of some random variable $Z_n(\xi)$ and to estimate it through Monte Carlo simulations of stochastic processes. This is a natural framework to apply the proposed probabilistic greedy algorithm, which allows a direct estimation of the targeted error norm and avoids the use of possibly highly biased residual-based error estimates. This leads to error estimators with better effectivity, which improves the behavior of weak-greedy algorithms. In practice, as the exact solution of the PDE is not available, the snapshots used for generating the reduced space V_n are numerical approximations computed from pointwise evaluations of the exact solution $u(\xi)$ by some interpolation or learning procedure. This results in a fully probabilistic setting which opens the route for the solution of high-dimensional PDEs (see, e.g., [18] where the authors rely on interpolation on sparse polynomial spaces).

This paper is structured as follows. In Section 2, we recall basic facts concerning the reduced basis method. Then, in Section 3, we present and analyze our new probabilistic greedy algorithm. Based on this algorithm, we derive in Section 4 a new reduced basis method for parameter-dependent PDEs with a probabilistic interpretation. Numerical results illustrating the performance of the proposed approaches are presented in Section 5.

2 Reduced basis greedy algorithms

As discussed in the introduction, the reduced basis method relies on two steps. We mainly focus on the offline stage during which the reduced subspace $V_n \subset V$ is constructed. In particular, we recall in this section some basic facts concerning greedy algorithms usually considered in that context. For a detailed overview on that topic see, e.g., surveys [19, 20].

Throughout this paper, V is some Hilbert space equipped with a norm $\| \cdot \|_V$. We seek an approximation $u_n(\xi)$ of $u(\xi)$ in a low-dimensional space V_n which is designed to well approximate the solution manifold

$$\mathcal{M} = \{u(\xi) : \xi \in \Xi\}.$$

A benchmark for optimal linear approximation is given by the Kolmogorov n-width

$$d_n(\mathcal{M})_V := \inf_{\dim V_n = n} \sup_{u \in \mathcal{M}} \|u - P_{V_n} u\|_V,$$

where the infimum is taken over all n-dimensional subspaces V_n of V and where P_{V_n} stands for the orthogonal projection onto V_n. However, an optimal space V_n is in general out of reach. A prominent approach is to rely on a greedy algorithm for generating a sequence of spaces from suitably selected parameter values. Starting from $V_0 = \{0\}$, the n-th step of this algorithm reads as follows. Given $\{\xi_1, \dots, \xi_{n-1}\} \subset \Xi$ and the corresponding subspace

$$V_{n-1} = \mathrm{span}\{u(\xi_1), \dots, u(\xi_{n-1})\},$$

a new parameter value ξ_n is selected as

$$\|u(\xi_n) - u_{n-1}(\xi_n)\|_V = \sup_{\xi \in \Xi} \|u(\xi) - u_{n-1}(\xi)\|_V, \tag{2}$$

where u_{n-1} stands for an approximation of $u(\xi)$ in V_{n-1}. However, this ideal algorithm is still unfeasible in practice, at least for the two following reasons:

1. computing the error $\|u(\xi) - u_{n-1}(\xi)\|_V$ for all $\xi \in \Xi$ may be unfeasible in practice (e.g., when $u(\xi)$ is only given by some parameter-dependent equation), and
2. maximizing this error over Ξ is a non trivial optimization problem.

Point 1) is usually tackled by selecting a parameter ξ_n which maximizes some surrogate error indicator $\Delta(u_{n-1}(\xi), \xi)$ that can be easily estimated. Assuming $u_n(\xi)$ is a quasi-optimal projection of $u(\xi)$ onto V_n and assuming $\Delta(u_{n-1}(\xi), \xi)$ is equivalent to $\|u(\xi) - u_{n-1}(\xi)\|_V$, there exists $\gamma \in (0, 1]$ such that

$$\|u(\xi_n) - P_{V_{n-1}} u(\xi_n)\|_V \geq \gamma \sup_{\xi \in \Xi} \|u(\xi) - P_{V_{n-1}} u(\xi)\|_V, \tag{3}$$

which yields a weak-greedy algorithm. Quasi-optimality means that the approximation u_n of u in V_n satisfies

$$\|u(\xi) - u_n(\xi)\|_V \leq C\|u(\xi) - P_{V_n}u(\xi)\|_V \tag{4}$$

for some constant C independent from V_n and ξ. Although the generated sequence V_n is not optimal, it has been proven in [21–23] that the approximation error

$$\sigma_n(\mathcal{M})_V := \sup_{u \in \mathcal{M}} \|u - P_{V_n}u\|_V$$

has the same type of decay as the benchmark $d_n(\mathcal{M})_V$ for algebraic or exponential convergence.

Remark 1 In the case of parameter-dependent linear equation arising, e.g., from the discretization of some parameter-dependent linear PDE of the form $r(u(\xi), \xi) = 0$ with $u(\xi) \in V = \mathbb{R}^N$, the approximation $u_n(\xi)$ is typically obtained through some (Petrov-)Galerkin projection onto V_n, with a complexity depending on $n \ll N$. In such a context, a weak-greedy algorithm classically involves a certified residual-based error estimate $\Delta(u_n(\xi), \xi)$, that is an upper bound of the true error. However, for some applications, such an error estimate can be pessimistic (when the underlying discrete operator is badly conditioned) so that the generated sequence V_n is far from being optimal. A possible strategy to improve such an estimate is to consider a preconditioned residual [12, 14, 24]. In Section 4, we overcome this limitation by considering for $\Delta(u_n(\xi), \xi)$ the targeted square error norm $\|u(\xi) - u_n(\xi)\|_V^2$, which is evaluated using adaptive Monte Carlo estimations.

Point 2) is addressed by transforming the continuous optimization problem over Ξ into a discrete optimization over a finite subset $\tilde{\Xi} \subset \Xi$. Choosing the training set $\tilde{\Xi}$ is a delicate task. As pointed out in [2, Section 2], if $\tilde{\Xi}$ is an ε-net of Ξ, then a greedy algorithm for the approximation of the discrete solution manifold $\widetilde{\mathcal{M}} = \{u(\xi) : \xi \in \tilde{\Xi}\}$ generates a sequence of spaces that are able to achieve a precision in $O(\epsilon)$ with similar performance as the ideal greedy algorithm. However, the cardinality of $\tilde{\Xi}$ may be very large for a parameter set Ξ in a high-dimensional space $\mathbb{R}^p$ and when a low precision ε is required. In [2], the authors propose a greedy algorithm which uses different training sets randomly chosen at each step. Under suitable assumptions on the approximability of the solution map $\xi \mapsto u(\xi)$ by sparse polynomial expansions, training sets can be chosen of moderate size independent of the parametric dimension p.

To conclude this section, we give a practical deterministic (weak)-greedy algorithm that can be summarized as follows.

Algorithm 1 Deterministic greedy algorithm.

```
Let $\tilde{\Xi} \subset \Xi$ be a discrete training set and $V_0 = \{0\}$.
For $n \geq 1$ proceed as follows.
Step 1. Select
        $\xi_n \in \arg\max_{\xi \in \tilde{\Xi}} \Delta(u_{n-1}(\xi), \xi)$.
Step 2. Compute $u(\xi_n)$ and update $V_n = \text{span}\{u(\xi_1), \dots, u(\xi_n)\}$.
```

Usually, Algorithm 1 is stopped when $\Delta(u_n(\xi), \xi)$ is below some target precision $\varepsilon > 0$ or for a given dimension n.

3 A probabilistic greedy algorithm

In this section, we motivate and present a probabilistic variant of Algorithm 1. Such an algorithm relies on the concept of probably approximately correct (PAC) maximum. It is proven to be a weak-greedy algorithm with high probability.

As a starting point for our work, we assume that for any value ξ in Ξ, the error estimator required at each step of Algorithm 1 admits the following form

$$\Delta(u_n(\xi), \xi) := \|u(\xi) - u_n(\xi)\|_V^2 = \mathbb{E}(Z_n(\xi)), \tag{5}$$

where $Z_n(\xi)$ is some parameter-dependent real-valued random variable, defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Here, $\mathbb{E}(Z_n(\xi))$ is a probabilistic representation of the current square error $\|u(\xi) - u_n(\xi)\|_V^2$ depending on the targeted applications as discussed in what follows.

Example 1 (Estimate of the norm of approximation error) Suppose that $u(\xi)$ belongs to $V = L^2(D)$ the Lebesgue space of square-integrable functions defined on a bounded set $D \subset \mathbb{R}^d$. If $\Delta(u_n(\xi), \xi) = \|u(\xi) - u_n(\xi)\|_{L^2}^2$, then (5) holds with $Z_n(\xi) = |D||u(Y, \xi) - u_n(Y, \xi)|^2$ where $Y \sim \mathcal{U}(D)$ is a random variable with uniform distribution over D.

Example 2 (Greedy algorithm for control variate [4, 5]) Let us suppose that we want to compute an MC estimate of the expectation of a parameter-dependent family of random variables $u(\xi)$ belonging to a Hilbert space of centered second-order random variables. MC estimate is known to slowly converge with respect to the number of samples of $u(\xi)$. Variance reduction techniques based on control variates are usually used to improve MC estimates. In [5], the authors propose an RB paradigm to compute a control variate with a greedy algorithm of the form of Algorithm 1 where $\Delta(u_n(\xi), \xi) = \mathbb{E}(Z_n(\xi))$ with $Z_n(\xi) = |u(\xi) - u_n(\xi)|^2$ in (5).

