Arch Comput Methods Eng (2010) 17: 435–454
DOI 10.1007/s11831-010-9056-z

ORIGINAL PAPER

Reduced Basis Techniques for Stochastic Problems

S. Boyaval · C. Le Bris · T. Lelièvre · Y. Maday · N.C. Nguyen · A.T. Patera

Received: 15 March 2010 / Accepted: 15 March 2010 / Published online: 16 October 2010

Abstract We report here on the recent application of a now classical general reduction technique, the *Reduced-Basis* (RB) approach initiated by C. Prud'homme et al. in J. Fluids Eng. 124(1), 70–80, 2002, to the specific context of differential equations with random coefficients. After an elementary presentation of the approach, we review two contributions of the authors: in Comput. Methods Appl. Mech. Eng. 198(41–44), 3187–3206, 2009, which presents the application of the RB approach for the discretization of a simple second order elliptic equation supplied with a random boundary condition, and in Commun. Math. Sci., 2009, which uses a RB type approach to reduce the variance in the Monte-Carlo simulation of a stochastic differential equation. We conclude the review with some general comments and also discuss possible tracks for further research in the direction.

S. Boyaval (✉) · C. Le Bris · T. Lelièvre
CERMICS, Ecole des ponts ParisTech, Université Paris-Est, 6-8 avenue Blaise Pascal, Cité Descartes, 77455 Marne la Vallée Cedex 2, France
e-mail: boyaval@cermics.enpc.fr

C. Le Bris
e-mail: lebris@cermics.enpc.fr

T. Lelièvre
e-mail: lelievre@cermics.enpc.fr

S. Boyaval · C. Le Bris · T. Lelièvre
MICMAC project team, INRIA, Domaine de Voluceau, Rocquencourt, 78153 Le Chesnay Cedex, France

Y. Maday
UMR 7598, Laboratoire Jacques-Louis Lions, UPMC Univ Paris 06, 75005 Paris, France
e-mail: maday@ann.jussieu.fr

Y. Maday
Division of Applied Mathematics, Brown University, Providence, USA

N.C. Nguyen · A.T. Patera
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

N.C. Nguyen
e-mail: cuongng@mit.edu

A.T. Patera
e-mail: patera@mit.edu

1 Introduction

In this work we describe reduced basis (RB) approximation and *a posteriori* error estimation methods for rapid and reliable evaluation of *input-output relationships* in which the *output* is expressed as a functional of a *field variable* that is the solution of an *input-parametrized* system. In this paper our emphasis is on stochastic phenomena: the parameter is random; the system is a partial differential equation with random coefficients, or a stochastic differential equation, namely a differential equation forced by a Brownian process.

The reduced basis approach is designed to serve two important, ubiquitous, and challenging engineering contexts: real-time, such as estimation and control; and many-query, such as design, multi-scale simulation, and—our emphasis here—statistical analysis. The parametric real-time and many-query contexts represent not only computational challenges, but also computational opportunities: we may restrict our attention to a manifold of solutions, which can be rather accurately represented by a *low-dimensional vector space*; we can accept greatly increased pre-processing or "Offline" cost in exchange for greatly decreased "Online" cost for each new input-output evaluation. (All of these terms, such as "Online," will be more precisely defined in

the Sect. 2.1 which constitutes a pedagogical introduction to the reduced basis approach.) Most variants of the reduced basis approach exploit these opportunities in some important fashion.

Early work on the reduced basis method focused on deterministic algebraic and differential systems arising in specific domains [2, 23, 51, 57–61]; the techniques were subsequently extended to more general finite-dimensional systems as well as certain classes of partial differential equations (and ordinary differential equations) [7, 22, 41, 62, 63, 73, 74, 77, 78]; the next decades saw further expansion into different applications and classes of equations, such as fluid dynamics and the incompressible Navier-Stokes equations [16, 28, 38, 71]. There is ample evidence of potential and realized success.

Recent research in reduced basis methods for deterministic parametrized partial differential equations both borrows from earlier efforts and also emphasizes new components: sampling techniques for construction of optimal reduced basis approximation spaces in particular in higher dimensional parameter domains [14, 53, 81]; rigorous *a posteriori* error estimation in appropriate norms and for particular scalar outputs of interest [29, 37]; and fastidious separation between the offline stage and online stage of the computations to achieve very rapid response [54]. These reduced basis methods can now be applied to larger, more global parameter domains, with much greater certainty and error control.

In this paper we emphasize the application of certified reduced basis methods to stochastic problems. Two illustrative approaches are explored. In the first approach [14] we consider application of the reduced basis method to partial differential equations with random coefficients: we associate realizations of the random solution field to deterministic solutions of a parametrized deterministic partial differential equation; we apply the classical reduced basis approach to the parametrized deterministic partial differential equation. Statistical information may finally be obtained, for example through Monte Carlo approximations. New issues arise related to the simultaneous approximation of both the input random field and the solution random field.

In the second approach [13] we directly consider a statistical embodiment of the reduced basis notions. Here reduced basis ideas originally conceived in the deterministic differential context are re-interpreted in the statistical context: the deterministic differential equation is replaced by a parametrized random process; snapshots on the parametric manifold are replaced by correlated ensembles on the parametric manifold; error minimization (in the Galerkin sense) is replaced by variance reduction; offline and online stages are effected through fine and coarse ensembles. This technique is here applied to parametrized stochastic differential equations.

We begin, in Sect. 2, with an initiation to the RB approach, considering a simple, prototypical elliptic problem, with deterministic coefficients. Section 3 then presents the first approach to stochastic problems, namely the application of the reduced basis method to a boundary value problem supplied with a random boundary condition. The section summarizes the results some of us obtained in [14]. With Sect. 4, we consider the second approach, and we address a problem different in nature, although also involving randomness. The issue considered is the variance reduction of a Monte-Carlo method for solving a stochastic differential equation. The RB approach has been successfully employed in [13] to efficiently generate companion variables that are used as control variate and eventually reduce the variance of the original quantities. The section outlines the approach and shows its success on representative results obtained. We conclude the article presenting in Sect. 5 some potential, alternate applications of the approach in the random context.

2 An Initiation to Reduced-Basis Techniques

We begin with an overview of Reduced Basis techniques. The level of our exposition is elementary. Our purpose here is to introduce the main ideas underlying the approach, leaving aside all unnecessary technicalities. The reader already familiar with this family of approximation approaches may easily skip this section and directly proceed to Sects. 3 and 4 where the adaptation of the general technique to the specific case of partial differential equations with random coefficients and to variance reduction using the RB approach will be addressed. We also refer to [45, 76, 79] for pedagogic introductions to the standard RB method, though with different perspectives.

2.1 Outline of the Reduced Basis Approach

Assume that we need to evaluate, for many values of the parameter μ, some *output* quantity $s(\mu) = F(u(\mu))$ function of the solution $u(\mu)$ to a partial differential equation parametrized by this parameter μ. If the computation of $u(\mu)$ and $s(\mu)$ for each single value of the parameter μ already invokes elaborate algorithms, and this is indeed the case in the context of partial differential equations, then the numerical simulation of $u(\mu)$ and $s(\mu)$ for many μ may become a computationally overwhelming task. Reducing the cost of parametrized computations is thus a challenge to the numerical simulation. This is the purpose of Reduced Basis techniques (abbreviated as RB throughout this article) to reduce this cost.

Let us formalize our discussion in the simple case of a partial differential equation which is an elliptic second order equation of the form (see (6) below):

$$-\operatorname{div}(\underline{\underline{A}}(\mu)\nabla u(\mu)) = f,$$

on a domain $\mathcal{D}$ with homogeneous Dirichlet boundary conditions. The mathematical setting is classical. We assume that the solution of interest $u(\mu) \in X$ is an element of a Hilbert space X with inner product $(\cdot,\cdot)_X$ and norm $\|\cdot\|_X$. The output $s(\mu) = F(u(\mu)) \in \mathbb{R}$ is a scalar quantity where $F : X \to \mathbb{R}$ is a smooth (typically linear) function and μ is a P-dimensional parameter varying in a fixed given range $\Lambda \subset \mathbb{R}^P$. An example of such output s is $s(\mu) = F(u(\mu)) := \int_{\mathcal{D}} f\, u(\mu)$ (see (8) below). The function $u(\mu)$ is mathematically defined as the solution to the general variational formulation:

$$\text{Find } u(\mu) \in X \text{ solution to } a(u(\mu), v; \mu) = l(v), \quad \forall v \in X, \tag{1}$$

where $a(\cdot,\cdot;\mu)$ is a symmetric bilinear form, continuous and coercive on X and where $l(\cdot)$ is a linear form, continuous on X. For all $\mu \in \Lambda$, $a(\cdot,\cdot;\mu)$ thus defines an inner product in X. The existence and uniqueness of $u(\mu)$, for each μ, is then obtained by standard arguments.

We henceforth denote by $\|\cdot\|_\mu$ the norm $\|\cdot\|_\mu = \sqrt{a(\cdot,\cdot;\mu)}$ equivalent to $\|\cdot\|_X$ (under appropriate assumptions on $\underline{\underline{A}}$, see below), which is usually termed the energy norm. In the sequel, we denote by $u_{\mathcal{N}}(\mu) \in X_{\mathcal{N}}$ an accurate Galerkin approximation for $u(\mu)$ in a linear subspace $X_{\mathcal{N}} \subset X$ of dimension $\mathcal{N} \gg 1$ and by $s_{\mathcal{N}}(\mu) = F(u_{\mathcal{N}}(\mu))$ the corresponding approximation for the output $s(\mu)$. For that particular choice of $X_{\mathcal{N}}$, we assume that the approximation error $|s(\mu) - s_{\mathcal{N}}(\mu)|$ is uniformly sufficient small for all $\mu \in \Lambda$. That is, $s_{\mathcal{N}}(\mu)$ is considered as a good approximation of the output $s(\mu)$ in practical applications. The difficulty is, we put ourselves in the situation where computing $s_{\mathcal{N}}(\mu)$ for all the values μ needed is too expensive, given the high dimensionality $\mathcal{N}$ of the space $X_{\mathcal{N}}$ and the number of parameters μ for which (1) need to be solved.

The RB approach typically consists of two steps. The purpose of the first step is to construct a linear subspace

$$X_{\mathcal{N},N} = \mathbf{Span}(u_{\mathcal{N}}(\mu_n^N), n = 1, \ldots, N), \tag{2}$$

subset of $X_{\mathcal{N}}$, of dimension $N \ll \mathcal{N}$, using a few approximate solutions to (1) for particular values of the parameter μ. The point is of course to carefully select these values $(\mu_n^N)_{1\le n\le N} \in \Lambda^N$ of the parameter, and we will discuss this below (see (4) and (5)). For intuitively clear reasons, the particular solutions $u_{\mathcal{N}}(\mu_n^N)$ are called *snapshots*. This first step is called the *offline* step, and is typically an expensive computation, performed once for all. In a second step, called the *online* step, an approximation $u_{\mathcal{N},N}(\mu) \in X_{\mathcal{N},N}$ of the solution to (1) is computed as a linear combination of the $u_{\mathcal{N}}(\mu_n^N)$. The problem solved states:

$$\text{Find } u_{\mathcal{N},N}(\mu) \in X_{\mathcal{N},N} \text{ solution to}$$
$$a(u_{\mathcal{N},N}(\mu), v; \mu) = l(v), \quad \forall v \in X_{\mathcal{N},N}. \tag{3}$$

This problem is much less computationally demanding than solving for the fine solution $u_{\mathcal{N}}(\mu)$, and will be performed for many values of the parameter μ. We denote by $s_{\mathcal{N},N}(\mu) = F(u_{\mathcal{N},N}(\mu))$ the corresponding approximation of the output $s(\mu)$. An *a posteriori* estimator $\Delta_N^s(\mu)$ for the output approximation error $|s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)|$ is needed in order to appropriately calibrate N and select the $(\mu_n^N)_{1\le n\le N}$. This *a posteriori* estimator may also be used in the online step to check the accuracy of the output. We shall make this precise below. For the time being, we only emphasize that the *a posteriori* analysis we develop aims at assessing the quality of the approximation of the output s (and not the overall quality of the approximation of the solution), see [1] and references therein. The method is typically called a *goal oriented* approximation method.

The formal argument that gives hope to construct an accurate approximation of the solution $u(\mu)$ to (1) using this process is that the manifold $\mathcal{M}_{\mathcal{N}} = \{u_{\mathcal{N}}(\mu), \mu \in \Lambda\}$ is expected to be well approximated by a linear space of dimension much smaller than $\mathcal{N}$, the dimension of the ambient space $X_{\mathcal{N}}$. An expansion on *a few* snapshots N has therefore a chance to succeed in accurately capturing the solution $u(\mu)$ for all parameter values μ. The reduced basis method is fundamentally a discretization method to approximate the state space $\mathcal{M}_{\mathcal{N}}$, with a view to computing an accurate approximation of the output. Of course, this requirement strongly depends on the choice of the parametrization which is a matter of modelling.