3.1 Main algorithm

Solving the following optimization problem

$$\xi_n \in \arg\max_{\xi \in \tilde{\Xi}} \mathbb{E}(Z_{n-1}(\xi)) \tag{6}$$

is in general out of reach, since $\mathbb{E}(Z_{n-1}(\xi))$ is unknown a priori or too costly to compute. Then, we propose a greedy algorithm with an approximate solution of (6).

Algorithm 2 Probabilistic greedy algorithm.

```
Let $\tilde{\Xi} \subset \Xi$ be a discrete training set. Starting from $V_0 = \{0\}$, proceed, for $n \geq 1$, as follows.
Step 1. Select
            $\xi_n \in \mathcal{S}(Z_{n-1}(\xi), \tilde{\Xi})$.
Step 2. Compute $u(\xi_n)$ and update $V_n = \mathrm{span}\{u(\xi_1), \ldots, u(\xi_n)\}$.
```

The question is now how to choose properly the set of candidate parameter values $\mathcal{S}(Z_{n-1}(\xi), \tilde{\Xi})$? In view of numerical applications, a first practical and naive approach is to seek ξ_n maximizing the empirical mean, i.e.,

$$\mathcal{S}(Z_{n-1}(\xi), \tilde{\Xi}) := \arg\max_{\xi \in \tilde{\Xi}} \overline{Z_{n-1}(\xi)}_K$$

where $\overline{Z_{n-1}(\xi)}_K = \frac{1}{K} \sum_{i=1}^{K} (Z_{n-1}(\xi))_i$ with K i.i.d. copies of $Z_{n-1}(\xi)$. In the following, this algorithm will be called **MC-greedy**. Despite its simplicity, it is well known that such an estimate for the expectation suffers from low convergence with respect to the number of samples leading to possible high computational costs especially if $Z_n(\xi)$ is expensive to evaluate. Moreover, nothing ensures that the returned (random) parameter ξ_n is a (quasi-)optimum for (6), almost surely or at least with high probability.

Instead, the so-called *bandit algorithms* (see, e.g., monograph [25]) are good candidates to address (6). Here, we particularly focus on PAC bandit algorithms that for each n return a parameter value ξ_n which is a probably approximately correct (PAC) maximum in relative precision for $\mathbb{E}(Z_n(\xi))$ over $\tilde{\Xi}$ (see [15]). For a given $\varepsilon \in (0, 1)$ and probability $\lambda_n \in (0, 1)$, letting $\xi_n^\star \in \arg\max_{\xi \in \tilde{\Xi}} \mathbb{E}[Z_{n-1}(\xi)]$, such an algorithm returns ξ_n satisfying

$$\mathbb{P}\left(\mathbb{E}(Z_{n-1}(\xi_n^\star)) - \mathbb{E}(Z_{n-1}(\xi_n)) \leq \varepsilon \mathbb{E}(Z_{n-1}(\xi_n^\star))\right) \geq 1 - \lambda_n. \tag{7}$$

We use the notation $\mathcal{S}(Z_{n-1}(\xi), \tilde{\Xi}) := \mathrm{PAC}_{\lambda_n, \varepsilon}(Z_{n-1}, \tilde{\Xi})$ when ξ_n satisfies (7). The resulting greedy algorithm is called **PAC-greedy**. In practice, the adaptive bandit algorithm in relative precision introduced in [15, Section 3.2] is particularly interesting in the case where $Z_n(\xi)$ is costly to evaluate since it preferentially samples the random variable $Z_n(\xi)$ for the parameter values for which it is more likely to find a maximum. Hence, it outperforms the mean complexity of a naive approach in terms of number of generated samples. Appendix A gives a detailed presentation of such a PAC adaptive bandit algorithm. As stated in Proposition 4, such an algorithm provides a PAC maximum in relative precision, that fulfills (7), in the particular case where $\{Z_n(\xi), \xi \in \tilde{\Xi}\}$ are random variables satisfying some concentration inequality. The interested reader can refer to [15] and included references for more details.

3.2 Analysis of PAC-greedy algorithm

Now, we propose and analyze a probabilistic greedy algorithm where the parameter ξ_n is a PAC maximum in relative precision for (5), i.e., satisfying (7), at each step n.

At a step n of the Algorithm 2, the reduced space $V_n = \operatorname{span}\{u(\xi_1), \ldots, u(\xi_n)\}$, as well as the approximation $u_n(\xi)$ are no longer deterministic. Indeed, they are related to the selected parameters $\xi_1, \ldots, \xi_n$ depending themselves on the errors at the previous steps through i.i.d. samples of the random variables $Z_i(\xi)$ for all $\xi \in \Xi$ and $i < n$ (required during PAC selection of ξ_n). Now, we prove that Algorithm 2 is a weak-greedy algorithm with high probability.

Theorem 1 *Take $(\lambda_n)_{n\geq 1} \subset (0,1)^{\mathbb{N}^+}$ such that $\sum_{n\geq 1} \lambda_n = \lambda < 1$, $\varepsilon \in (0,1)$ and $\tilde{\Xi} \subset \Xi$ a discrete training set. Moreover, suppose that for $n \geq 1$ the approximation $u_n(\xi)$ of $u(\xi)$ in V_n is quasi-optimal in the sense that it satisfies (3) with Ξ replaced by $\tilde{\Xi}$. Then, Algorithm 2 is a weak-greedy algorithm of parameter $\frac{\sqrt{1-\varepsilon}}{C}$, with probability at least $1-\lambda$, i.e.,*

$$\mathbb{P}\left(\|u(\xi_n) - P_{V_{n-1}} u(\xi_n)\|_V \geq \frac{\sqrt{1-\varepsilon}}{C} \max_{\xi \in \tilde{\Xi}} \|u(\xi) - P_{V_{n-1}} u(\xi)\|_V, \forall n \geq 1 \right) \geq 1-\lambda. \tag{8}$$

Proof Let first introduce some useful notation. We denote by $\mathbb{P}^{n-1}(\cdot) := \mathbb{P}(\cdot | Z_{<n})$ the conditional probability measure with respect to $Z_{<n}$ that denotes the collection of random variables $Z_i(\xi)_k$ for all $\xi \in \Xi$ and $i < n$, where $Z_i(\xi)_k$ are i.i.d. copies of $Z_i(\xi)$. The related conditional expectation is $\mathbb{E}^{n-1}(\cdot) = \mathbb{E}(\cdot | Z_{<n})$. Now, let $A = \cap_{n\geq 1} A_n$, each event A_n being defined as

$$A_n := \left\{ \mathbb{E}^{n-1}(Z_{n-1}(\xi_n^\star)) - \mathbb{E}^{n-1}(Z_{n-1}(\xi_n)) \leq \varepsilon \mathbb{E}^{n-1}(Z_{n-1}(\xi_n^\star)) \right\},$$

with $\xi_n^\star \in \arg\max_{\xi \in \tilde{\Xi}} \mathbb{E}^{n-1}(Z_{n-1}(\xi))$. Then, at each step n of Algorithm 2, the parameter ξ_n is a PAC maximum knowing $Z_{<n}$, i.e.,

$$\mathbb{P}^{n-1}(A_n) \geq 1 - \lambda_n. \tag{9}$$

Finally, as $u_{n-1}(\xi)$ is completely determined by all the steps before n (i.e., depending only on $Z_{<n}$), we have

$$\|u(\xi) - u_{n-1}(\xi)\|_V^2 = \mathbb{E}^{n-1}(Z_{n-1}(\xi)). \tag{10}$$

For all $n \geq 1$, the quasi-optimality condition (4) and probabilistic representation (10) lead to

$$\|u(\xi_n) - P_{V_{n-1}} u(\xi_n)\|_V^2 \geq \frac{1}{C^2} \|u(\xi_n) - u_{n-1}(\xi_n)\|_V^2 = \frac{1}{C^2} \mathbb{E}^{n-1}(Z_{n-1}(\xi_n)). \tag{11}$$

Moreover, if A holds, we have for $n \geq 1$

$$\mathbb{E}^{n-1}(Z_{n-1}(\xi)) \geq (1-\varepsilon)\mathbb{E}^{n-1}(Z_{n-1}(\xi_n^\star)) \geq (1-\varepsilon)\max_{\xi\in\tilde{\Xi}} \|u(\xi)-P_{V_{n-1}}u(\xi)\|_V^2 \quad (12)$$

by definition of $\xi_n^\star$. Thus, by combining (11) and (12), we have that A implies for all $n \geq 1$

$$\|u(\xi_n) - P_{V_{n-1}}u(\xi_n)\|_V \geq \frac{\sqrt{1-\varepsilon}}{C} \max_{\xi\in\tilde{\Xi}} \|u(\xi) - P_{V_{n-1}}u(\xi)\|_V .$$

We now estimate $\mathbb{P}(A)$

$$\begin{aligned}\mathbb{P}(A) &= 1 - \mathbb{P}(\overline{A}) \geq 1 - \sum_{n\geq 1} \mathbb{P}(\overline{A_n}) = 1 - \sum_{n\geq 1} \mathbb{E}\left(\mathbb{1}_{\overline{A_n}}\right) \\ &= 1 - \sum_{n\geq 1} \mathbb{E}\left(\underbrace{\mathbb{E}\left(\mathbb{1}_{\overline{A_n}}|Z_{<n}\right)}_{\mathbb{P}^{n-1}(\overline{A_n})}\right) \geq 1 - \sum_{n\geq 1} \lambda_n,\end{aligned}$$

where the last inequality derives from (9), which concludes the proof. □

Remark 2 Theorem 1 proves that Algorithm 2 is a weak-greedy algorithm, with probability $1 - \lambda$, for the approximation of the discrete solution manifold $\widetilde{\mathcal{M}}$. Thus, the approximation error $\sigma_n(\widetilde{\mathcal{M}})$ has the same decay rate as $d_n(\widetilde{\mathcal{M}})$ for algebraic or exponential convergence. In the lines of [2], it is possible to consider also a fully probabilistic variant of Algorithm 2, in which a training set Ξ_n randomly chosen is used at each step n of Algorithm 2 instead of $\tilde{\Xi}$. For a particular class of functions that can be approximated by polynomials with a certain algebraic rate, it can be proven that, for a suitable chosen size of random training set Ξ_n, the resulting algorithm is a weak-greedy algorithm with high probability with respect to the continuous solution manifold $\mathcal{M}$.

4 Reduced basis method for parameter-dependent PDEs with probabilistic interpretation

We recall that this work is motivated by the approximation, in a reduced basis framework, of a costly function $u(\xi) : D \to \mathbb{R}$ defined on the spatial domain $D \subset \mathbb{R}^d$ depending on the parameters ξ lying in $\Xi \subset \mathbb{R}^p$. Here, we consider the problem where u is the solution of a parameter-dependent PDE with probabilistic interpretation.

Let D be an open bounded domain in $\mathbb{R}^d$. For any parameter $\xi \in \Xi$, we seek $u(\xi) : \overline{D} \to \mathbb{R}$ the solution of the following boundary value problem,

$$\begin{aligned} -\mathcal{A}(\xi)u(\xi) &= g(\xi) \quad \text{in} \quad D, \\ u(\xi) &= f(\xi) \quad \text{on} \quad \partial D, \end{aligned} \qquad (13)$$

where $f(\xi) : \partial D \to \mathbb{R}$, $g(\xi) : \overline{D} \to \mathbb{R}$ are respectively the boundary condition and source term, and $\mathcal{A}(\xi)$ is a linear and elliptic partial differential operator.

Since the exact solution of (13) is not computable in general, it is classical to consider instead $u_h(\xi)$ an approximation in some finite dimensional space $V_h \subset V$ deduced from some numerical discretization of the PDE. Classical RBM applies in that context, relying on some variational principles to provide an approximation $u_n(\xi)$ of $u_h(\xi)$ in a reduced space $V_n \subset V_h$, of small dimension, obtained through a greedy algorithm (see Section 2). Here, we overcome such a priori discretization of the PDE and directly address the approximation of the solution $u(\xi)$ of (13). The key idea is to use the so-called *Feynman-Kac representation formula* that allows to compute pointwise estimates of $u(\xi)$ for any $x \in \bar{D}$. This particular framework raises the following practical questions. During the offline step, how to choose a computable error estimator $\Delta(u_n(\xi), \xi)$ required in greedy algorithm and compute the snapshots required for generating the reduced basis and related reduced space V_n? During the online step, how to compute the approximation u_n?

To that goal, in this section, a probabilistic RBM using only (noisy) pointwise evaluations is presented. We first recall the Feynman-Kac formula in Section 4.1. In Section 4.2, we detail a probabilistic greedy algorithm for the construction of the reduced space V_n in this setting. Finally, in Section 4.3, we discuss possible approaches for computing the approximation $u_n(\xi)$.

4.1 Feynman-Kac representation formula for an elliptic PDE

In what follows, $W = (W_t)_{t\geq 0}$ denotes a standard d-dimensional Brownian motion defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ endowed with its natural filtration $(\mathcal{F}_t)_{t\geq 0}$. For the sake of simplicity, the dependence to parameter ξ is omitted in the presentation of the Feynman-Kac formula.

Let us consider the boundary problem (13), where the partial differential operator $\mathcal{A}$ is given as

$$\mathcal{A} = \frac{1}{2}\sum_{i,j=1}^{d} (\sigma\sigma^T)_{ij}\frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{d} b_i \frac{\partial}{\partial x_i}. \tag{14}$$

It is the *infinitesimal generator* associated to the parameter-dependent d-dimensional diffusion process $X^x = (X_t^x)_{t\geq 0}$, adapted to $(\mathcal{F}_t)_{t\geq 0}$, solution of the following stochastic differential equation (SDE)

$$dX_t^x = b(X_t^x)dt + \sigma(X_t^x)dW_t, \quad X_0^x = x \in \overline{D}, \tag{15}$$

where $b(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ are the drift and diffusion coefficients, respectively.

Before recalling Feynman-Kac formula, we introduce additional assumptions and notation. Denoting by $\|\cdot\|$ both euclidean norm on $\mathbb{R}^d$ and Frobenius norm on $\mathbb{R}^{d\times d}$, we first introduce the assumption that b and σ are Lipschitz continuous.

Assumption 1 There exists a constant $0 < M < +\infty$ such that for all $x, y \in \bar{D}$, we have

$$\|b(x) - b(y)\| + \|\sigma(x) - \sigma(y)\| \leq M\|x - y\|. \tag{16}$$

Under Assumption 1, there exists a unique *strong* solution to Equation (15) (see, e.g., [26, Chapter 5, Theorem 1.1.]).

Denoting $a = \sigma\sigma^T$, we introduce the following uniform ellipticity assumption.

Assumption 2 There exists $c > 0$ such that

$$y^T a(x) y \geq c\|y\|^2, \quad \text{for all } y \in \mathbb{R}^d,\ x \in \overline{D}.$$

As problem (13) is defined on a bounded domain, we define the *first exit time* of D for the process X^x as

$$\tau^x = \inf\left\{s > 0 \ : \ X_s^x \notin D\right\}. \tag{17}$$

Also, we assume some regularity property on the spatial domain $\overline{D}$ and data.

Assumption 3 The domain D is an open connected bounded domain of $\mathbb{R}^d$, regular in the sense that it satisfies

$$\mathbb{P}(\tau^x = 0) = 1, \quad x \in \partial D.$$

Assumption 4 We assume that f is continuous on ∂D, g is Hölder-continuous on $\overline{D}$.

The following probabilistic representation theorem [26, Chapter 6, Theorem 2.4] holds.

Theorem 2 (Feynman-Kac formula *Under* Assumptions 1-4 *there exists a unique solution of* (13) *in* $\mathcal{C}(\overline{D}) \cap \mathcal{C}^2(D)$*, which satisfies for all* $x \in \overline{D}$

$$u(x) = \mathbb{E}\left(f(X_{\tau^x}^x) + \int_0^{\tau^x} g(X_t^x)dt\right) =: \mathbb{E}(F(x, X^x)), \tag{18}$$

where X^x *is the stopped diffusion process solution of* (15).

Theorem 2 allows to derive a probabilistic numerical method for the computation of pointwise MC estimate of u, see Appendix B.

4.2 Offline step

During the offline step, the probabilistic greedy algorithm presented in Section 3 is considered to construct the reduced space V_n. The keystone of such an algorithm is the probabilistic reinterpretation of the error estimate $\Delta(u_n(\xi), \xi)$ as in Equation (5). Using the Feynman-Kac representation formula, we show in Section 4.2.1 that it is possible to rewrite the square of the approximation error as an expectation. Then, in Section 4.2.2, we discuss possible strategies for the practical implementation of such an algorithm.

4.2.1 Probabilistic error estimate

Let us assume that $u_n(\xi)$ is a linear approximation of $u(\xi)$ in a given reduced space $V_n \subset V$ (e.g., obtained using Algorithm 2). We recall that $u(\xi) \in \mathcal{C}(\overline{D}) \cap \mathcal{C}^2(D)$ is the unique solution of (13) with the following probabilistic representation

$$u(x,\xi) = \mathbb{E}\left(F(x, X^{x,\xi}, \xi)\right), \quad x \in \bar{D}, \tag{19}$$

with $X^{x,\xi}$ the parameter-dependent stopped diffusion process solution of (15). In classical RB methods, the error estimate $\Delta(u_n(\xi), \xi)$ used in Algorithm 1 is usually related to some suitable norm of the equation residual. Here, we follow another path by considering the L^2-norm of the current approximation error $e_n(\xi) = u(\xi) - u_n(\xi)$, i.e.,

$$\Delta(u_n(\xi), \xi) = \|e_n(\xi)\|_{L^2}^2.$$

In what follows, we give a possible probabilistic reinterpretation of this error. Assuming that $u_n(\xi)$ is regular enough (the regularity being inherited from the snapshots), the error $e_n(\xi) := u(\xi) - u_n(\xi)$ is the unique solution, for all ξ in Ξ, of

$$\begin{aligned} -\mathcal{A}(\xi)e_n(\xi) &= g_n(\xi) \quad \text{on} \quad D, \\ e_n(\xi) &= f_n(\xi) \quad \text{on} \quad \partial D, \end{aligned} \tag{20}$$

where $f_n(\xi) := f(\xi) - u_n(\xi)$ and $g_n(\xi) = g(\xi) + \mathcal{A}(\xi)u_n(\xi)$. By Feynman-Kac representation theorem, for all ξ in Ξ, $e_n(\xi)$ is the unique solution of (20) in $\mathcal{C}(\overline{D}) \cap \mathcal{C}^2(D)$ and satisfies for all $x \in \overline{D}$

$$e_n(x,\xi) = \mathbb{E}\left(f_n(X^{x,\xi}_{\tau^{x,\xi}}, \xi) + \int_0^{\tau^{x,\xi}} g_n(X^{x,\xi}_t, \xi)dt\right) =: \mathbb{E}\left(F_n(x, X^{x,\xi}, \xi)\right), \tag{21}$$

with $X^{x,\xi}$ the stopped diffusion process solution of (15). Then, we have the following probabilistic reinterpretation for $\|e_n(\xi)\|_{L^2}^2$.