The RB method yields good approximations $s_{\mathcal{N},N}(\mu)$ of $s_{\mathcal{N}}(\mu)$ under appropriate assumptions on the dependency of the solution $u(\mu)$ on the input parameter μ. As a consequence, optimal choices for the approximation space $X_{\mathcal{N},N}$ should account for the dependency of the problem with respect to μ. More precisely, the method should select parameter values $(\mu_n^N)_{1\le n\le N} \in \Lambda^N$ with a view to controlling the norm of the output approximation error $|s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)|$ as a function of μ. For most applications, the appropriate norm to consider for the error as a function of μ is the L^∞ norm and this is the choice indeed made by the RB approach, in contrast to many other, alternative approaches. The desirable choice of $(\mu_n^N)_{1\le n\le N}$ is thus defined by:

$$(\mu_n^N)_{1\le n\le N} \in \operatorname*{arginf}_{(\mu_n)_{1\le n\le N}\in\Lambda^N} \Big(\sup_{\mu\in\Lambda} |s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)|\Big). \tag{4}$$

Note that, although not explicitly stated, the rightmost term $s_{\mathcal{N},N}(\mu)$ in (4) parametrically depends on $(\mu_n)_{1\le n\le N}$ because the solution to (3) for μ is developed as a linear combination of the corresponding snapshots $u_{\mathcal{N}}(\mu_n)$.

It is unfortunately very difficult to compute (4) in practice. With the publication [75], the RB approach suggests an alternative, practically feasible procedure. Instead of the

parameters $(\mu_n^N)_{1\le n\le N}$ defined by (4), the idea is to select approximate minimizers of

$$(\mu_n^N)_{1\le n\le N} \in \underset{(\mu_n)_{1\le n\le N}\in\Lambda^N}{\operatorname{arginf}} \Big(\sup_{\mu\in\Lambda_{\text{trial}}} \Delta_N^s(\mu) \Big). \tag{5}$$

Note that there are two differences between (4) and (5). First, the set Λ has been discretized into a very large trial sample of parameters $\Lambda_{\text{trial}} \subset \Lambda$. Second, and more importantly, the quantity $\Delta_N^s(\mu)$ minimized in (5) is an *estimator* of $|s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)|$. A fundamental additional ingredient is that the approximate minimizers of (5) are selected using a specific procedure, called *greedy* because the parameter values μ_n^N, $n = 1, \ldots, N$, are selected incrementally. Such an incremental procedure is in particular interesting when N is not known in advance, since the computation of approximate μ_n^N $(1 \le n \le N)$ does not depend on N and may be performed until the infimum in (5) is judged sufficiently low.

Of course, the computation of approximations to (5) with such a greedy algorithm can still be expensive, because a very large trial sample of parameters $\Lambda_{\text{trial}} \subset \Lambda$ might have to be explored. The RB method is thus only considered efficient when the original problem, problem (1) here, has to be computed for such a large number of input parameter values μ, that the overall procedure (computationally expensive offline step and then, efficient online step) is practically more amenable than following the original, direct approach. One often speaks of a *many-query* computational context when it is the case. Notice that the RB method is not to be seen as a competitor to the usual discretization methods; it rather builds upon already efficient discretization methods using appropriate choices of $X_{\mathcal{N}}$ in order to speed up computations that have to be performed repeatedly.

The approach can therefore be reformulated as the following two-step procedure

- in the *offline* stage (which, we recall, may possibly be computationally expensive), one "learns" from a very large trial sample of parameters $\Lambda_{\text{trial}} \subset \Lambda$ how to choose a small number N of parameter values; this is performed using a greedy algorithm that incrementally selects the μ_n, $n = 1, \ldots, N$; the selection is based on the estimator $\Delta_N^s(\mu)$; accurate approximations $u_{\mathcal{N}}(\mu_n)$ for solutions $u(\mu_n)$ to (1) are correspondingly computed at those few parameter values;
- in the *online* stage, computationally inexpensive approximations $u_{\mathcal{N},N}(\mu)$ of solutions $u(\mu)$ to (1) are computed for many values $\mu \in \Lambda$ of the parameter, using the Galerkin projection (3); the latter values need not be in the sample Λ_{trial}, and yield approximations $s_{\mathcal{N},N}(\mu)$ for the output $s(\mu)$; the estimator $\Delta_N^s(\mu)$, already useful in the offline step, is again employed to check the quality of the online approximation (this check is called *certification*).

Notice that the computation of the error estimator $\Delta_N^s(\mu)$ needs to be inexpensive, in order to be efficiently used on the very large trial sample in the offline stage, and for each new parameter values in the online stage.

One might ask why we proceed with the reduced basis approach rather than simply interpolate $s(\mu)$, given the few values $\{s_{\mathcal{N}}(\mu_1), \ldots, s_{\mathcal{N}}(\mu_N)\}$. There are several important reasons: first, we have rigorous error estimators based on the residual that are simply not possible based on direct interpolation; second, these residuals and error bounds drive the greedy procedure; third, the state-space approach provides Galerkin projection as an "optimal" interpolant for the particular problem of interest; and fourth, in higher parameter dimensions (say of the order of 10 parameters), in fact the a priori construction of scattered-data interpolation points and procedures is very difficult, and the combination of the greedy and Galerkin is much more effective.

We are now in position to give some more details on both the offline and online steps of the RB approach in a very simple case: an elliptic problem, with an affine dependency on the parameter. Our next section will make specific what the greedy algorithm, the estimator $\Delta_N^s(\mu)$, along with other objects abstractly manipulated above, are.

2.2 Some More Details on a Simple Case

As mentioned above, we consider for simplicity the Dirichlet problem

$$\begin{cases} -\operatorname{div}(\underline{\underline{A}}(\mu)\nabla u(\mu)) = f & \text{in } \mathcal{D}, \\ u(\mu) = 0 & \text{on } \partial\mathcal{D}, \end{cases} \tag{6}$$

where $\mathcal{D}$ is a two-, or three-dimensional domain and the matrix $\underline{\underline{A}}(\mu)$ is parameterized by a single scalar parameter $\mu \in \Lambda = [\mu_{\min}, \mu_{\max}] \subset \mathbb{R}_+^*$. We assume that the matrix A is symmetric and depends on μ in an *affine* way:

$$\underline{\underline{A}}(\mu) = \underline{\underline{A_0}} + \mu\, \underline{\underline{A_1}}, \quad \forall \mu \in \Lambda. \tag{7}$$

This assumption (7) is a crucial ingredient, responsible, as we shall explain below, for a considerable speed-up and thus for the success of the RB approach here. More generally, either we must identify by inspection or construction an "affine" decomposition of the form (7), or we must develop an appropriate affine approximation; both issues are discussed further below.

We assume we are interested in efficiently computing, for many values of $\mu \in \Lambda$, the output:

$$s(\mu) = F(u(\mu)) := \int_{\mathcal{D}} f\, u(\mu). \tag{8}$$

This is of course only a specific situation. The output function can be much more general, like a linear form $\int_{\mathcal{D}} g\, u(\mu)$ with some $g \neq f$. Many other cases are possible, but they all come at a cost, both in terms of analysis and in terms of workload. The case (8), where the output coincides with the linear form present in the right-hand side of the variational formulation of (6) (and where the bilinear form a involved in the variational formulation is symmetric), is called *compliant*. Having (8) as an output function in particular simplifies the *a posteriori* error analysis of the problem (namely the construction of $\Delta_N^s(\mu)$).

We equip the problem, somewhat vaguely formulated in (6)–(7)–(8) above, with the appropriate mathematical setting that allow for all our necessary manipulations below to make sense. For consistency, we now briefly summarize this setting. The domain $\mathcal{D}$ is an open bounded connected domain with Lipschitz boundary $\partial\mathcal{D}$, the right-hand side $f \in L^2(\mathcal{D})$ belongs to the Lebesgue space of square integrable functions, $\underline{\underline{A}}(\mu)$ is a *symmetric* matrix, which is positive-definite almost everywhere in $\mathcal{D}$. Each entry of $\underline{\underline{A}}(\mu)$ is assumed in $L^\infty(\mathcal{D})$. We assume $\underline{\underline{A_0}}$ is symmetric positive-definite, and $\underline{\underline{A_1}}$ is symmetric positive. The ambient Hilbert space X is chosen equal to the usual Sobolev space $H_0^1(\mathcal{D})$. The function $u(\mu)$ is defined as the solution to the variational formulation (1) with $a(w, v; \mu) = \int_{\mathcal{D}} \underline{\underline{A}}(\mu)\nabla w \cdot \nabla v$, $l(v) = \int_{\mathcal{D}} f\, v$, for all v, w, in X and all $\mu \in \Lambda$.

As for the direct discretization of the problem, we also put ourselves in a classical situation. If $\mathcal{D}$ is polygonal for instance, there exist many discretization methods that allow to compute Galerkin approximations $u_{\mathcal{N}}(\mu)$ of $u(\mu)$ in finite dimensional linear subspaces $X_{\mathcal{N}}$ of X for any fixed parameter value $\mu \in \Lambda$. The Finite-Element method [17, 83] is of course a good example. Then, for each parameter value $\mu \in \Lambda$, the numerical computation of $u_{\mathcal{N}}(\mu) = \sum_{n=1}^{\mathcal{N}} U_n(\mu)\phi_n$ on the Galerkin basis $(\phi_n)_{1\leq n\leq \mathcal{N}}$ of $X_{\mathcal{N}}$ is achieved by solving a large linear system

$$\text{Find } U(\mu) \in \mathbb{R}^{\mathcal{N}} \text{ solution to } \underline{\underline{B}}(\mu)U(\mu) = b,$$

for the vector $U(\mu) = (U_n(\mu))_{1\leq n\leq \mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$, where $b = (l(\phi_n))_{1\leq n\leq \mathcal{N}}$ is a vector in $\mathbb{R}^{\mathcal{N}}$ and $\underline{\underline{B}}(\mu) = \underline{\underline{B_0}} + \mu\, \underline{\underline{B_1}}$ is a $\mathcal{N} \times \mathcal{N}$ real invertible matrix. Note that the assumption of affine parametrization (7) makes possible, for each parameter value μ, the computation of the entries of the matrix $\underline{\underline{B}}(\mu)$ in $O(\mathcal{N})$ operations (due to sparsity), using the precomputed integrals $(\underline{\underline{B_q}})_{ij} = \int_{\mathcal{D}} \underline{\underline{A_q}}\nabla\phi_i \cdot \nabla\phi_j$, $i, j = 1, \ldots, \mathcal{N}$ for $q = 0, 1$. The evaluation of $U(\mu)$ for many $J \gg 1$ parameter values μ using iterative solvers costs $J \times O(\mathcal{N}^k)$ operations with $k \leq 3$ [25], where k depends on the sparsity and the conditioning number of the involved matrices.

As mentioned above in our general, formal presentation, the goal of the RB approach is to build a smaller finite dimensional approximation space $X_{\mathcal{N},N} \subset X_{\mathcal{N}}$ sufficiently good for all $\mu \in \Lambda$, with $N \ll \mathcal{N}$, so that the computational cost is approximately reduced to $N \times O(\mathcal{N}^k) + J \times O(N^3)$, where $N \times O(\mathcal{N}^k)$ is the cost of offline computations and $J \times O(N^3)$, the cost of online computations, is *independent of* $\mathcal{N}$, using the Galerkin approximation (3) in $X_{\mathcal{N},N}$.

We now successively describe in the following three paragraphs the construction of the *a posteriori* estimator, that of the *greedy* algorithm employed in the offline step, and the combination of all ingredients in the online step.

2.2.1 A Posteriori Estimator

For the coercive elliptic problem (6), the *a posteriori* error estimator $\Delta_N^s(\mu)$ for the output RB approximation error $|s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)|$ is simple to devise, based on a global *a posteriori* error estimator $\Delta_N(\mu)$ for $\|u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu)\|_\mu$ using a classical technique with residuals [8].

We refer to [20, 32, 55, 56, 69, 85, 86] for the construction of similar *a posteriori* error estimators in various applied settings of the RB method.

We first define the residual bilinear form

$$g(w, v; \mu) = a(w, v; \mu) - l(v), \quad \forall w, v \in X,\ \forall \mu \in \Lambda,$$

and the operator $G(\mu) : X_{\mathcal{N}} \to X_{\mathcal{N}}$ such that

$$g(w, v; \mu) = (G(\mu)\, w, v)_X, \quad \forall w, v \in X_{\mathcal{N}},\ \forall \mu \in \Lambda.$$

We next assume we are given, for all $\mu \in \Lambda$, a lower bound $\alpha_{LB}(\mu)$ for the coercivity constant of $a(\cdot, \cdot; \mu)$ on $X_{\mathcal{N}}$, that is,

$$0 < \alpha_{LB}(\mu) \leq \alpha_c(\mu) = \inf_{w \in X_{\mathcal{N}} \setminus \{0\}} \frac{a(w, w; \mu)}{\|w\|_X^2}, \quad \forall \mu \in \Lambda. \tag{9}$$

The lower bound $\alpha_{LB}(\mu)$ can be given by an *a priori* analysis before discretization ($\alpha_{LB}(\mu)$ would then be the coercivity constant of $a(\cdot, \cdot; \mu)$ on X), or numerically evaluated based on an approximation procedure, which might be difficult in some cases, see [33, 79].