Theorem 3 *Let $Y \sim U(D)$ be uniformly distributed on D. Let W and $\tilde{W}$ be two independent standard d-dimensional Brownian motions defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and independent of Y. For any $x \in \overline{D}$, let $X^{x,\xi}$ and $\tilde{X}^{x,\xi}$ be solutions of* (15) *with W, $\tilde{W}$ respectively. Then, we have for any ξ in Ξ*

$$\|e_n(\xi)\|_{L^2}^2 = |D|\mathbb{E}\left(Z_n(\xi)\right), \tag{22}$$

with $Z_n(\xi) = F_n(Y, X^{Y,\xi}, \xi)F_n(Y, \tilde{X}^{Y,\xi}, \xi)$ and $|D|$ the Lebesgue measure of D.

Proof We first recall

$$\|e_n(\xi)\|_{L^2}^2 = \int_D e_n(x,\xi)^2 dx = |D|\mathbb{E}\left(e_n(Y,\xi)^2\right).$$

Since, for any x, $X^{x,\xi}$ and $\tilde{X}^{x,\xi}$ are i.i.d. random processes, we have

$$\mathbb{E}\left(e_n(Y,\xi)^2\right) = \mathbb{E}\left(\mathbb{E}\left(F_n(Y, X^{Y,\xi}, \xi))|Y\right)^2\right)$$

$$= \mathbb{E}\left(\mathbb{E}\left(F_n(Y, X^{Y,\xi}, \xi)|Y\right)\mathbb{E}\left(F_n(Y, \tilde{X}^{Y,\xi}, \xi)|Y\right)\right),$$

and

$$\|e_n(\xi)\|_{L^2}^2 = |D|\mathbb{E}\left(\mathbb{E}\left(F_n(Y, X^{Y,\xi}, \xi)F_n(Y, \tilde{X}^{Y,\xi}, \xi)|Y\right)\right).$$

Then, by the law of iterated expectation, we obtain (22). □

Remark 3 Assuming the existence of probabilistic representations for the gradient of $u(\xi)$ and $u_n(\xi)$, it would be possible to consider probabilistic interpretation of other norms of the approximation error, such as the H^1-norm. Such probabilistic representations have been derived in simple cases, see, e.g., [27, Corollary IV.5.2].

4.2.2 A probabilistic greedy algorithm using pointwise evaluations

For the purpose of numerical applications, we can apply Algorithm 2 together with the error estimate (22) for the construction of the reduced space V_n.

Sample computation
The samples of $Z_n(\xi)$ (as defined in Theorem 3) are generated from the functional F_n and independent trajectories of the discrete diffusion process $X^{Y,\Delta t}$, with $Y \sim \mathcal{U}(D)$. The discrete diffusion process $X^{Y,\Delta t}$ is computed using a suitable time integration scheme (see Appendix B) for the stochastic ODE (15).

Snapshot computation
Within Algorithm 2, the reduced space corresponds to $V_n = \text{span}\{u(\xi_1), \dots, u(\xi_n)\}$. However, the snapshots $\{u(\xi_1), \dots, u(\xi_n)\}$ are generally not available since it requires to compute the exact solution of (13) for parameter instances $\{\xi_1, \dots, \xi_n\}$. From Feynman-Kac formula (18), it is possible to compute MC estimates $u_{\Delta t,M}(x, \xi)$ of $u(x, \xi)$ from independent realizations of the diffusion process $X^{x,\Delta t}$ starting from $x \in \bar{D}$ (as detailed in Appendix B). Then, a global numerical approximation can be computed in some finite-dimensional linear space of dimension N (potentially much larger than n), e.g., by interpolation or least-square method, from these MC pointwise estimates. To compensate possible slow convergence of MC estimates, one can consider a sequential approach which uses the approximation error at each step as a control variate in order to reduce the variance of MC estimates. Such a strategy has been initially proposed in [28, 29] for interpolation, and recently extended for high-dimensional problems in [18].

Projection computation
For given ξ, the approximation $u_n(\xi)$ in V_n can be computed by interpolation or a least-square projection using MC estimates $u_{\Delta t,M}(x, \xi)$ given by (B10), with suitable choice for evaluation points in D (e.g., using magic points for interpolation [30], or optimal sampling for least-squares [31]). The resulting complexity of this projection step is only linear in n (up to log).

4.3 Online step

Given the reduced space V_n obtained during the offline stage, the approximation $u_n(\xi)$ is computed, with complexity depending only on n (and not on N), following the procedure described in the projection step of Section 4.2.2.

5 Numerical applications

The aim of this section is twofold. We first illustrate the feasibility of a greedy algorithm with probabilistic error estimate for the approximation of a parameter-dependent function from its pointwise evaluations. Then, we present some numerical experiments concerning the probabilistic RBM, discussed in Section 4, for the solution of parameter-dependent PDEs with probabilistic interpretation.

5.1 Approximation of parameter-dependent functions

Let us consider the problem of computing an approximation $u_n(\xi)$ of $u(\xi)$, from its pointwise evaluations at given points in D. Particularly, we seek $u_n(\xi)$ as the interpolation of $u(\xi)$ in the finite-dimensional space $V_n \subset V$, such that

$$u_n(x_i, \xi) = u(x_i, \xi), \;\; x_i \in \Gamma,$$

with $\Gamma = \{x_1, \ldots, x_n\}$ an unisolvant grid of suitably chosen interpolation points in D. Numerical experiments with a least-square projection provided similar results. Thus, they are not presented in this section.

5.1.1 Procedures for the construction of V_n

For constructing the space $V_n = \text{span}\{u(\xi_1), \ldots, u(\xi_n)\}$, we compare different greedy procedures for the selection of the snapshots $u(\xi_i)$, $i = 1, \ldots, n$. First, we use the deterministic greedy Algorithm 1, for which $\Delta(u_{n-1}(\xi), \xi)$ is a numerical estimate of the L^2-norm of the approximation error $\|u(\xi) - u_n(\xi)\|_{L^2(D)}$ using some integration rule. This approach is confronted to probabilistic alternatives relying on probabilistic reinterpretation of the approximation error

$$\|u(\xi) - u_n(\xi)\|^2_{L^2(D)} = \mathbb{E}(Z_n(\xi)).$$

where $Z_n(\xi) = |D||u_n(X, \xi) - u(X, \xi)|^2$, $X \sim \mathcal{U}(D)$, as discussed in Example 1. In this setting, the set $\mathcal{S}(Z_{n-1}(\xi), \tilde{\Xi})$ within Algorithm 2 is obtained using either a crude MC estimate of the expectation or adaptive bandit algorithms discussed in Appendix A. When non-asymptotic concentration inequalities are used, the parameter ξ_n returned by Algorithm 2 is a PAC maximum under suitable assumptions on the distribution of $Z_n(\xi)$. In particular, for any $\xi \in \tilde{\Xi}$, if there exist $a_n(\xi), b_n(\xi) \in \mathbb{R}$ such that $a_n(\xi) \leq Z_n(\xi) \leq b_n(\xi)$ a.s., concentration inequalities under the form (A2) hold.

Having such knowledge a priori of the distribution of $Z_n(\xi)$ and finding the bounds $a_n(\xi), b_n(\xi)$ is not an easy task. Here, since $u(\xi)$ is known, we set the following heuristic bounds

$$a_n(\xi) = \min_{x \in \tilde{D}} |u(x, \xi) - u_n(x, \xi)|^2 \text{ and } b_n(\xi) = \max_{x \in \tilde{D}} |u(x, \xi) - u_n(x, \xi)|^2,$$

to perform our computations with $\tilde{D} \subset D$ a finite subset. Moreover, by Remark 5, we have to define the sequence $(d_m)_{m \geq 1}$ with $d_m = \frac{\lambda_n}{\#\tilde{\Xi}} \frac{(p-1)}{p} m^{-p}$, $p = 2$ and $\lambda_n = \frac{\lambda}{n}$. We also consider a variant relying on asymptotic concentration inequality as a central limit theorem (CLT), which overcomes the necessity of computing any bound for $Z_n(\xi)$ and defining the sequence $(d_m)_{m \geq 1}$. These probabilistic approaches are also compared to another naive approach, in which ξ_n is chosen at random in $\tilde{\Xi}$ (without replacement) at each step n of Algorithm 2.

In what follows, the deterministic approach is called **D-greedy**, whereas the probabilistic ones using MC estimate and bandit algorithms are named **MC-greedy** and **PAC-greedy** relying on non-asymptotic (**Bounded**) or asymptotic concentration inequalities (**CLT**). The last one is simply referred to as **Random**.

5.1.2 Numerical setting

We perform some numerical tests with the methods discussed in the previous section for the approximation of the two subsequent functions

$$u(x, \xi) = 10x \sin(2\pi x \xi), \ \ (x, \xi) \in [0, 1] \times [2, 4]$$

and, following [5],

$$u(x, \xi) = \sqrt{x + 0.1}\mathbb{1}_{[0,\xi]}(x) + \left(\frac{x - \xi}{2\sqrt{\xi + 0.1}} + \sqrt{\xi + 0.1}\right)\mathbb{1}_{[\xi,1]}(x), \ \ (x, \xi) \in [0, 1]^2.$$

The test case related to each function will be designated by **(TC1)** and **(TC2)**, respectively.

For the numerical experiments, the training set $\tilde{\Xi}$ is obtained using $\#\tilde{\Xi} = 300$ equally spaced points in Ξ, similarly for $\tilde{D}$ made from equally spaced points in D (10000 for (TC1), and 1000 for (TC2)). Then, the L^2-norm of the approximation error is estimated by the trapezium rule. Both deterministic and probabilistic greedy algorithms are stopped for given $n = 20$ for (TC1) and $n = 30$ for (TC2). The interpolation grid Γ is set to be the sequence of magic points [30], with respect to the basis of V_n. For the probabilistic procedure with naive MC estimate, we set $K \in \{1, 50\}$. Finally, for bandit algorithms, the stopping criterion is $\varepsilon = 0.9$ and $\lambda = 0.1$. In [15, Section 4], it was observed that λ has little influence on the number of samples $m_n(\xi)$ used by adaptive algorithm. However, an open problem is to find a λ that gives an optimal compromise between a high probability of returning a PCA maximum and a small sampling complexity.

5.1.3 Numerical results

Let us first study the quality of the approximations provided by the different approaches. Figures 1 and 2 represent the evolution of the estimated expectation $\mathbb{E}_\xi$ and maximum, with respect to ξ, of the approximation error $\|u_n(\xi) - u(\xi)\|_{L^2(\tilde{D})}$ for (TC1) and (TC2). These estimates have been computed using 100 independent realizations of $u_n(\xi)$ obtained from uniform draws of ξ in Ξ. In that case, only one realization of the probabilistic algorithms is performed for the comparison. For both test cases, D-greedy, MC-greedy, and PAC-greedy procedures behave similarly with the same error decay with respect to n reaching a precision around 10^{-14} for (TC1) and 10^{-5} for (TC2). Let us mention that for (TC2), the function to approximate has a slow decay of its Kolmogorov n-width (see, e.g., discussion in [5, Section 4.3.2]) which explains higher error for larger n. For the random approach, the selection of interpolation points is less optimal. For the first iterates, it behaves similarly to other approaches, but we observe that the approximation $u_n(\xi)$ is less accurate with n, from around 15 for (TC1) and 10 for (TC2) respectively. However, despite no guarantee of the optimality of the returned parameter ξ_n, the PAC algorithm with asymptotic concentration inequality and especially the MC-greedy algorithm, either with a single random evaluation of the error estimate ($K = 1$), lead to very satisfactory results with an error close to the deterministic interpolation approach for both test cases.

Greedy procedure

We now turn to the study of the greedy procedures used for the selection of the snapshots. Figures 3 and 4 represent the error estimate $\Delta(u_{n-1}(\xi), \xi)$ as well as the number of samples $m_n(\xi)$ required during the greedy selection of ξ_n for deterministic and probabilistic greedy algorithms based on bandit algorithms for (TC1) and (TC2). These curves correspond to one realization of the probabilistic algorithms. First, we observe that parameters selected (indicated with the symbol $*$ on the curves) by probabilistic algorithms do not necessarily coincide with the ones selected by D-greedy. For MC-greedy, we observe a much higher variability of the error relatively to parameter ξ, for $K = 1$. This is due to the high variance of the estimate. However, in this simple example, even a crude MC estimate with $K = 1$ allows to select a value of parameter

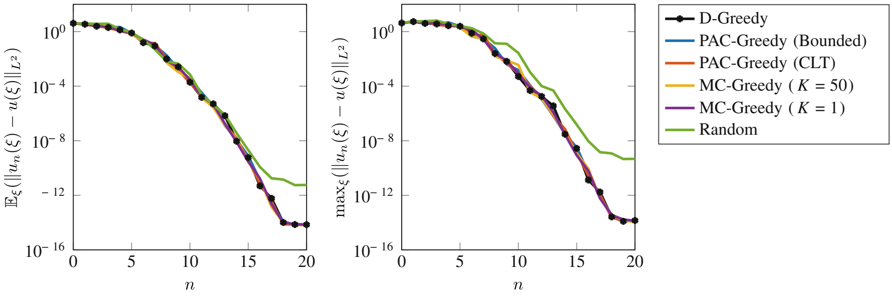

Fig. 1 (TC1) Evolution with respect to n, of the estimated expectation and maximum of the approximation error in L^2-norm, computed for 100 instances of ξ, for one realization of the probabilistic greedy algorithms compared to the deterministic one

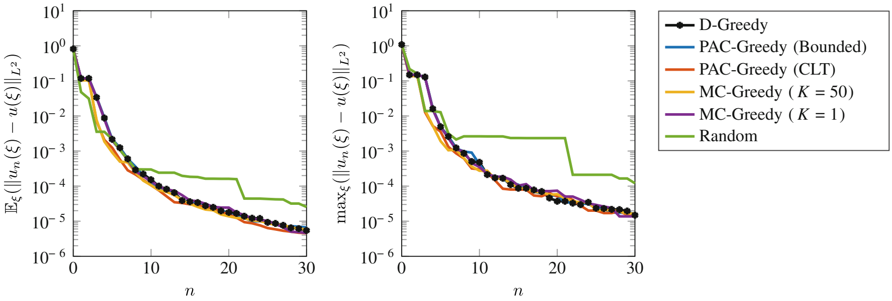

Fig. 2 (TC2) Evolution with respect to n, of the estimated expectation and maximum of the approximation error in L^2-norm, computed for 100 instances of ξ, for one realization of the probabilistic greedy algorithms compared to the deterministic one

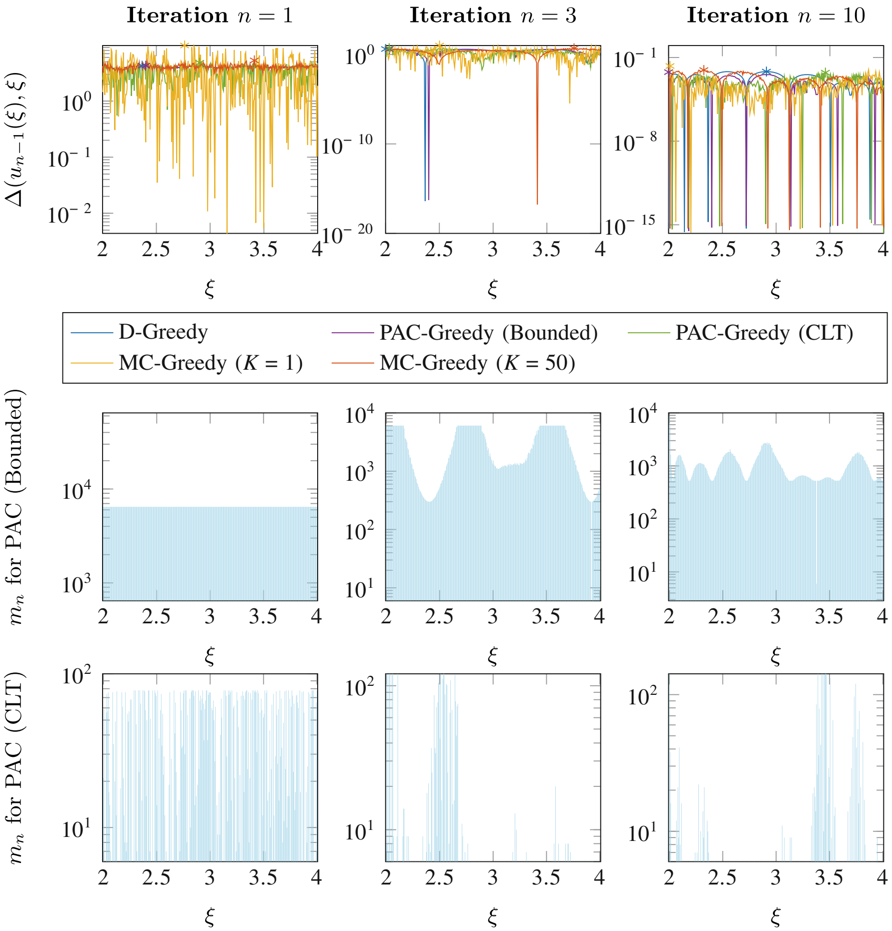

Fig. 3 (TC1) Evolution of the error during greedy procedures based on PAC bandit algorithms (top), together with required samples $m_n(\xi)$ for selecting $\xi_n \in \tilde{\Xi}$

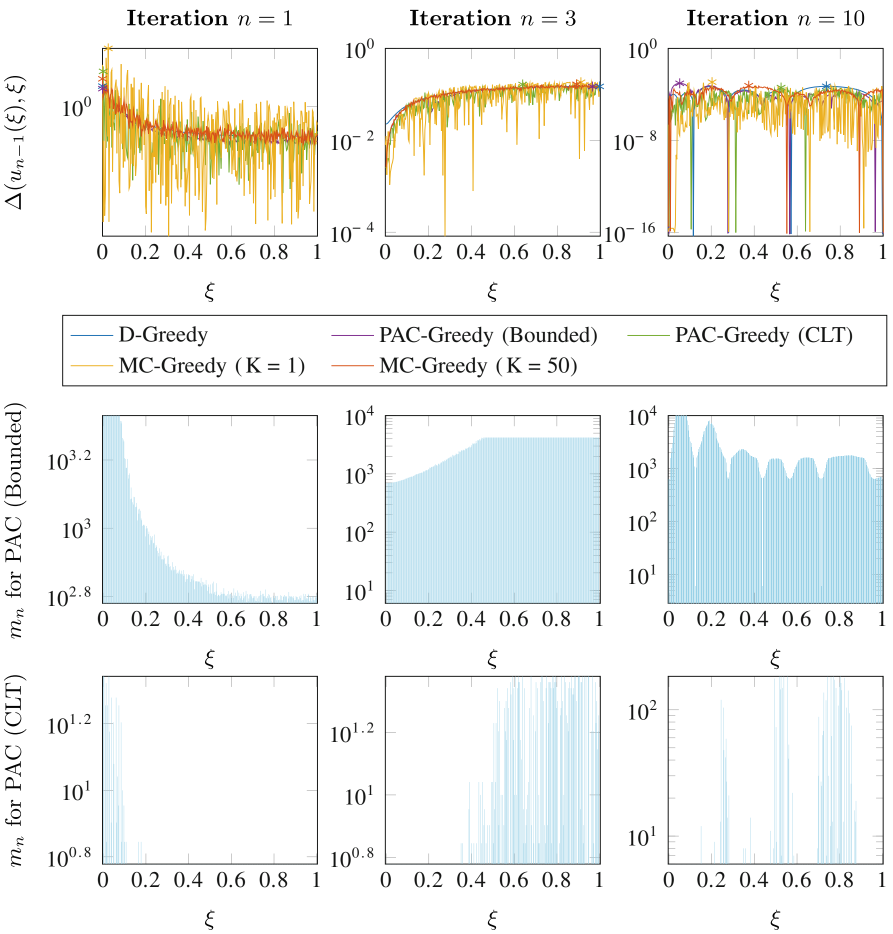

Fig. 4 (TC2) Evolution of the error during greedy procedures based on PAC bandit algorithms (top), together with required samples $m_n(\xi)$ for selecting $\xi_n \in \tilde{\Xi}$