Then the a posteriori estimator we use is defined in the following.

Proposition 1 *For any linear subspace $X_{\mathcal{N},N}$ of $X_{\mathcal{N}}$, there exists a computable error bound $\Delta_N^s(\mu)$ such that*:

$$|s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)| \leq \Delta_N^s(\mu) := \frac{\|G(\mu)\, u_{\mathcal{N},N}(\mu)\|_X^2}{\alpha_{LB}(\mu)}, \quad \forall \mu \in \Lambda. \tag{10}$$

For consistency, we now briefly outline the proof of this proposition. We simply observe the sequence of equalities

$$\begin{aligned} |s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)| &= |F(u_{\mathcal{N}}(\mu)) - F(u_{\mathcal{N},N}(\mu))| \\ &= |l(u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu))| \\ &= |a(u_{\mathcal{N}}(\mu), u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu); \mu)| \\ &= |a(u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu), u_{\mathcal{N}}(\mu) \\ &\quad - u_{\mathcal{N},N}(\mu); \mu)| \\ &= \|u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu)\|_{\mu}^2 \end{aligned} \tag{11}$$

using the linearity of $F = l$, the variational problem and its discretized approximation in $X_{\mathcal{N}}$, the symmetry of $a(\cdot, \cdot; \mu)$ and the fact that $a(u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu), v) = 0$, for all $v \in X_{\mathcal{N},N}$. On the other hand, inserting $v = u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu)$ in the general equality $a(u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu), v; \mu) = -g(u_{\mathcal{N},N}(\mu), v; \mu)$ (for all $v \in X_{\mathcal{N}}$), and using the bound $\sqrt{\alpha_{LB}(\mu)}\|v\|_X \leq \|v\|_{\mu}$ (for all $v \in X_{\mathcal{N}}$), we note that

$$\|u_{\mathcal{N}}(\mu) - u_{\mathcal{N},N}(\mu)\|_{\mu} \leq \Delta_N(\mu) := \frac{\|G(\mu)\, u_{\mathcal{N},N}(\mu)\|_X}{\sqrt{\alpha_{LB}(\mu)}}. \tag{12}$$

We conclude the proof of (10) combining (11) with (12).

We may similarly prove (but we will omit the argument here for brevity) the inverse inequality:

$$\Delta_N^s(\mu) \leq \left(\frac{\gamma(\mu)}{\alpha_{LB}(\mu)}\right)^2 |s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N}(\mu)|, \tag{13}$$

using the continuity constant

$$\gamma(\mu) = \sup_{w \in X_{\mathcal{N}} \setminus \{0\}} \sup_{v \in X_{\mathcal{N}} \setminus \{0\}} \frac{a(w, v; \mu)}{\|w\|_X \|v\|_X} \tag{14}$$

of the bilinear form $a(\cdot, \cdot; \mu)$ on $X_{\mathcal{N}}$ for all $\mu \in \Lambda$, which is bounded above by the continuity constant on X. The inequality (13) ensures sharpness of the *a posteriori* estimator (10), depending of course on the quality of the lower-bound $\alpha_{LB}(\mu)$.

2.2.2 *Offline Stage and Greedy Algorithm*

The greedy algorithm employed to select the snapshots $u_{\mathcal{N}}(\mu_n)$ typically reads:

1: choose $\mu_1 \in \Lambda$ randomly
2: compute $u_{\mathcal{N}}(\mu_1)$ to define $X_{\mathcal{N},1} = \mathbf{Span}(u_{\mathcal{N}}(\mu_1))$
3: **for** $n = 2$ to N **do**
3: choose $\mu_n \in \text{argmax}\{\Delta_{n-1}^s(\mu), \mu \in \Lambda_{\text{trial}}\}$
3: compute $u_{\mathcal{N}}(\mu_n)$ to define $X_{\mathcal{N},n} = \mathbf{Span}(u_{\mathcal{N}}(\mu_m), m = 1, \ldots, n)$
4: **end for**

In the initialization step, we may equally use $\mu_1 \in \text{argmax}\{|s(\mu)|, \ \mu \in \Lambda_{\text{smalltrial}}\}$, where $\Lambda_{\text{smalltrial}} \subset \Lambda$ is a very small trial sample in Λ, much smaller than Λ itself. Likewise, the algorithm can in practice be terminated when the output approximation error is judged sufficiently small (say, $|\Delta_N^s(\mu)| \leq \varepsilon$ for all $\mu \in \Lambda_{\text{trial}}$), and not when the iteration number reaches a maximum $n = N$.

The choice of the trial sample Λ_{trial} (and similarly, the smaller sample $\Lambda_{\text{smalltrial}}$) is a delicate practical issue. It is often simply taken as a random sample in Λ. Of course, this first guess may be insufficient to reach the required accuracy level ε in $\Delta_N^s(\mu)$, for all $\mu \in \Lambda$, in the online stage. But fortunately, if the computation of $\Delta_N^s(\mu)$ for any $\mu \in \Lambda$ is sufficiently inexpensive, one can check this accuracy online for each query in μ. Should $\Delta_N^s(\mu) > \varepsilon$ occur for some online value of the parameter μ, one can still explicitly compute $u_{\mathcal{N}}(\mu)$ for that exact same μ and enrich the space $X_{\mathcal{N},N}$ correspondingly. This bootstrap approach of course allows to reach the required accuracy level ε at that μ. It provides significant computational reductions in the online stage *provided that* the RB approximation space $X_{\mathcal{N},N}$ does not need to be enriched too often online. We will explain the methodology for fast computations of $\Delta_N^s(\mu)$ below.

The offline selection procedure needs to be consistent with the online procedure, and thus the above greedy algorithm uses the same estimator $\Delta_N^s(\mu)$ for all $\mu \in \Lambda_{\text{trial}}$ as the online procedure. Since the computation of $\Delta_N^s(\mu)$ is, by construction and on purpose, fast for all $\mu \in \Lambda$, the exploration of a very large training sample Λ_{trial} (which is a subset of Λ) is possible offline.

No systematic procedure seems to be available, which allows to build good initial guesses Λ_{trial} *ex nihilo*. Even for a specific problem, we are not aware either of any *a priori* results that quantify how good an initial guess Λ_{trial} is. The only option is, as is indeed performed by the RB approach, to *a posteriori* check, and possibly improve, the quality of the initial guess Λ_{trial} (however, the quality of the initial guess Λ_{trial} can be slightly improved offline by using adaptive training samples in the greedy algorithm [30]).

The estimators $\Delta_N^s(\mu)$ are employed in the greedy algorithm to filter candidate values for Λ. Numerous numerical evidences support the success of this pragmatic approach [20, 32, 55, 56, 69, 70, 85, 86].

Last, notice that the cost of offline computations scales as $W_{\text{offline}} = O(|\Lambda_{\text{trial}}|) \times (\sum_{n=1}^{N-1} w_{\text{online}}(n)) + N \times O(\mathcal{N}^k)$ where $w_{\text{online}}(n)$ is the marginal cost of one online-type computation for $u_{\mathcal{N},n}(\mu)$ and $\Delta_n^s(\mu)$ at a selected parameter value $\mu \in \Lambda_{\text{trial}}$ (where $1 \leq n \leq N-1$), and $O(|\Lambda_{\text{trial}}|)$ includes a max-search in Λ_{trial}. (Recall that $k \leq 3$ depends on the solver used for large sparse linear systems.)

2.2.3 *Online Stage: Fast Computations Including A Posteriori Estimators*

We now explain how to efficiently compute $u_{\mathcal{N},n}(\mu)$, $s_{\mathcal{N},n}(\mu)$ and $\Delta_n^s(\mu)$ once the RB approximation space $X_{\mathcal{N},n}$ has been constructed. This task has to be completed twice in the RB approach. First, this is used in the many offline computations when $\mu \in \Lambda_{\text{trial}}$ explores the trial sample in order to find $\mu_n \in \text{argmax}\{\Delta_{n-1}^s(\mu), \mu \in \Lambda_{\text{trial}}\}$ at each iteration n of the greedy algorithm. Second, this is used for the many online computations (when $n = N$). We present the procedure in the latter case.

By construction, the family $(u_{\mathcal{N}}(\mu_n))_{1\le n\le N}$ generated by the greedy algorithm described in Sect. 2.2.2 is a basis of

$$X_{\mathcal{N},N} = \textbf{Span}(u_{\mathcal{N}}(\mu_n),\ n = 1, \ldots, N).$$

For any $\mu \in \Lambda$, we would then like to compute the RB approximation $u_{\mathcal{N},N}(\mu) = \sum_{n=1}^N U_{N,n}(\mu) u_{\mathcal{N}}(\mu_n)$, which can be achieved by solving a small $N \times N$ (full) linear system

$$\underline{\underline{C}}(\mu) U_N(\mu) = c,$$

for the vector $U_N(\mu) = (U_{N,n}(\mu))_{1\le n\le N} \in \mathbb{R}^N$, with $c = (l(u_{\mathcal{N}}(\mu_n)))_{1\le n\le N}$ a vector in $\mathbb{R}^N$ and $\underline{\underline{C}}(\mu) = \underline{\underline{C_0}} + \mu\, \underline{\underline{C_1}}$ is a $N \times N$ real invertible matrix. In practice, the matrix $\underline{\underline{C}}(\mu)$ is close to a singular matrix, and it is essential to compute the RB approximation as $u_{\mathcal{N},N}(\mu) = \sum_{n=1}^N \tilde{U}_{N,n}(\mu)\zeta_n$ using a basis $(\zeta_n)_{1\le n\le N}$ of $X_{\mathcal{N},N}$ that is orthonormal for the inner-product $(\cdot,\cdot)_X$. The determination of appropriate $(\zeta_n)_{1\le n\le N}$ is easily performed, since N is small, using Simple or Modified Gram-Schmidt procedures. The problem to solve states:

$$\text{Find } \tilde{U}_N(\mu) \in \mathbb{R}^N \text{ solution to } \underline{\underline{\tilde{C}}}(\mu)\tilde{U}_N(\mu) = \tilde{c}, \tag{15}$$

where $\tilde{U}_N(\mu) = (\tilde{U}_{N,n}(\mu))_{1\le n\le N} \in \mathbb{R}^N$, $\tilde{c} = (l(\zeta_n))_{1\le n\le N}$ is a vector in $\mathbb{R}^N$ and $\underline{\underline{\tilde{C}}}(\mu) = \underline{\underline{\tilde{C}_0}} + \mu\, \underline{\underline{\tilde{C}_1}}$ is a $N \times N$ real invertible matrix. So, for each parameter value $\mu \in \Lambda$, the entries of the latter matrix $\underline{\underline{\tilde{C}}}(\mu)$ can be computed in $O(N^2)$ operations using the precomputed integrals $(\underline{\underline{\tilde{C}_q}})_{ij} = \int_{\mathcal{D}} \underline{\underline{A_q}} \nabla\zeta_i \cdot \nabla\zeta_j$, $i, j = 1, \ldots, N$ for $q = 0, 1$. (Note that the assumption of affine parametrization is essential here.) And the evaluation of $\tilde{U}_N(\mu)$ for many $J \gg 1$ parameter values $\mu \in \Lambda$ finally costs $J \times O(N^3)$ operations using direct solvers for symmetric problems like Cholesky [25].

For each $\mu \in \Lambda$, the output $s_{\mathcal{N},N}(\mu) = F(u_{\mathcal{N},N}(\mu))$ can also be computed very fast in $O(N)$ operations upon noting that F is linear and all the values $F(u_{\mathcal{N},N}(\mu_n))$, $n = 1, \ldots, N$ can be precomputed offline. The corresponding *a posteriori* estimator $\Delta_N^s(\mu)$ given by (10) has now to be computed, hopefully equally fast. Because of the affine dependence of $\underline{\underline{A}}(\mu)$ on μ, a similar affine dependence $G(\mu) = G_0 + \mu\, G_1$ holds for the operator G (for all $\mu \in \Lambda$), where $(G_0\, w, v)_X = \int_{\mathcal{D}} \underline{\underline{A_0}} \nabla w \cdot \nabla v - \int_{\mathcal{D}} f\, v$, and $(G_1\, w, v)_X = \int_{\mathcal{D}} \underline{\underline{A_1}} \nabla w \cdot \nabla v$ for all v, w, in $X_{\mathcal{N}}$. So one can evaluate very fast the norm

$$\begin{aligned}\|G(\mu)\, u_{\mathcal{N},N}(\mu)\|_X^2 = {} & \|G_0\, u_{\mathcal{N},N}(\mu)\|_X^2 \\ & + 2\mu (G_0\, u_{\mathcal{N},N}(\mu), G_1\, u_{\mathcal{N},N}(\mu))_X \\ & + \mu^2 \|G_1\, u_{\mathcal{N},N}(\mu)\|_X^2 \end{aligned} \tag{16}$$

for $\mu \in \Lambda$, once, with obvious notation, the scalar products $(G_i\, u_{\mathcal{N},N}(\mu_p), G_j\, u_{\mathcal{N},N}(\mu_q))_X$, have been precomputed offline and stored. Assuming that the lower-bound $\alpha_{LB}(\mu)$ used in (10) is known, the computation of the *a posteriori* estimator $\Delta_N^s(\mu)$ itself is thus also very fast. Notice that the *affine* parametrization (7) plays again a crucial role in the above decomposition of the computation.

Finally, the marginal cost of one online-type computation on $X_{\mathcal{N},n}$ for one parameter value μ is $w_{\text{online}}(n) = O(n^3)$ (where $n = 1, \ldots, N$). So, assuming that no basis enrichment is necessary during the online stage using the RB approximation space $X_{\mathcal{N},N}$ (that is, $\Delta_N^s(\mu) < \varepsilon$ for all the parameter values μ queried online), the total online cost for many $J \gg 1$ parameter values μ scales as $W_{\text{online}} = J \times O(N^3)$. And, the total cost of computations with the RB approach is then $W_{\text{offline}} + W_{\text{online}} = N \times O(\mathcal{N}^k) + (J + O(|\Lambda_{\text{trial}}|)) \times O(N^3)$, which has to be compared to $J \times O(\mathcal{N}^k)$ operations for a direct approach (with $k \le 3$ depending on the solver used for large sparse linear systems). In the limit of infinitely many online evaluations $J \gg 1$, the computational saving of the RB approach is tremendous.

2.3 Some Elements of Analysis, and Some Extensions, of the RB Method

2.3.1 *Some Elements of Theory*

The RB approach has undoubtedly proved successful in a large variety of applications [20, 32, 55, 56, 69, 70, 85, 86]. The theoretical understanding of the approach is however still limited, and is far from covering all practical situations of interest. Of course, little theory is to be expected in the usual *a priori* way. As already explained, the RB approach is deliberately designed to *a posteriori* adapt to practical settings. The only available *a priori* analysis is related to two issues: the expected "theoretical" quality of the RB approximation, and the efficiency of the greedy algorithm. We now briefly summarize what is known to date on both issues.

The RB approach is in fact expected to perform *ideally*, in the following sense. In the context of our simple problem (6), it is possible, adapting the classical Lagrange interpolation theory to the context of parameterized boundary

value problems and assuming that the matrix $\underline{\underline{A_1}}$ is non-negative, to obtain an upper bound of (4). The following theoretical *a priori* analysis result follows. It states the exponential accuracy of the RB approximation in terms on the dimension N of the reduced basis.

Proposition 2 *For all parameter ranges* $\Lambda := [\mu_{\min}, \mu_{\max}] \subset \mathbb{R}_+^*$, *there exists an integer* $N_0 = O(\ln(\frac{\mu_{\max}}{\mu_{\min}}))$ *as* $\frac{\mu_{\max}}{\mu_{\min}} \to +\infty$, *and a constant* $c > 0$ *independent of* Λ *such that, for all* $N \geq N_0 \geq 2$, *there exist* N *parameter values* $\mu_{\min} =: \lambda_1^N < \cdots < \lambda_n^N < \lambda_{n+1}^N < \cdots < \lambda_N^N := \mu_{\max}$, $n = 2, \ldots, N-2$, *satisfying (recall* $\|\cdot\|_0 = \|\cdot\|_\mu$ *with* $\mu = 0$ *is an Hilbertian norm on* X*):*

$$\sup_{\mu\in\Lambda} \big(\inf\{\|u_{\mathcal{N}}(\mu) - w\|_0,\ w \in \mathbf{Span}(u_{\mathcal{N}}(\lambda_n^N), n = 1, \ldots, N)\}\big) \leq e^{-\frac{c}{N_0-1}(N-1)} \sup_{\mu\in\Lambda} \|u_{\mathcal{N}}(\mu)\|_0. \tag{17}$$

We refer to [12, Chap. 4] and [46, 70] for the proof of Proposition 2.

The approximation space $\mathbf{Span}(u_{\mathcal{N}}(\lambda_n^N), n = 1, \ldots, N)$ used for the statement and the proof of Proposition 2 is different from the RB approximation space $X_{\mathcal{N},N}$ built in practice by the RB greedy algorithm. Numerical experiments even suggest that it is not an equally good choice (see [70]). So it is desirable to better understand the actual outcome of the RB greedy algorithm used offline. The concept of *greedy* algorithm appears in many numerical approaches for problems of approximation. It typically consists in a recursive procedure approximating an optimal solution to a complex problem, using a sequence of sub-optimal solutions incrementally improved. Otherwise stated, each iteration takes the solution of the previous iteration as an initial guess and improves it. In the theory of approximation of functions in particular [21, 84], greedy algorithms are used to incrementally compute the combinations of functions from a given dictionnary which best approximate some given function. The RB greedy algorithm has a somewhat different viewpoint: it incrementally computes for integers N some basis functions $u_{\mathcal{N}}(\mu_n)$, $n = 1, \ldots, N$, spanning a linear space $X_{\mathcal{N},N}$ that best approximates a *family* of functions $u_{\mathcal{N}}(\mu)$, $\forall \mu \in \Lambda$. The RB greedy algorithm however has a flavour similar to other greedy algorithms that typically build best-approximants in general classes of functions. It is therefore possible to better understand the RB greedy algorithm using classical ingredients of approximation theory. The notion of *Kolmogorov width* [72] is an instance of such a classical ingredient. We refer to [12, Chap. 3] and [15] for more details and some elements of analysis of the RB greedy algorithm.

2.3.2 *Extensions of the Approach to Cases More General than (6)*

The RB approach of course does not only apply to simple situations like (6). Many more general situations may be addressed, the major limitation to the genericity of the approach being the need for constructing fast computable *a posteriori* error estimators.

Instances of problems where the RB approach has been successfully tested are the following: affine formulations, non-coercive linear elliptic problems, non-compliant linear elliptic problems, problems with non-affine parameters, nonlinear elliptic problems, semi-discretized (nonlinear) parabolic problems. The purpose of this section is to briefly review these extensions of our above simple setting. In the next section, we will then introduce a problem with random coefficients. For simplicity, we take it almost as simple as the above problem (6), see (23)–(24) below. We anticipate that, if they involve a random component, most of the extensions outlined in the present section could also, *in principle*, be treated using the RB approach.

Affine Formulations Beyond the simple case presented above in Sect. 2.2, which involves an elliptic operator in divergence form affinely depending on the parameter, the RB approach can be extended to general elliptic problems with variational formulation of the form

$$\text{Find } u(\mu) \in X \text{ solution to } g(u(\mu), v; \mu) = 0, \quad \forall v \in X, \tag{18}$$

where the form $g(\cdot, \cdot; \mu)$ on $X \times X$ admits an *affine* parametrization, that is, writes

$$g(w, v; \mu) = \sum_{q=1}^{Q} \Theta_q(\mu) g_q(w, v), \quad \forall w, v \in X,\ \forall \mu \in \Lambda, \tag{19}$$

with parameter-independent forms $(g_q(\cdot, \cdot))_{1\leq q\leq Q}$ (where some of the g_q may only depend on v) and coefficients $(\Theta_q(\mu))_{1\leq q\leq Q}$. We emphasize that the whole RB algorithm presented in the simple case above directly translates in this situation. In particular, the matrices used in the online evaluation procedure can be constructed offline.

Non-coercive Symmetric Linear Elliptic Problems The RB approach can be extended to the case where the symmetric continuous bilinear form $a(\cdot, \cdot; \mu)$ is not coercive but only *inf-sup stable*. An example is the Helmholtz problem treated in [82]. Our discussion of the elliptic problem above can be adapted in a straightforward way, the only change in offline and online computations being that the inf-sup stability con-

stant on $X_{\mathcal{N}}$:

$$0 < \beta_{LB}(\mu) \le \beta(\mu) := \inf_{w \in X_{\mathcal{N}} \setminus \{0\}} \sup_{v \in X_{\mathcal{N}} \setminus \{0\}} \frac{a(w, v; \mu)}{\|w\|_X \|v\|_X}, \quad \forall \mu \in \Lambda, \tag{20}$$

is substituted for $\alpha_{LB}(\mu)$. In practice, the evaluation of $\beta_{LB}(\mu)$ is typically more involved than the evaluation of the coercivity constant $\alpha_{LB}(\mu)$. We refer to [34] for an appropriate technique.

Non-compliant Linear Elliptic Problems In (8), the particular choices of $F = l$ for the output and of symmetric matrices $\underline{\underline{A}}(\mu)$ for the definition of the bilinear form $a(\cdot, \cdot; \mu)$ correspond to a particular class of problems called, we recall, *compliant*. Non-compliant linear elliptic problems can be treated as well, but this is somewhat more technical. These are the cases where, for some $\mu \in \Lambda$ at least, either $u(\mu)$ is solution to a weak form (18) with $g(v, w; \mu) = a(v, w; \mu) - l(w), \forall v, w \in X$ and the bilinear form $a(\cdot, \cdot; \mu)$ is not symmetric, or the output is $s(\mu) = F(u(\mu)) \neq l(u(\mu))$ with any linear continuous function $F : X \to \mathbb{R}$.

For instance, we explain how to treat the case of a bilinear form $a(\cdot, \cdot; \mu)$ that is not symmetric, but of course still continuous and inf-sup stable. The analysis requires considering the solution to the *adjoint* problem

$$\text{Find } \psi(\mu) \in X \text{ solution to } a(v, \psi(\mu); \mu) = -F(v), \quad \forall v \in X, \tag{21}$$

along with the corresponding Galerkin discretization $\psi_{\mathcal{N}}(\mu) \in X_{\mathcal{N}}$, the approximation space $X^{\star}_{\mathcal{N},N^{\star}}$ for the solution to (21), and an additional RB approximation space $X^{\star}_{\mathcal{N},N^{\star}} \subset X_{\mathcal{N}}$ of dimension $N^{\star} \ll \mathcal{N}$. The *a posteriori* estimator obtained is similar to (10), and writes

$$|s_{\mathcal{N}}(\mu) - s_{\mathcal{N},N,N^{\star}}(\mu)| \le \Delta^{s}_{N,N^{\star}}(\mu) := \frac{\|G(\mu) u_{\mathcal{N},N}(\mu)\|_X \|G^{\star}(\mu) \psi_{\mathcal{N},N^{\star}}(\mu)\|_X}{\beta_{LB}(\mu)}, \tag{22}$$

where $G^{\star}$ is defined from the adjoint problem (21) similarly to how G is defined from the original problem. Notice that we again used the inf-sup stability condition (20), which indeed holds true after permutation of the arguments v and w (since we work in a finite dimensional space $X_{\mathcal{N}}$), the value of the inf sup constant however being not the same. To build the reduced basis of the primal (respectively the dual) problem, in the offline stage, the *a posteriori* estimator is based on $\|G(\mu)\, u_{\mathcal{N},N}(\mu)\|_X$ (respectively $\|G^{\star}(\mu)\, \psi_{\mathcal{N},N^{\star}}(\mu)\|_X$). Apart from the above introduction and use of the adjoint problem, the treatment of the non-compliant case then basically follows the same lines as that of the compliant case.

Notice that a simple, but less sharp, estimate of the error (namely the left-hand side of (22)) can be obtained as $(\sup_{x \in \boldsymbol{X}_{\mathcal{N}}} \frac{|F(x)|}{\|x\|_X}) \|G(\mu) u_{\mathcal{N},N}(\mu)\|_X$. This simple error bound does not involve the solution of any dual problem, and may be of interest in particular in the case when multiple outputs are considered. However, the primal–dual error bound (22) will be much smaller (since it is quadratic and not linear in the residual) and in many situations, very easy to obtain, since the dual problem is typically simpler to solve than the primal problem (it is indeed linear).