that will make the error decrease significantly at the next iteration (see Figs. 1 and 2). Second, as expected the number of samples $m_n(\xi)$ is adapted for both algorithms resulting in higher sampling in the region where it is likely to find maximum. Globally, we observe that PAC-greedy (CLT) works quite similarly to PAC-greedy (Bounded). But the two approaches differ in terms of the required number of samples. Indeed, CLT-based approach only requires around a maximum of $10 - 10^2$ samples whereas the one based on concentration inequalities requires between $10^3 - 10^5$ samples.

Sampling complexity

Now, we briefly discuss the sampling complexities of the different methods, which are summarized in Table 1. From Figs. 1 and 2, we see that MC- and PAC-greedy yield roughly the same level of error for a given dimension n. Therefore, the table provides the cumulated number of evaluations of the function for constructing the reduced basis of a given dimension n for each test case.

Table 1 Cumulative number of samples required by each algorithm, for (TC1) and (TC2), to construct the reduced basis of dimension n

Method	(TC1) for $n = 20$	(TC2) for $n = 30$
D-greedy	6×10^7	9×10^6
MC-greedy $K = 1$	6×10^3	9×10^3
MC-greedy $K = 50$	3×10^5	4.5×10^5
PAC-greedy (Bounded)	1.284326×10^7	5.764507×10^7
PAC-greedy (CLT)	7.507800×10^4	2.333380×10^5

We present as a reference the results of D-greedy using a fine grid $\tilde{D}$ for numerical integration. The most costly probabilistic algorithm is PAC-greedy (Bounded). It requires a high number of samples for constructing non-asymptotic confidence intervals so that it does not outperform D-greedy. Let us mention that PAC-greedy (CLT) is quite competitive with MC-greedy with $K = 50$ and underlines the interest of using some adaptive procedure for snapshot selection. However, the most interesting trade-off between efficiency and accuracy is MC-greedy with $K = 1$.

In regard to these observations, in the next section, the MC-greedy approach with few samples K for the error estimation will be retained in practice to reduce computational costs.

5.2 Parameter-dependent PDE

Now, we focus on the solution of parameter-dependent PDEs, as introduced in Section 4. Given $\Xi = [0.005, 1]$, we seek $u(\xi), \xi \in \Xi$, solution on $D =]0, 1[$ of the following boundary problem

$$\begin{aligned} -\mathcal{A}(\xi)u(\xi) &:= -a(\xi)u''(\xi) - b(\xi)u'(\xi) = g(\xi) \text{ in }]0, 1[, \\ u(\xi) &= f(\xi) \text{ at } x \in \{0, 1\}, \end{aligned} \tag{23}$$

for given boundary values $f(\xi) : \{0, 1\} \to \mathbb{R}$ and source term $g(\xi) : [0, 1] \to \mathbb{R}$. Moreover, we denote $a(\xi) \in (0, +\infty)$ and $b(\xi) \in \mathbb{R}$ the diffusion and advection coefficients respectively. We assume that (23) admits a unique solution in $\mathcal{C}^2([0, 1])$ whose probabilistic representation is given by

$$u(x, \xi) = \mathbb{E}(F(x, X^{x,\xi}, \xi)) := \mathbb{E}\left(f(X^{x,\xi}_{\tau^{x,\xi}}) + \int_0^{\tau^{x,\xi}} g(X^{x,\xi}_t, \xi) dt \right).$$

The associated parameter-dependent stopped diffusion process $X^{x,\xi}$ is solution of

$$dX^{x,\xi}_t = b(\xi)dt + \sqrt{2a(\xi)}dW_t, \qquad X^{x,\xi}_0 = x. \tag{24}$$

In the following, we take $a(\xi) = \xi$ and $b(\xi) = -10$. The source term as well as Dirichlet boundary conditions are set such that the exact solution to Equation (23) is

$$u(x, \xi) = \frac{\exp(x/\xi) - 1}{\exp(1/\xi) - 1}. \tag{25}$$

5.2.1 Compared procedures

In what follows, we test the probabilistic RBM discussed in Section 4. Since the exact solution is known, we use it for the snapshots. The projection u_n is obtained from evaluations of the exact solution through interpolation (**Interp**) using magic points or a least-squares (**LS**) approximation using a set of points $\tilde{D}$ in D, with $\#\tilde{D} \geq n$. It is also compared to the minimal residual based (**MinRes**) approximation defined by

$$\begin{aligned} u_n(\xi) = \arg\min_{v \in V_n} \Big(& \sum_{x_i \in \tilde{D}} |\mathcal{A}(x_i, \xi) v(x_i, \xi) + g(x_i, \xi)|^2 \\ & + |u(0, \xi) - v(0, \xi)|^2 + |u(1, \xi) - v(1, \xi)|^2 \Big). \end{aligned} \tag{26}$$

During the offline stage, we consider different greedy algorithms for the generation of the reduced spaces. First, we perform the proposed probabilistic greedy Algorithm 2 for the construction of the reduced space V_n using MC estimates with $Z_n(\xi)$ defined as in Theorem 3. This approach is compared to an alternative RBM in a deterministic setting. Since $u(\xi)$ is implicitly known through the boundary value problem (23), the greedy selection of V_n is performed using Algorithm 1 where $\Delta(u_n(\xi), \xi)$ is an estimate (using trapezoidal integration rule) of the residual-based error estimate $\|\mathcal{A}(\xi) u_n(\xi) + g(\xi)\|^2_{L^2(D)}$ during the offline stage.

In the presented results, the residual-based greedy algorithm is referred to as **Residual**. The MC estimate using Feynman-Kac representation is named **FK-MC**. These approaches are also compared to a naive one named **Random** in which the parameters $\xi_1, \dots, \xi_n$ that define V_n are chosen at random.

5.2.2 Numerical results

For the numerical experiments, the training set $\tilde{\Xi}$ is made of 200 samples from a log uniform distribution over Ξ. This distribution is chosen as the solution strongly varies with respect to the viscosity ξ, in particular, we want to reach small viscosities. Realizations of $Z_n(\xi)$, given by (B10), are computed using $M = 500$ realizations of the approximate stochastic diffusion process solution of Equation (24), obtained by Euler-Maruyama scheme (see Appendix B) with $\Delta t = 10^{-3}$. For computing $u_n(\xi)$, magic points are used for interpolation while for LS and MinRes approaches, we choose for $\tilde{D}$ a regular grid of 100 points in D. Here, greedy algorithms are stopped when $n = 30$. The MC-FK greedy algorithm is performed for $K \in \{1, 10\}$.

Figure 5 represents the estimated expectation and maximum, with ξ, of the approximation error for the compared procedures, with respect to n. The obtained results

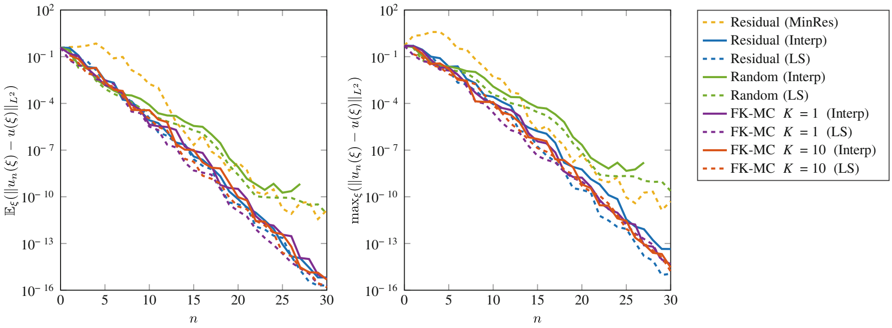

Fig. 5 Parameter-dependent equation: evolution with respect to n, of the estimated expectation and maximum of the approximation error in L^2-norm, computed for 100 instances of ξ, for one realization of the probabilistic greedy algorithms compared to the deterministic one

underline the importance of both projection and reduced basis construction. When MinRes method is used, the approximation is less accurate than for other deterministic approaches (dashed yellow curve). Especially, when it is compared to approaches using a residual-based error estimate (blue curves) with interpolation or LS approximation. In that case, the mean approximation error is of order 10^{-11} for MinRes against 10^{-15} for the latter (for the maximum error, we have 10^{-10} against 10^{-14}). Second, let us comment the impact of the probabilistic reduced basis selection. As for function approximation, picking at random the reduced basis for the considered problem is far from optimal since the error expectation tends to stagnate around 10^{-10} for $n \geq 20$ (around 10^{-9} for the maximum). However, when considering residual-based or FK-based error estimates (even with $K = 1$), with interpolation or least-square for projection, the error behaves quite similarly and reaches 10^{-15} for $n = 30$. This shows that the proposed probabilistic-based error procedure performs well in practice.

6 Conclusion

In this work, we have proposed a probabilistic greedy algorithm for the approximation of a family of parameter-dependent functions for which we only have access to (noisy) pointwise evaluations. It relies on an error indicator that can be written as an expectation of some parameter-dependent random variable. Different variants of this algorithm have been proposed using either naive MC estimates or PAC bandit algorithms, the latter leading to a weak-greedy algorithm with high probability. Several test cases have demonstrated the performances of the proposed procedures.

For parameter-dependent PDEs whose solution admits a probabilistic representation, through the Feynman-Kac formula, such an algorithm can be embedded within a probabilistic RBM using only (noisy) pointwise evaluations. Numerical results have shown the main relevance of considering Feynman-Kac error-based estimate for greedy basis selection. We have also illustrated the influence of the projection during the online and offline step. Indeed, we observed in Section 5.2.2 that interpolation or least-square projection (using pointwise evaluations of the solution) clearly outperform MinRes projection. Following the discussion of Section 4.2.2, using a sequential procedure as proposed in [18, 28, 29] should be an interesting alternative to avoid limitations of residual-based projections. However, further work should be conducted to provide a projection with controlled error and at a low cost, which is crucial for efficient model reduction.

The purpose of our simple numerical experiments was to illustrate the behavior of our probabilistic greedy algorithms. The application to more complex problems in higher dimensions will be the focus of future works.

Appendix A: Adaptive bandit algorithms

We present an adaptive bandit algorithm to find a PAC maximum in relative precision of $\mathbb{E}(Z(\xi))$ over the discrete set $\tilde{\Xi}$. Here $\{Z(\xi) : \xi \in \tilde{\Xi}\}$ is a finite collection of random variables satisfying $\mathbb{E}(Z(\xi)) \neq 0$, defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. After introducing some required notation, we present a practical adaptive bandit algorithm which returns a PAC maximum in relative precision for (6) when assuming suitable assumptions on the distribution of $Z(\xi)$.

A.1 Notations and assumptions

We denote by $\overline{Z(\xi)}_m$ the empirical mean of $Z(\xi)$ and $\overline{V(\xi)}_m$ its empirical variance, respectively defined by

$$\overline{Z(\xi)}_m = \frac{1}{m}\sum_{k=1}^{m} Z(\xi)_k \quad \text{and} \quad \overline{V(\xi)}_m = \frac{1}{m}\sum_{k=1}^{m}\left(Z(\xi)_k - \overline{Z(\xi)}_m\right)^2, \tag{A1}$$

where $Z(\xi)_1, \ldots, Z(\xi)_m$ are m independent copies of $Z(\xi)$. Moreover, the random variable $Z(\xi)$ is assumed to satisfy the following concentration inequality

$$\mathbb{P}\left(|\overline{Z(\xi)}_m - \mathbb{E}(Z(\xi))| \leq c(m, x, \xi)\right) \geq 1 - x, \tag{A2}$$

for each $\xi \in \tilde{\Xi}$, $0 \leq x \leq 1$ and $m \geq 1$. The bound $c(m, x, \xi)$ depends on the probability distribution of $Z(\xi)$.

Remark 4 In view of numerical experiments, we can consider standard concentration inequalities for sub-Gaussian or bounded random variables, see, e.g., [15, Section 2]. In particular, if for any $\xi \in \tilde{\Xi}$, there exists $a(\xi), b(\xi) \in \mathbb{R}$ such that almost surely, we have $a(\xi) \leq Z(\xi) \leq b(\xi)$, then (A2) holds with

$$c(m, x, \xi) = \sqrt{\frac{2\overline{V(\xi)}_m \log(3/x)}{m}} + \frac{3\,(b(\xi) - a(\xi))\log(3/x)}{m}.$$

An alternative to (A2) is to rely on asymptotic confidence intervals for $\mathbb{E}(Z(\xi))$ based on limit theorems of the form

$$\mathbb{P}\left(|\overline{Z(\xi)}_m - \mathbb{E}(Z(\xi))| \leq c(m, x, \xi)\right) \to 1 - x, \text{ as } m \to \infty. \tag{A3}$$

For example, for second-order random variables, the central limit theorem provides such a property with

$$c(m, x, \xi) = \gamma_x \sqrt{\frac{\overline{V(\xi)}_m}{m}},$$

and γ_x the $(x/2)$-quantile of the normal distribution.

A.2 Algorithm

Now, let us define a sequence $(d_m)_{m\geq 1} \subset (0, 1)^{\mathbb{N}}$, independent from ξ, and such that $\sum_{m\geq 1} d_m < \infty$. Then, we introduce $c_{\xi,m} = c(m, d_m, \xi)$, and $\beta^{\pm}_{\xi,m(\xi)} = \overline{Z(\xi)}_{m(\xi)} \pm c_{\xi,m(\xi)}$. Note that the concentration inequality (A2) implies that $[\beta^-_{m(\xi)}(\xi), \beta^+_{m(\xi)}(\xi)]$ is a confidence interval for $\mathbb{E}(Z(\xi))$ with level $1 - d_{m(\xi)}$.

Letting $s(\xi) := \text{sign}(\overline{Z(\xi)}_{m(\xi)})$ and $\varepsilon_{\xi,m(\xi)} = \begin{cases} \frac{c_{\xi,m(\xi)}}{|\overline{Z(\xi)}_{m(\xi)}|} & \text{if } \overline{Z(\xi)}_{m(\xi)} \neq 0, \\ +\infty & \text{otherwise.} \end{cases}$, we define the following estimate for $\mathbb{E}(Z(\xi))$ given by

$$\hat{\mathbb{E}}_{m(\xi)}(Z(\xi)) = \begin{cases} \overline{Z(\xi)}_{m(\xi)} - \varepsilon_{\xi,m(\xi)}\, s(\xi) c_{\xi,m(\xi)} & \text{if } \varepsilon_{\xi,m(\xi)} < 1, \\ \overline{Z(\xi)}_{m(\xi)} & \text{otherwise.} \end{cases} \tag{A4}$$

Then, the adaptive bandit algorithm proposed in [15] is as follows.

Algorithm 3 Adaptive bandit algorithm.

```
Let $\varepsilon, \lambda \in (0, 1)$ and $K \in \mathbb{N}$. Set $\ell = 0$, $\Xi_0 = \tilde{\Xi}$, $m(\xi) = K$ and $\varepsilon_{\xi,m(\xi)} = +\infty$ for all $\xi \in \Xi$.
while $\#\Xi_\ell > 1$ and $\max_{\xi\in\Xi_\ell} \varepsilon_{\xi,m(\xi)} > \frac{\varepsilon}{2+\varepsilon}$ do
    for all $\xi \in \Xi_\ell$ do
        Sample $Z(\xi)$, increment $m(\xi)$ and update $\varepsilon_{\xi,m(\xi)}$.
        Compute the estimate $\hat{\mathbb{E}}_{m(\xi)}(Z(\xi))$ using (A4).
    end for
    Increment $\ell$ and put in $\Xi_\ell$ every $\xi \in \tilde{\Xi}$ such that
        $\beta^+_{\xi,m(\xi)} \geq \max_{\nu\in\tilde{\Xi}} \beta^-_{\nu,m(\nu)}$.    (A5)
end while
Select $\hat{\xi}$ such that
    $\hat{\xi} \in \arg\max_{\xi\in\Xi_\ell} \hat{\mathbb{E}}_{m(\xi)}(Z(\xi))$.
```

At each step ℓ, the principle of Algorithm 3 is to successively increase the number of samples $m(\xi)$ of the random variables $Z(\xi)$ in the subset $\Xi_\ell \subset \tilde{\Xi}$, obtained using confidence intervals $[\beta^-_{m(\xi)}(\xi), \beta^+_{m(\xi)}(\xi)]$ of $\mathbb{E}(Z(\xi))$ according to (A5). The idea behind is to use those confidence intervals to find regions of $\tilde{\Xi}$ where one has a high chance to find a maximum. Then, $\hat{\xi}$ is returned as a maximizer over Ξ_ℓ of the expectation estimate defined by (A4). Under suitable assumptions, and when using certified non-asymptotic concentration inequalities, this algorithm returns a PAC maximum in relative precision of $\mathbb{E}(Z(\xi))$ over $\tilde{\Xi}$, as stated in [15, Proposition 3.2], recalled below.

Proposition 4 *Let $\varepsilon, \lambda \in (0, 1)$ and $(d_m)_{m\geq 1} \subset (0, 1)$ be a sequence that satisfies*

$$\sum_{m\geq 1} d_m \leq \frac{\lambda}{\#\tilde{\Xi}} \quad \textit{and} \quad \log(1/d_m)/m \underset{m\to+\infty}{\to} 0, \tag{A6}$$

and assume that c satisfies (A2). If for all ξ in Ξ, $Z(\xi)$ is a random variable with $\mathbb{E}(Z(\xi)) \neq 0$, then Algorithm 3 almost surely stops and returns a PAC maximum in relative precision, i.e., $\hat{\xi} = \mathrm{PAC}_{\lambda,\varepsilon}(Z, \tilde{\Xi})$.