Non-affine Parameters We have exploited in several places the affine dependence of $\underline{\underline{A}}(\mu)$ in (6) in terms of the coefficient μ. However, there are many cases for which the dependency on the parameter is more complicated, as for example, when associated with certain kinds of geometric variations. Extending the RB approach to the case of non-affine parametrization is feasible using suitable affine approximations. The computation of approximations $\sum_{m=1}^{M} \beta_m^M(\mu) \underline{\underline{A_m}}(x)$ for functions $\underline{\underline{A}}(x; \mu)$ (having in mind as an example the prototypical problem (6)), is a general problem of approximation. A possibility, introduced and further developed in [6, 27, 47] is to modify the standard greedy procedure described above, using interpolation. In short, the approach consists in selecting the coefficients $(\tilde{\beta}_m^M(\mu))_{m=1,\ldots,M}$ of the approximation $\mathcal{I}_M[g(\cdot; \mu)] := \sum_{m=1}^{M} \tilde{\beta}_m^M(\mu) g(\cdot; \mu_m^g)$ of order M to $g(\cdot; \mu)$ (g denoting here a general bilinear form, as in (18)–(19)) using an interpolation at the so-called *magic points* x_m selected sequentially with

$$x_1 \in \operatorname*{argmax}_{x \in \mathcal{D}} |g(\cdot; \mu_1^g)|,$$

$$x_m \in \operatorname*{argmax}_{x \in \mathcal{D}} |g(\cdot; \mu_m^g) - \mathcal{I}_{m-1}[g(\cdot; \mu_m^g)]|,$$

for all $m = 2, \ldots, M$. We refer to the contributions cited above for more details.

Nonlinear Elliptic Problems For the extension of the RB approach to *nonlinear* problems, one major difficulty is again the construction of appropriate *a posteriori* error estimators, which, additionally, need to be computed efficiently. Several examples of successful extensions are reported on in the literature [20, 32, 55, 56, 69, 85, 86]. But no general theory can of course be developed in the nonlinear context.

Semi-discretized Parabolic Problems After time-discretization, parametrized parabolic problems can be viewed as a collection of elliptic problems with the time variable as an additional parameter. A natural idea is then to build a reduced basis spanned by solutions for given values of the parameter and the time variable. Examples of contributions

are [26, 27]. This first approach has been improved by techniques combining the RB idea for the parameter with a proper orthogonal decomposition (POD) in the time variable, first introduced in [29] and further discussed in [37]. A route which would be interesting to follow could be to try to adapt on-the-fly, as time goes, the reduced basis which is the most adapted to the current time.

3 RB Approach for Boundary Value Problems with Stochastic Coefficients

This first application of the RB approach to a problem with stochastic coefficients is introduced in [14]. The purpose of this section is to overview this contribution, in particular showing how the general RB approach needs to be adapted to the specificities of the problem. We refer to [14] for all the details omitted below.

3.1 Setting of the Problem

Let us denote by $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space, and by $\omega \in \Omega$ the stochastic variable. We consider the stochastic field $U(\cdot, \omega)$ that is the almost sure solution to

$$-\mathrm{div}\big(\underline{\underline{A}}(x)\nabla U(x,\omega)\big) = 0, \quad \forall x \in \mathcal{D}, \tag{23}$$

supplied with a random Robin boundary condition

$$\underline{n}(x) \cdot \underline{\underline{A}}(x)\nabla U(x,\omega) + B(x,\omega)U(x,\omega) = g(x), \quad \forall x \in \partial\mathcal{D}. \tag{24}$$

In (24), the matrix $\underline{\underline{A}}(x)$ writes $\underline{\underline{A}}(x) = \sigma(x)\underline{\underline{I_d}}$ where $0 < \sigma(x) < \infty$ for a.e. $x \in \mathcal{D}$. Of course, $\underline{n}$ denotes the outward unit normal at the boundary of the smooth domain $\mathcal{D}$. The boundary is divided into three non-overlapping open subsets: $\partial\mathcal{D} = (\overline{\Gamma_N} \cup \overline{\Gamma_R} \cup \overline{\Gamma_B})$ (see Fig. 1). The boundary source term g is assumed to vanish everywhere except on Γ_R where it has constant unit value: $g(x) = 1_{\Gamma_R}, \forall x \in \partial\mathcal{D}$. The scalar random field $B(\cdot, \omega)$, parametrizing the boundary condition, also vanishes almost everywhere on the boundary $\partial\mathcal{D}$, except on some subset Γ_B of the boundary $\partial\mathcal{D}$ with non-zero measure, where $0 < \bar{b}_{\min} \leq B(\cdot, \omega) \leq \bar{b}_{\max} < \infty$ almost surely and almost everywhere. Note that on Γ_N, (24) thus reduces to homogeneous Neumann conditions. Physically, $U(\cdot, \omega)$ models the steady-state temperature field in a heat sink consisting of an isotropic material of thermal conductivity σ, contained in the domain $\mathcal{D}$. The sink is subject to zero heat flux on Γ_N, a constant flux on Γ_R modeling the heat source, and a convective heat transfer on Γ_B. The Biot number B models the effect of the exterior fluid convection on the solid thermal conduction problem inside $\mathcal{D}$. In real world engineering applications, the value of B is only approximately known. It is therefore legitimate to encode the uncertainties on B using a random field $B(\cdot, \omega)$, see [42] for more details.

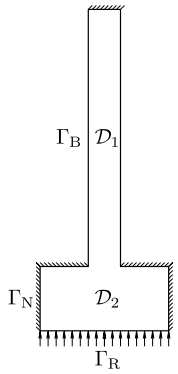

Fig. 1 $\mathcal{D}$ has the geometry of a (piece of) heat sink: a spreader $\mathcal{D}_2$ with a fin $\mathcal{D}_1$ on top

Correspondingly, the solution to (23)–(24), along with any output computed from this solution, are also random quantities. Only statistics on these quantities are relevant. We thus consider two *statistical* outputs for the problem: the expected value $\mathbf{E}(S)$ and the variance $\mathbf{Var}(S)$ of the random variable

$$S(\omega) = \mathcal{F}(U(\cdot, \omega)) = \int_{\Gamma_R} U(\cdot, \omega) \tag{25}$$

linearly depending on the trace of the solution $U(\cdot, \omega)$ on Γ_R.

A typical question, example of an *Uncertainty Quantification* problem, is to quantify the sensitivity of the output $S(\omega)$. Many existing contributions already addressed the issue: [4, 5, 18, 19, 24, 48, 64, 65].

A possible approach (which we will indeed adopt here) is to evaluate $\mathbf{E}(S)$ and $\mathbf{Var}(S)$ with the plain Monte-Carlo method using M independent random variables $(S^m)_{1 \leq m \leq M}$ with the same distribution law as S. The expectation and the variance are respectively approached by the empirical sums

$$E_M[(S^m)] = \frac{1}{M}\sum_{m=1}^{M} S^m,$$
$$V_M[(S^m)] = \frac{1}{M-1}\sum_{n=1}^{M}\left(S^n - E_M[(S^m)]\right)^2, \tag{26}$$

where the normalization factors used (respectively $\frac{1}{M}$ and $\frac{1}{M-1}$) allow, as is traditional in the community of Monte-Carlo methods, to have unbiased estimators: $\mathbf{E}\left(E_M[(S^m)]\right) = \mathbf{E}(S)$ and $\mathbf{E}\left(V_M[(S^m)]\right) = \mathbf{Var}(S)$ for all M. Large values of M are typically needed to obtain from (26) accurate approximations of $\mathbf{E}(S)$ and $\mathbf{Var}(S)$. Since, for each $m = 1, \ldots, M$, a new realization of the random parameter B is considered and the boundary value problem (23)–(24) has to be solved,

the task is clearly computationally demanding. It is a many-query context, appropriate for the application of the RB approach.

3.2 Discretization of the Problem

We begin by considering the Karhunen–Loève (abbreviated as KL) expansion

$$B(x,\omega) = \overline{b}\, G(x) + \overline{b} \sum_{k=1}^{\mathcal{K}} \Phi_k(x) Y_k(\omega) \tag{27}$$

of the coefficient $B(x,\omega)$ (see [35, 43, 80]). In (27), $\mathcal{K}$ denotes the (possibly infinite) rank of the covariance operator for $B(\cdot,\omega)$, which has eigenvectors $(\Phi_k)_{1\le k\le \mathcal{K}}$ and eigenvalues $(\lambda_k)_{1\le k\le \mathcal{K}}$ (sorted in decreasing order). The random variables $(Y_k)_{1\le k\le \mathcal{K}}$ are mutually uncorrelated in $L^2_{\mathbb{P}}(\Omega)$ with zero mean, G is supposed to be normalized $\int_{\partial\mathcal{D}} G = 1$ and $\overline{b} = \int_\Omega d\mathbb{P}(\omega) \int_{\partial\mathcal{D}} B(\cdot,\omega)$ is a fixed intensity factor.

Based on (27), we introduce the deterministic function

$$b(x,y) = \overline{b} G(x) + \overline{b} \sum_{k=1}^{\mathcal{K}} \Phi_k(x) y_k \tag{28}$$

defined for almost all $x \in \partial\mathcal{D}$ and all $y \in \Lambda^y \subset \mathbb{R}^{\mathcal{K}}$, where Λ^y denotes the range of the sequence $Y = (Y_k)_{1\le k\le \mathcal{K}}$ of random variables appearing in (27). Notice that $B(x,\omega) = b(x, y(\omega))$.

It is next useful to consider, for any positive integer $K \le \mathcal{K}$, truncated versions of the expansions above, and to define, with obvious notation, $U_K(\cdot,\omega)$ as the solution to the problem (23)–(24) where $B(\cdot,\omega)$ is replaced by the truncated KL expansion $B_K(\cdot,\omega)$ at order K. Similarly, for all $y^K \in \Lambda^y$, $u_K(\cdot\,; y^K)$ is defined as the solution to

$$\begin{cases} -\mathrm{div}(\underline{\underline{A}}(x)\nabla u_K(x; y^K)) = 0, \quad \forall x \in \mathcal{D}, \\ \underline{n}(x) \cdot \underline{\underline{A}}(x)\nabla u_K(x; y^K) + b_K(x, y^K) u_K(x; y^K) \\ \qquad = g(x), \quad \forall x \in \partial\mathcal{D}, \end{cases} \tag{29}$$

where b_K is the K-truncated sum (28).

For a given integer $K \le \mathcal{K}$, we approximate the random variable $S(\omega)$ by $S_K(\omega) := \mathcal{F}(U_K(\cdot,\omega))$ where $U_K(\cdot,\omega) \equiv u_K(\cdot; Y^K(\omega))$, and the statistical outputs $\mathbf{E}(S_K)$ and $\mathbf{Var}(S_K)$ by the empirical sums

$$E_M[(S_K^m)] = \frac{1}{M}\sum_{m=1}^{M} S_K^m,$$

$$V_M[(S_K^m)] = \frac{1}{M-1}\sum_{n=1}^{M} \left(S_K^n - E_M[(S_K^m)]\right)^2, \tag{30}$$

using M independent realizations of the random vector Y^K. In practice, $u_K(\cdot\,; Y_m^K)$ is approached using, say, a finite element approximation $u_{K,\mathcal{N}}(\cdot\,; Y_m^K)$ with $\mathcal{N} \gg 1$ degrees of freedom. Repeating the task for M realizations of the K-dimensional random vector Y^K may be overwhelming, and this is where the RB approach comes into the picture. We now present the application of the RB approach to solve problem (29), parametrized by $y^K \in \Lambda^y$.

In echo to our presentation of Sect. 2, note that problem (29) is *affine* in the input parameter y^K thanks to the KL expansion (28) of b, which decouples the dependence on x and the other variables. To use the RB approach for this problem, we consider S in (30) as the output of the problem, the parameter being y^K (this parameter takes the values $Y^{K,m}$, $m \in \{1,\dots,M\}$ being the realization number of Y^K) and, as will become clear below, the offline stage is standard. On the other hand, in the online stage, the *a posteriori* estimation is completed to take into account the truncation error in K in (28).

Before we turn to this, we emphasize that we have performed above an *approximation* of the coefficient b, since we have truncated its KL expansion. The corresponding error should be estimated. In addition, the problem (29) after truncation might be ill-posed, even though the original problem (23)–(24) is well posed. To avoid any corresponding pathological issue, we consider a stochastic coefficient b having a KL expansion (28) that is positive for any truncation order K (which is a sufficient condition to ensure the well-posedness of (23)–(24)), and which converges absolutely a.e. in $\partial\mathcal{D}$ when $K \to \mathcal{K}$. For this purpose, (i) we require for $k = 1,\dots,\mathcal{K}$ a uniform bound $\|\Phi_k\|_{L^\infty(\Gamma_B)} \le \phi$, (ii) we set $Y_k := \Upsilon\sqrt{\lambda_k} Z_k$ with independent random variables Z_k uniformly distributed in the range $(-\sqrt{3},\sqrt{3})$, Υ being a positive coefficient, and (iii) we also ask $\sum_{k=1}^{\mathcal{K}} \sqrt{\lambda_k} < \infty$. Note that, if $\mathcal{K} = \infty$, condition (iii) imposes a sufficiently fast decay of the eigenvalues λ_k while k increases. We will see in Sect. 3.4 that this fast decay is also important for the practical success of our RB approach. Of course, (i)–(ii)–(iii) are arbitrary conditions that we impose for simplicity. Alternative settings are possible.

3.3 Reduced-Basis Ingredients

We know from Sect. 2 that two essential ingredients in the RB method are an *a posteriori* estimator and a greedy selection procedure. Like in most applications of the RB method, both ingredients have to be adapted to the specificities of the present context.

As mentioned above, the statistical outputs (30) require new *a posteriori* estimators. Moreover, the statistical outputs can only be computed after M queries Y_m^K, $m = 1,\dots,M$, in the parameter y^K, so these new *a posteriori* estimators cannot be used in the offline step.

The global error consists of two, independent contributions: the first one is related to the RB approximation, the second one is related to the truncation of the KL expansion.

In the greedy algorithm, we use a standard *a posteriori* estimation $|S^m_{K,\mathcal{N}} - S^m_{K,\mathcal{N},N}| \leq \Delta^s_{N,K}(Y^K_m)$ for the error between the finite element approximation $S^m_{K,\mathcal{N}} := \mathcal{F}(u_{K,\mathcal{N}}(\,\cdot\,; Y^K_m))$ and the RB approximation $S^m_{K,\mathcal{N},N} := \mathcal{F}(u_{K,\mathcal{N},N}(\,\cdot\,; Y^K_m))$ of S^m_K at a fixed truncation order K, for any realization $Y^K_m \in \Lambda^y$. This is classical [11, 54, 79] and similar to our example of Sect. 2, see [14] for details. Note however that the coercivity constant of the bilinear form for the variational formulation

$$\text{Find} \quad u(\cdot; y^K) \in H^1(\mathcal{D}) \quad \text{s.t.}$$
$$\int_{\mathcal{D}} \sigma \nabla u(\cdot; y^K) \cdot \nabla v + \int_{\Gamma_B} b(\cdot, y^K) u(\cdot; y^K) v$$
$$= \int_{\Gamma_R} g\, v, \quad \forall v \in H^1(\mathcal{D}) \tag{31}$$

of problem (29) depends on K. To avoid the additional computation of the coercivity constant for each K, we impose $b(x, y^K) \geq \overline{b} G(x)/2$, for all $x \in \Gamma_B$, and thus get a uniform lower bound for the coercivity constant. In practice, this imposes a limit $0 < \Upsilon \leq \Upsilon_{\max}$ on the intensity factor in the ranges of the random variables Y_k, thus on the random fluctuations of the stochastic coefficient, where $\Upsilon_{\max}$ is fixed for all $K \in \{0, \ldots, \mathcal{K}\}$

Let us now discuss the online *a posteriori* error estimation. As for the truncation error, an *a posteriori* estimation $|S^m_{\mathcal{N}} - S^m_{K,\mathcal{N}}| = |\mathcal{F}(u_{\mathcal{N}}(\,\cdot\,; Y^K_m)) - \mathcal{F}(u_{K,\mathcal{N}}(\,\cdot\,; Y^K_m))| \leq \Delta^t_{N,K}(Y^K_m)$ is derived in [14]. The error estimators $\Delta^s_{N,K}(Y^K_m)$ and $\Delta^t_{N,K}(Y^K_m)$, respectively for the RB approximation and the truncation, are eventually combined for $m = 1, \ldots, M$ to yield global error bounds in the Monte-Carlo estimations of the statistical outputs: $|E_M[(S^m_{K,\mathcal{N},N})] - E_M[(S^m_{\mathcal{N}})]| \leq \Delta_E((S^m_{K,\mathcal{N},N}))$ and $|V_M[(S^m_{K,\mathcal{N},N})] - V_M[(S^m_{\mathcal{N}})]| \leq \Delta_V((S^m_{K,\mathcal{N},N}))$. The control of the truncation error may be used to improve the performance of the reduced basis method. In particular, if the truncation error happens to be too small compared to the RB approximation error, the truncation rank $\mathcal{K}$ may be reduced.

3.4 Numerical Results

Our numerical simulations presented in [14] are performed on the steady heat conduction problem (23)–(24) inside the T-shaped heat sink $\mathcal{D} \subset \overline{\mathcal{D}_1} \cup \overline{\mathcal{D}_2}$ pictured in Fig. 1. The heat sink comprises a 2×1 rectangular substrate (spreader) $\mathcal{D}_2 \equiv (-1, 1) \times (0, 1)$ and a 0.5×4 thermal fin $\mathcal{D}_1 \equiv (-0.25, 0.25) \times (1, 5)$ on top. The diffusion coefficient is piecewise constant, $\sigma = 1_{\mathcal{D}_1} + \sigma_0\, 1_{\mathcal{D}_2}$, where $1_{\mathcal{D}_i}$ of course denotes the characteristic function of domain $\mathcal{D}_i$ ($i = 1, 2$). The finite element approximation is computed using quadratic finite elements on a regular mesh, with $\mathcal{N} = 6\,882$ degrees of freedom. The thermal coefficient is $\sigma_0 = 2.0$. To construct the random input field $B(\cdot, \omega)$, we consider the covariance function $\mathbf{Covar}(b(x, \omega)B(y, \omega)) = (\overline{b}\Upsilon)^2 \exp(-(x-y)^2/\delta^2)$ for $\overline{b} = 0.5$, $\Upsilon = 0.058$, and a correlation length $\delta = 0.5$. We perform its KL expansion and keep only the largest $\mathcal{K} = 25$ terms. We then fix $G(x) \equiv 1$ and the variables $Y_k(\omega)$, $1 \leq k \leq \mathcal{K}$, as independent, uniformly distributed random variables. This defines $B(\cdot, \omega)$ as the right-hand side of (27).

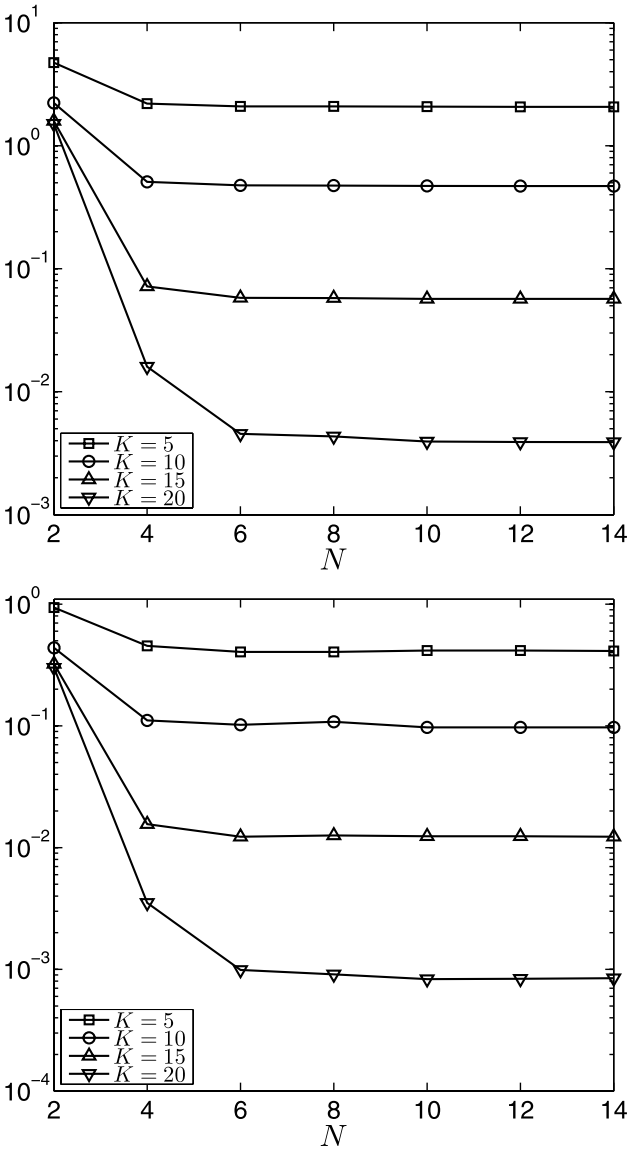

Fig. 2 Global error bounds for the RB approximation error and the KL truncation error of the output expectation (*top*: $\Delta_E((S^m_{K,\mathcal{N},N}))$) and of the output variance (*bottom*: $\Delta_V((S^m_{K,\mathcal{N},N}))$), as functions of the size $N = 2, \ldots, 14$ of the reduced basis, at different truncation orders $K = 5, 10, 15, 20$

After computing the reduced basis offline with our RB greedy algorithm on a trial sample of size $|\Lambda_{\text{trial}}| = 10\,000$, the global approximation error in the output Monte-Carlo sums $E_M[(S^m_{K,\mathcal{N},N})]$ and $V_M[(S^m_{K,\mathcal{N},N})]$ decays very fast (in fact, exponentially) with the size $N = 1, \ldots, 14$ of the reduced basis, see Fig. 2 with $K = 20$. Note that $M = 10\,000$ for the Monte-Carlo sums. We would also like to mention that these reduced bases have actually been obtained letting varying not only the parameter Y^K, but also additional parameters (namely the diffusion coefficient σ and the mean $\overline{b}$ of the Biot number) but this does not influence qualitatively the results presented here, and we omit this technical issue for simplicity (see [14] for more details).

It is observed that the global approximation error for truncated problems at a fixed order K and for various N (the size of the reduced basis) is quickly dominated by the truncation error. More precisely, beyond a critical value $N \geq N_{\text{crit}}(K)$, where $N_{\text{crit}}(K)$ is increasing with K, the global approximation error becomes constant. Notice that the approximation error is estimated as usual by *a posteriori* estimation techniques.

When $\mathcal{K}$ is infinite (or finite but huge), the control of the KL truncation error may be difficult. This is a general issue for problems involving a decomposition of the stochastic coefficient. Our RB approach is still efficient in some regimes with large K, but not all. In particular, a fast decay of the ranges of the parameters $(y_k)_{1\leq k\leq K}$ facilitates the exploration of Λ^y by the greedy algorithm, which allows in return to treat large K when the eigenvalues λ_k decay sufficiently fast with k.

In [14], we have decreased the correlation length to $\delta = 0.2$ and could treat up to $K = 45$ parameters, obtaining the results in a total computational time still fifty times as short as for the same Monte-Carlo sampling with direct finite element computations.

4 Variance Reduction Using an RB Approach

In this section, we present a variance reduction technique based upon an RB approach, which has been proposed recently in [13]. In short, the RB approximation is used as a control variate to reduce the variance of the original Monte-Carlo calculations.

4.1 Setting of the Problem

Suppose we need to compute repeatedly, for many values of the parameter $\lambda \in \Lambda$, the Monte-Carlo approximation (using an empirical mean) of the expectation $\mathbf{E}(Z^\lambda)$ of a functional

$$Z^\lambda = g^\lambda(X^\lambda_T) - \int_0^T f^\lambda(s, X^\lambda_s)\, ds \tag{32}$$

of the solutions $(X^\lambda_t, t \in [0,T])$ to the Stochastic Differential Equation (SDE)

$$X^\lambda_t = x + \int_0^t b^\lambda(s, X^\lambda_s)\, ds + \int_0^t \sigma^\lambda(s, X^\lambda_s) dB_s, \tag{33}$$

where $(B_t \in \mathbb{R}^d, t \in [0,T])$ is a d-dimensional standard Brownian motion. The parameter λ parametrizes the functions g^λ, f^λ, b^λ and σ^λ. In (33), we assume b^λ and σ^λ allow for the Itô processes $(X^\lambda_t \in \mathbb{R}^d, t \in [0,T])$ to be well defined, for every $\lambda \in \Lambda$. Notice that we have supplied the equation with the deterministic initial condition $X^\lambda_0 = x \in \mathbb{R}^d$. In addition, f^λ and g^λ are also assumed smooth, such that $Z^\lambda \in L^2(\Omega)$. Recall that a symbolic concise notation for (33) is

$$dX^\lambda_t = b^\lambda(t, X^\lambda_t)\, dt + \sigma^\lambda(t, X^\lambda_t)\, dB_t \quad \text{with } X^\lambda_0 = x.$$

Such parametrized problems are encountered in numerous applications, such as the calibration of the volatility in finance, or the molecular simulation of Brownian particles in materials science. For the applications in finance, $\mathbf{E}(Z^\lambda)$ is typically the price of an European option in the Black-Scholes model, and λ enters the diffusion term (the latter being called the volatility in this context). The calibration of the volatility consists in optimizing λ so that the prices observed on the market are close to the prices predicted by the model. Any optimization procedure requires the evaluation of $\mathbf{E}(Z^\lambda)$ for many values of λ. On the other hand, the typical application we have in mind in materials science is related to polymeric fluids modelling. There, $\mathbf{E}(Z^\lambda)$ is a stress tensor which enters the classical momentum conservation equation on velocity and pressure, and X^λ_t is a vector describing the configuration of the polymer chain, which evolves according to an overdamped Langevin equation, namely a stochastic differential equation such as (33). In this context, λ is typically the gradient of the velocity field surrounding the polymer chain at a given point in the fluid domain. The parameter λ enters the drift coefficient b^λ. The computation of the stress tensor has to be performed for each time step, and for many points in the fluid domain, which again defines a many-query context, well adapted to the RB approach. For more details on these two applications, we refer to [13, 39].