Remark 5 In practice a possible choice for the sequence $(d_m)_{m\geq 1}$ is to take

$$d_m = \delta c m^{-p} \text{ with } \delta = \frac{\lambda}{\#\tilde{\Xi}} \text{ and } c = \frac{p-1}{p}, \tag{A7}$$

which satisfies (A6) for any $p > 1$.

In general, confidence intervals based on asymptotic theorems are much smaller than those obtained with non-asymptotic concentration inequalities and yield a selection of much smaller sets Ξ_ℓ of candidate maximizers, hence a much faster convergence of the algorithm. However, when using asymptotic theorems, we can not guarantee to obtain a PAC maximizer.

Appendix B. Probabilistic approximation of the solution of a PDE

Here, we discuss the numerical computation of an estimate of $u(x)$ for any $x \in \bar{D}$. To that goal, we use a suitable integration scheme to get an approximation of the diffusion process X^x and a MC method to evaluate the expectation in formula (18).

An approximation of the diffusion process is obtained using an Euler-Maruyama scheme. More precisely, setting $t_n = n\Delta t, n \in \mathbb{N}$, X^x is approximated by a piecewise constant process $X^{x,\Delta t}$, where $X^{x,\Delta t}_t = X^{x,\Delta t}_n$ for $t \in [t_n, t_{n+1}[$ and

$$\begin{aligned} X^{x,\Delta t}_{n+1} &= X^{x,\Delta t}_n + \Delta t\, b(X^{x,\Delta t}_n) + \sigma(X^{x,\Delta t}_n)\, \Delta W_n, \\ X^{x,\Delta t}_0 &= x, \end{aligned} \tag{B8}$$

where $\Delta W_n = W_{n+1} - W_n$ is an increment of the standard Brownian motion.

Numerical computation of $u(x)$ for all $x \in \bar{D}$ requires the computation of a stopped process $X^{x,\Delta t}$ at time $\tau^{x,\Delta t}$, an estimation of the first exit time of D. Here, we consider the simplest way to define this discrete exit time

$$\tau^{x,\Delta t} = \min \left\{ t_n > 0 \; : \; X_{t_n}^{x,\Delta t} \notin D \right\}. \tag{B9}$$

Such a discretization choice may lead to over-estimation of the exit time with an error in $O(\Delta t^{1/2})$. More sophisticated approaches are possible to improve the order of convergence, as Brownian bridge, boundary shifting or Walk On Sphere (WOS) methods, see, e.g., [27, Chapter 6]. These are not considered here.

Letting $\{X^{x,\Delta t}(\omega_m)\}_{m=1}^{M}$ be M independent samples of $X^{x,\Delta t}$, we obtain a MC estimate noted $u_{\Delta t,M}(x)$ for $u(x)$ defined as

$$u_{\Delta t,M}(x) = \frac{1}{M} \sum_{m=1}^{M} \left[f(X_{\tau^{x,\Delta t}}^{x,\Delta t}(\omega_m)) + \int_0^{\tau^{x,\Delta t}} g(X_t^{x,\Delta t}(\omega_m))dt \right]. \tag{B10}$$

Declarations

Competing of interests The authors declare no competing interests.

References

1. Nouy, A.: Low-rank tensor methods for model order reduction. SpringerLink, pp. 857–882 (2017). https://doi.org/10.1007/978-3-319-12385-1_21
2. Cohen, A., Dahmen, W., DeVore, R., Nichols, J.: Reduced basis greedy selection using random training sets. ESAIM: Math. Model. Numer. Anal. **54**(5), 1509–1524 (2020)
3. Cai, D., Yao, C., Liao, Q.: A stochastic discrete empirical interpolation approach for parameterized systems. Symmetry **14**(3), 556 (2022). https://doi.org/10.3390/sym14030556
4. Boyaval, S., Lelièvre, T.: A variance reduction method for parametrized stochastic differential equations using the reduced basis paradigm. Commun. Math. Sci. **8**(3), 735–762 (2010). https://doi.org/10.4310/CMS.2010.v8.n3.a7
5. Blel, M.-R., Ehrlacher, V., Lelièvre, T.: Influence of sampling on the convergence rates of greedy algorithms for parameter-dependent random variables. (2021). arXiv:2105.14091
6. Homescu, C., Petzold, L.R., Serban, R.: Error estimation for reduced-order models of dynamical systems. SIAM Rev. **49**(2), 277–299 (2007)
7. Janon, A., Nodet, M., Prieur, C., Prieur, C.: Goal-oriented error estimation for parameter-dependent nonlinear problems. ESAIM: Math. Model. Numer. Anal. **52**(2), 705–728 (2018)
8. Smetana, K., Zahm, O.: Randomized residual-based error estimators for the proper generalized decomposition approximation of parametrized problems. Int. J. Numer. Methods Eng. **121**(23), 5153–5177 (2020). https://doi.org/10.1002/nme.6339
9. Smetana, K., Zahm, O., Patera, A.T.: Randomized residual-based error estimators for parametrized equations. SIAM J. Sci. Comput. **41**(2), 900–926 (2019)
10. Balabanov, O., Nouy, A.: Randomized linear algebra for model reduction. Part I: Galerkin methods and error estimation. Adv. Comput. Math. **45**(5-6), 2969–3019 (2019). https://doi.org/10.1007/s10444-019-09725-6
11. Balabanov, O., Nouy, A.: Randomized linear algebra for model reduction. Part II: minimal residual methods and dictionary-based approximation. Adv. Comput. Math. **47**(2), 26–54 (2021). https://doi.org/10.1007/s10444-020-09836-5

12. Balabanov, O., Nouy, A.: Preconditioners for model order reduction by interpolation and random sketching of operators. arXiv (2021). https://doi.org/10.48550/ARXIV.2104.12177. https://arxiv.org/abs/2104.12177
13. Saibaba, A.K.: Randomized discrete empirical interpolation method for nonlinear model reduction. SIAM J. Sci, Comput (2020)
14. Zahm, O., Nouy, A.: Interpolation of inverse operators for preconditioning parameter-dependent equations. SIAM J. Scie. Comput. **38**(2), 1044–1074 (2016) http://dx.doi.org/10.1137/15M1019210. https://doi.org/10.1137/15M1019210
15. Billaud-Friess, M., Macherey, A., Nouy, A., Prieur, C.: A PAC algorithm in relative precision for bandit problem with costly sampling. Math. Methods Oper. Res. **96**(2), 161–185 (2022)
16. Bebendorf, M.: Approximation of boundary element matrices. Numer. Math. **86**(4), 565–589 (2000)
17. Tyrtyshnikov, E.: Incomplete cross approximation in the mosaic-skeleton method. Computing **64**(4), 367–380 (2000). https://doi.org/10.1007/s006070070031
18. Billaud-Friess, M., Macherey, A., Nouy, A., Prieur, C.: Stochastic methods for solving high-dimensional partial differential equations. In: International conference on monte Carlo and quasi-Monte Carlo methods in scientific computing, Springer, pp. 125–141 (2018)
19. Haasdonk, B.: 2. In: Benner, P., Cohen, A., Ohlberger, M., Willcox, K. (eds.) Reduced basis methods for parametrized PDEs–a tutorial introduction for stationary and instationary problems. SIAM, Philadelphia, PA (2017)
20. Nouy, A.: 4. In: Benner, P., Cohen, A., Ohlberger, M., Willcox, K. (eds.) Low-rank methods for high-dimensional approximation and model order reduction. SIAM, Philadelphia, PA (2017)
21. Binev, P., Cohen, A., Dahmen, W., DeVore, R., Petrova, G., Wojtaszczyk, P.: Convergence rates for greedy algorithms in reduced basis methods. SIAM J. Math, Anal (2011)
22. Buffa, A., Maday, Y., Patera, A.T., Prud'homme, C., Turinici, G.: A priori convergence of the greedy algorithm for the parametrized reduced basis method. ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique **46**(3), 595–603 (2012). https://doi.org/10.1051/m2an/2011056
23. DeVore, R., Petrova, G., Wojtaszczyk, P.: Greedy algorithms for reduced bases in Banach spaces. Constr. Approx. **37**(3), 455–466 (2013). https://doi.org/10.1007/s00365-013-9186-2
24. Cohen, A., Dahmen, W., Welper, G.: Adaptivity and variational stabilization for convection-diffusion equations. ESAIM: M2AN **46**(5), 1247–1273 (2012). https://doi.org/10.1051/m2an/2012003
25. Lattimore, T., Szepesvári, C.: Bandit algorithms. Cambridge University Press, Cambridge, England (2020)
26. Friedman, A.: Stochastic differential equations and applications. In: Stochastic differential equations, pp. 75–148. Springer, Berlin, Germany (2010). https://doi.org/10.1007/978-3-642-11079-5_2
27. Gobet, E.: Monte-Carlo methods and stochastic processes: from linear to non-linear. CRC Press, Boca Raton, FL, USA (2016)
28. Gobet, E., Maire, S.: A spectral Monte Carlo method for the Poisson equation. De Gruyter **10**(3–4), 275–285 (2004). https://doi.org/10.1515/mcma.2004.10.3-4.275
29. Gobet, E., Maire, S.: Sequential control variates for functionals of Markov processes on JSTOR. SIAM J. Numer. Anal. **43**(3), 1256–1275 (2006)
30. Maday, Y., Nguyen, N.C., Patera, A.T., Pau, S.H.: A general multipurpose interpolation procedure: the magic points. CPAA **8**(1), 383–404 (2008). https://doi.org/10.3934/cpaa.2009.8.383
31. Cohen, A., Migliorati, G.: Optimal weighted least-squares methods. SMAI Journal of Computational Mathematics **3**, 181–203 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.