We consider the general form (32)–(33) of the problem and as output the Monte-Carlo estimation $\mathrm{E}_M[(Z^\lambda_m)] = \frac{1}{M}\sum_{m=1}^M Z^\lambda_m$ parametrized by $\lambda \in \Lambda$, where we recall (Z^λ_m) denotes i.i.d. random variables with the same law as Z^λ. These random variables are build in practice by considering a collection of realizations of (33), each one driven by a Brownian motion independent from the others. In view of the Central Limit Theorem, the rate at which the Monte-Carlo approximation $\mathrm{E}_M[(Z^\lambda_m)]$ approaches its limit $\mathbf{E}(Z^\lambda)$

is given by $\frac{1}{\sqrt{M}}$, the prefactor being proportional to the variance of Z^λ. A standard approach for reducing the amount of computations is therefore *variance reduction* [3, 10, 31, 49, 50, 66]. We focus on one particular variance reduction technique: the control variate method. It consists in introducing a so called *control variate* $Y^\lambda \in L^2(\Omega)$, assumed centered here for simplicity:

$$\mathbf{E}(Y^\lambda) = 0,$$

and in considering the equality:

$$\mathbf{E}(Z^\lambda) = \mathbf{E}(Z^\lambda - Y^\lambda).$$

The expectation $\mathbf{E}(Z^\lambda - Y^\lambda)$ is approximated by Monte-Carlo estimations $\mathrm{E}_M[(Z_m^\lambda - Y_m^\lambda)]$ which hopefully have, for a well chosen Y^λ, a smaller statistical error than direct Monte-Carlo estimations $\mathrm{E}_M[(Z_m^\lambda)]$ of $\mathbf{E}(Z^\lambda)$. More precisely, Y^λ is expected to be chosen so that $\mathbf{Var}(Z^\lambda) \gg \mathbf{Var}(Z^\lambda - Y^\lambda)$. The law of large numbers yields

$$\mathrm{E}_M[(Z_m^\lambda - Y_m^\lambda)] := \frac{1}{M}\sum_{m=1}^{M}(Z_m^\lambda - Y_m^\lambda) \xrightarrow[M\to\infty]{\mathbb{P}\text{-}a.s.} \mathbf{E}(Z^\lambda - Y^\lambda), \tag{34}$$

where, by the central limit theorem, the error is controlled by confidence intervals, in turns functions of the variance of the random variable at hand. The empirical variance

$$\mathrm{Var}_M\left((Z_m^\lambda - Y_m^\lambda)\right) := \frac{1}{M-1}\sum_{n=1}^{M}\left(Z_n^\lambda - Y_n^\lambda - \mathrm{E}_M((Z_m^\lambda - Y_m^\lambda))\right)^2 \tag{35}$$

which, as $M \to \infty$, converges to $\mathrm{Var}(Z^\lambda)$, yields a computable error bound. The Central Limit Theorem indeed states that: for all $a > 0$,

$$\mathbb{P}\left(\left|\mathbf{E}(Z^\lambda - Y^\lambda) - \mathrm{E}_M\left((Z_m^\lambda - Y_m^\lambda)\right)\right| \le a\sqrt{\frac{\mathrm{Var}_M((Z_m^\lambda - Y_m^\lambda))}{M}}\right) \xrightarrow[M\to\infty]{} \int_{-a}^{a}\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx. \tag{36}$$

Evaluating the empirical variance (35) is therefore an ingredient in Monte-Carlo computations, similar to what *a posteriori* estimates are for a deterministic problem.

Of course, the ideal control variate is, $\forall\lambda \in \Lambda$:

$$Y^\lambda = Z^\lambda - \mathbf{E}(Z^\lambda), \tag{37}$$

since then, $\mathbf{Var}(Z^\lambda - Y^\lambda) = 0$. This is however not a practical control variate since $\mathbf{E}(Z^\lambda)$ itself, the quantity we are trying to evaluate, is necessary to compute (37). Itô calculus shows that the optimal control variate (37) also writes:

$$Y^\lambda = \int_0^T \nabla u^\lambda(s, X_s^\lambda)\cdot\sigma^\lambda(s, X_s^\lambda)dB_s, \tag{38}$$

where $u^\lambda(t, y) \in C^1([0, T], C^2(\mathbb{R}^d))$ satisfies the backward Kolmogorov equation [52]:

$$\begin{cases} \partial_t u^\lambda + b^\lambda\cdot\nabla u^\lambda + \frac{1}{2}\sigma^\lambda(\sigma^\lambda)^T : \nabla^2 u^\lambda = f^\lambda, \\ u^\lambda(T, \cdot) = g^\lambda(\cdot). \end{cases} \tag{39}$$

Even using this reformulation, the choice (37) is impractical since solving the partial differential equation (39) is at least as difficult as computing $\mathbf{E}(Z^\lambda)$. We will however explain now that both "impractical" approaches above may give birth to a practical variance reduction method, when they are combined with a RB type approximation.

Loosely speaking, the idea consists in: (i) in the offline stage, compute fine approximations of $\mathbf{E}(Z^\lambda)$ or respectively u^λ for some appropriate values of λ, in order to obtain fine approximations of the optimal control variate Y^λ (at those values) and (ii) in the online stage, for a new parameter λ, use as a control variate the best linear combination of the variables built offline.

4.2 Two Algorithms for Variance Reduction by the RB Approach

Using suitable time discretization methods [36], realizations of the stochastic process (33) and the corresponding functional (32) can be computed for any $\lambda \in \Lambda$, as precisely as needed. Leaving aside all technicalities related to time discretization, we thus focus on the Monte Carlo discretization.

We construct two algorithms, which can be outlined as follows.

Algorithm 1 (based on formulation (37)):

- *Offline stage*: Build an appropriate set of values $\{\lambda_1, \ldots, \lambda_N\}$ and, concurrently, for each $\lambda \in \{\lambda_1, \ldots, \lambda_N\}$ compute an accurate approximation $\mathrm{E}_{M_{\text{large}}}[(Z_m^\lambda)]$ of $\mathbf{E}(Z^\lambda)$ (for a very large number M_{large} of realizations). At the end of the offline step, accurate approximations

  $$\tilde{Y}^\lambda = Z^\lambda - \mathrm{E}_{M_{\text{large}}}[Z_m^\lambda]$$

  of the optimal control variate Y^λ are at hand. The set of values $\{\lambda_1, \ldots, \lambda_N\}$ is chosen in order to ensure the maximal variance reduction in the forthcoming online computations (see below for more details).
- *Online stage*: For any $\lambda \in \Lambda$, compute a control variate $\tilde{Y}_N^\lambda$ for the Monte-Carlo estimation of $\mathbf{E}(Z^\lambda)$ as a linear combination of

  $$(\bar{Y}_i = \tilde{Y}^{\lambda_i})_{1\le i\le N}.$$

Algorithm 2 (based on formulation (38)):

- *Offline stage*: Build an appropriate set of values $\{\lambda_1, \ldots, \lambda_N\}$ and, concurrently, for each $\lambda \in \{\lambda_1, \ldots, \lambda_N\}$, compute an accurate approximation $\tilde{u}^\lambda$ of u^λ, by solving the partial differential equation (39). The set of values $\{\lambda_1, \ldots, \lambda_N\}$ is chosen in order to ensure the maximal variance reduction in the forthcoming online computations (see below for more details).
- *Online stage*: For any $\lambda \in \Lambda$, compute a control variate $\tilde{Y}_N^\lambda$ for the Monte-Carlo estimation of $\mathbf{E}(Z^\lambda)$ as a linear combination of

$$\left(\bar{Y}_i = \int_0^T \nabla \tilde{u}^{\lambda_i}(s, X_s^\lambda) \cdot \sigma^\lambda(s, X_s^\lambda) dB_s\right)_{1 \le i \le N}.$$

In both algorithms, we denote by $\tilde{Y}_N^\lambda$ the control variate built online as a linear combinations of the $\bar{Y}_i$'s, the lowerscript index N emphasizing that the approximation is computed on a basis with N elements. An important practical ingredient in both algorithms is to use for the computation of Z^λ the exact same Brownian motions as those used to build the control variates.

The construction of set of values $\lambda \in \{\lambda_1, \ldots, \lambda_N\}$ in the offline stage of both algorithms is done using a greedy algorithm similar to those considered in the preceding sections. The only difference is that the error estimator used is the empirical variance. Before entering that, we need to make precise how the linear combinations are built online, since this linear combination construction is also used offline to choose the λ_i's.

The online stages of both Algorithms 1 and 2 follow the same line: for a given parameter value $\lambda \in \Lambda$, a control variate $\tilde{Y}_N^\lambda$ for Z_λ is built as an appropriate linear combination of the control variates $(\bar{Y}_i)_{1 \le i \le N}$ (obtained from the offline computations). The criterium used to select this appropriate combination is based on a minimization of the variance of the output:

$$\tilde{Y}_N^\lambda = \sum_{n=1}^N \alpha_n^* \bar{Y}_n, \tag{40}$$

where

$$(\alpha_n^*)_{1 \le n \le N} = \arg \min_{(\alpha_n)_{1 \le n \le N} \in \mathbb{R}^N} \mathbf{Var}\left(Z^\lambda - \sum_{n=1}^N \alpha_n \bar{Y}_n\right). \tag{41}$$

In practice the variance in (41) is of course replaced by its empirical approximation $\mathrm{Var}_{M_{\mathrm{small}}}$. Notice that we have an error estimate of the Monte Carlo approximation by considering $\mathrm{Var}_{M_{\mathrm{small}}}(Z^\lambda - \tilde{Y}_N^\lambda)$. It is easy to check that the least squares problem (41) is computationally inexpensive to solve since it amounts to solving a linear $N \times N$ system, with N small. More precisely, this linear system writes:

$$C_{M_{\mathrm{small}}} \alpha^* = b_{M_{\mathrm{small}}}$$

where α^* here denotes the vector with components α_n^*, $C_{M_{\mathrm{small}}}$ is a matrix with (i, j)-th entry

$$\mathrm{Cov}_{M_{\mathrm{small}}}(\bar{Y}_{i,m}, \bar{Y}_{j,m})$$

and $b_{M_{\mathrm{small}}}$ is a vector with j-th component

$$\mathrm{Cov}_{M_{\mathrm{small}}}(Z_m^\lambda, \bar{Y}_{j,m})$$

where for two collections of random variables U_m and V_m,

$$\mathrm{Cov}_M(U_m, V_m) = \frac{1}{M} \sum_{m=1}^M U_m V_m - \left(\frac{1}{M} \sum_{m=1}^M U_m\right)\left(\frac{1}{M} \sum_{m=1}^M V_m\right).$$

In summary, the computational complexity of one online evaluation is the sum of the computational cost of the construction of $b_{M_{\mathrm{small}}}$ (which scales like NM_{small}), and of the resolution of the linear system (which scales like N^2 for Algorithm 1 since the SVD decomposition of $C_{M_{\mathrm{small}}}$ may be precomputed offline, and scales like $N^3 M_{\mathrm{small}}$ for Algorithm 2, since the whole matrix $C_{M_{\mathrm{small}}}$ has to be recomputed for each new value of λ).

The greedy algorithms used in the offline stages follow the same line as in the classical RB approach. More precisely, for Algorithm 1, the offline stage writes: Let $\lambda_1 \in \Lambda_{\mathrm{trial}}$ be already chosen and compute $\mathrm{E}_{M_{\mathrm{large}}}(Z^{\lambda_1})$. Then, for $i = 1, \ldots, N-1$, for all $\lambda \in \Lambda_{\mathrm{trial}}$, compute $\tilde{Y}_i^\lambda$ and inexpensive approximations:

$$E_i(\lambda) := \mathrm{E}_{M_{\mathrm{small}}}(Z^\lambda - \tilde{Y}_i^\lambda) \quad \text{for } \mathbf{E}(Z^\lambda),$$

$$\epsilon_i(\lambda) := \mathrm{Var}_{M_{\mathrm{small}}}\left(Z^\lambda - \tilde{Y}_i^\lambda\right) \quad \text{for } \mathbf{Var}(Z^\lambda - \tilde{Y}_i^\lambda).$$

Select $\lambda_{i+1} \in \underset{\lambda \in \Lambda_{\mathrm{trial}} \setminus \{\lambda_j, j=1,\ldots,i\}}{\mathrm{argmax}} \{\epsilon_i(\lambda)\}$, and compute $\mathrm{E}_{M_{\mathrm{large}}}(Z^{\lambda_{i+1}})$.

In practice, the number N is determined such that $\epsilon_N(\lambda_{N+1}) \le \varepsilon$, for a given threshold ε. The greedy procedure for Algorithm 2 is similar.

4.3 Reduced-Basis Ingredients

The algorithms presented above to build a control variate using a reduced basis share many features with the classical RB approach. The approach follows a two-stage offline/online strategy. The reduced basis is built using snapshots (namely solutions for well chosen values of the parameters). An inexpensive error estimator is used both in the offline stage to build the reduced basis in the greedy algorithm, and in the online stage to check that the variance reduction is correct for new values of the parameters. The construction of

the linear combinations for the control variates is based on a minimization principle, which is reminiscent of the Galerkin procedure (3).

The practical efficiency observed on specific examples is similar for the two algorithms. They both satisfactorily reduce variance. Compared to the plain Monte Carlo method without variance reduction, the variance is divided at least by a factor 10^2, and typically by a factor 10^4. Algorithm 2 appears to be computationally much more demanding than Algorithm 1 and less general, since it requires the computation (and the storage) of an approximation of the solution to the backward Kolmogorov equation (39) for a few values of the parameter. In particular, Algorithm 2 seems impractical for high dimensional problems ($X_t^\lambda \in \mathbb{R}^d$ with d large). On the other hand, Algorithm 2 seems to be more robust with respect to the choice of Λ_{trial}: it yields good variance reduction even for large variations of the parameter λ, in the online stage. We refer to [13] for more details.

Notice also that Algorithm 1 is not restricted to a random variable Z^λ that is defined as a functional of a solution to a SDE. The approach can be generalized to any parametrized random variables, as long as there is a natural method to generate correlated samples for various values of the parameter. A natural setting for such a situation is the computation of a quantity $\mathbb{E}(g^\lambda(X))$ for a random variable X with given arbitrary law, *independent of the parameter* λ. In such a situation, it is easy to generate correlated samples by using the same realizations of the random variable X for various values of the parameter λ.

4.4 Numerical Results

The numerical results shown on Fig. 3 are taken from [13] and relate to the second application mentioned in the introduction, namely multiscale models for polymeric fluids (see [40] for a general introduction). In this context, the non-Newtonian stress tensor is defined by the Kramers formula as an expectation $\mathbf{E}(Z^\lambda)$ of the random variable:

$$Z^\lambda = X_T^\lambda \otimes F(X_T^\lambda), \tag{42}$$

where X_t^λ is a vector modelling the conformation of the polymer chain. The latter evolves according to an over-damped Langevin equation:

$$dX_t^\lambda = \left(\lambda\, X_t^\lambda - F(X_t^\lambda)\right) dt + dB_t. \tag{43}$$

Equation (43) holds at each position of the fluid domain, the parameter $\lambda \in \mathbb{R}^{d\times d}$ ($d = 2$ or 3) being the local instantaneous value of the velocity gradient field at the position considered. The evolution of the "end-to-end vector" X_t^λ is governed by three forces: a hydrodynamic force λX_t^λ, Brownian collisions B_t against the solvent molecules, and an entropic force $F(X_t^\lambda)$ specific to the polymer molecule. Typically, this entropic force reads either $F(X_t^\lambda) = X_t^\lambda$ (for the Hookean dumbbells), or $F(X_t^\lambda) = \frac{X_t^\lambda}{1-|X_t^\lambda|^2/b}$ (for the Finitely-Extensible Nonlinear Elastic (FENE) dumbells, assuming $|X_t^\lambda| < \sqrt{b}$).

The numerical simulations of the flow evolution of a polymeric fluid using such a model typically consist, on many successive time slots $[nT, (n+1)T]$, of two steps: (i) the computation of (43), for a given gradient velocity

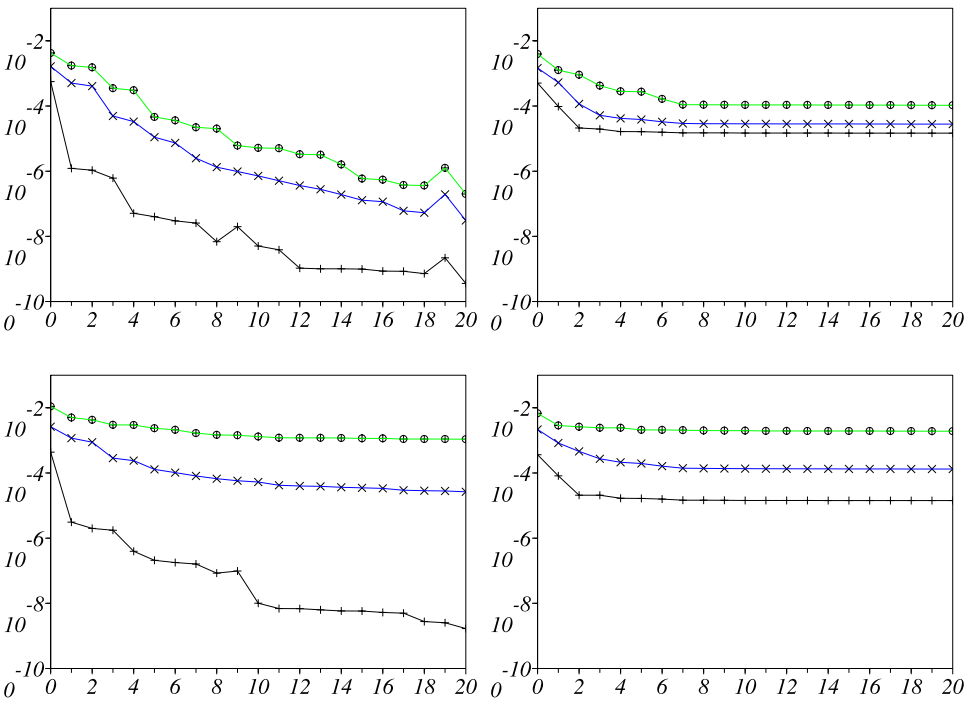

Fig. 3 Algorithm 1 (*left*) and 2 (*right*) for FENE model with $b = 16$. The x-axis is the size N of the reduced basis. We represent the minimum $+$, mean $\times$ and maximum $\circ$ of $\mathrm{Var}_M[Z^\lambda - \tilde{Y}_N^\lambda]/\mathrm{E}_M[Z^\lambda - \tilde{Y}_N^\lambda]^2$ over online test samples $\Lambda_{\text{test}} \subset \Lambda$ (*top*) and $\Lambda_{\text{test wide}} \supset \Lambda$ (*bottom*) of parameters

field λ, at many points of the fluid domain (think of the nodes of a finite element mesh) and (ii) the computation of a new velocity gradient field in the fluid domain, for a given value of the non-Newtonian stress tensor, by solving the classical momentum and mass conservation equations, which we omit here for brevity. Thus, $\mathbf{E}(Z^\lambda)$ has to be computed for many values λ corresponding to many spatial positions and many possible velocity fields at each such positions in the fluid domain.

In the numerical simulations of Fig. 3, the SDE (43) for FENE dumbbells when $d = 2$ is discretized with the Euler-Maruyama scheme using 100 iterations with a constant time step $\Delta t = 10^{-2}$ starting from a deterministic initial condition $x = (1, 1)$. Reflecting boundary conditions are imposed on the boundary of the ball with radius $\sqrt{b}$. For $b = 16$ and $|\Lambda_{\text{trial}}| = 100$ trial parameter values randomly chosen in the cubic range $\Lambda = [-1, 1]^3$ (the traceless matrix $\underline{\underline{\lambda}}$ has entries $(\lambda_{11} = -\lambda_{22}, \lambda_{12}, \lambda_{21})$), a greedy algorithm is used to incrementally select $N = 20$ parameter values after solving $|\Lambda_{\text{trial}}| = 100$ least-squares problems (41) (with $M_{\text{small}} = 1000$) at each step of the greedy algorithm (one for each of the trial parameter values $\lambda \in \Lambda_{\text{trial}}$). Then, the $N = 20$ selected parameter values are used online for variance reduction of a test sample of $|\Lambda_{\text{test}}| = 1000$ random parameter values.

The variance reduction obtained online by Algorithm 1 with $M_{\text{large}} = 100\, M_{\text{small}}$ is very interesting, of about 4 orders of magnitude. For the Algorithm 2, we use the *exact* solution $\tilde{u}^\lambda$ to the Kolmogorov backward equation for Hookean dumbells as an approximation to u^λ solution to (39). This also yields satisfying variance reduction though apparently not as good as in Algorithm 1. As mentioned above, Algorithm 2 is computationally more demanding but seems to be slightly more robust than Algorithm 1 (namely when some online sample test $\Lambda_{\text{testwide}}$ uniformly distributed in $[-2, 2]^3$ extrapolates the trial sample used offline, see Fig. 3).

Our numerical tests, although preliminary, already show that the reiterated computations of parametrized Monte-Carlo estimations seem to be a promising opportunity of applications for RB approaches. More generally, even if RB approaches may not be accurate enough for some applications, they may be seen as good methods to obtain first estimates, which can then be used to construct more refined approximations (using variance reduction as mentioned here, or maybe preconditioning based on the coarse-grained RB model). This is perhaps the most important conclusion of the work described in this section.

5 Perspectives

The standard RB method has proved numerically efficient and reliable at reducing the cost of computations for the approximation of solutions to parametrized boundary value problems in numerous benchmark many-query frameworks. These accomplishments claim for a wider use of the RB method in more realistic settings, and even suggest that some RB ideas could still be extended in numerous many-query frameworks yet largely unexplored, including the stochastic context. The success of the RB approach in parametrized boundary value problems is only understood precisely from a mathematical perspective in a few very simple cases. This should motivate further theoretical investigations.

In this section, we discuss various tracks for the development of reduced basis techniques, both from a methodological viewpoint and in terms of possible applications, with a focus on the stochastic context presented above.

5.1 A Posteriori Estimation in the Stochastic Context

We already emphasized that a crucial ingredient in the RB approach is an accurate and fast *a posteriori estimator* for the approximation error between two levels of discretization (the initially discretized, non-reduced one and the reduced one). Therefore, before everything, the future developments of the RB method should definitely concentrate on improving the *a posteriori* estimators. In particular, for the new contexts of application that are *stochastic*, there seems to remain some room for a yet better understanding of the *a posteriori* error estimation. More precisely, the best way to evaluate the reduction error when the Galerkin approximations (used by deterministic applications) are replaced with Monte-Carlo approximations is still unclear. For a first application of the RB ideas to stochastic applications, we have used confidence intervals as a probabilistic measure of the Monte-Carlo approximation error. These confidence intervals are only reliable in the limit of infinitely many realizations of the random variables. But there are other possibilities, like using non-asymptotic upper-bounds for the error which hold whatever the number of realizations (using for example Chebyshev inequalities or Berry-Esseen type bounds). In addition, the numerical evaluation of the variance is not obvious either. Until now, we have used Monte-Carlo estimators, but there exist other possibilities too which could be faster or more accurate and should thus be tested. Finally, another idea related to the method presented in Sect. 4 would be to mimick the usual RB approach, by considering that the reference result is the one obtained with M_{large} realizations, and to develop *a posteriori* error bounds with respect to this reference solution (using for example conditional expectations with respect to the M_{large} realizations).

5.2 Affine Decompositions and the Stochastic Context

As explained above, the RB approach is to date only efficient at yielding computational reductions in the context of

affine parametrization. However, as shown in the previous sections, a many-query parametrized framework is not necessarily parametrized in an affine way. So one may have to pretreat the problem in order to transform it as the limit of a sequence of affinely-parametrized problems. It would thus be interesting to derive rapidly convergent affine approximations for non-affine problems. For instance, the Karhunen-Loeve decomposition used to pretreat a random field entering a partial differential equation as a coefficient may converge too slowly for an efficient use of the RB method applied to truncated decompositions, in the context of random fields with small correlation lengths. One should then look for other possible affine representation of the random variations in the input coefficient. Now, there are many possible tracks to solve this problem, like projecting the random field on a well-chosen basis for the oscillation modes of the coefficient for instance (many possible bases may exist for the realizations of the random field, depending on its regularity), interpolating (recall the so-called *empirical interpolation* with magic points), homogenizing fast oscillations (see the preliminary RB approach to homogenization [11]), etc.

5.3 Application to Bayesian Statistics

A context where the RB ideas could be applied is *Bayesian statistics* where the many-query parametrized framework is naturally encountered. We now make this more precise by presenting a specific example that would be well-suited for the application of Algorithm 1 in Sect. 4.

Let us consider, for given values of a parameter μ_1, an ensemble of observations $(x_i^{\mu_1})_{1\le i\le N_{\text{data}}^{\mu_1}}$. Following the Bayesian framework, a stochastic model is proposed to model the observation: the quantities $(x_i^{\mu_1})_{1\le i\le N_{\text{data}}^{\mu_1}}$ are supposed to form a set of independent and identically distributed samples following a given distribution parametrized by another set of parameters μ_2 (think for example of a mixture of Gaussians, μ_2 being then the triplets of weights, means and variances of each Gaussians). The Bayesian approach then consists in postulating a so-called *prior distribution* (with a probability density function denoted $\text{Prior}(\mu_2)$ below) on the parameters μ_2, and to compute the so-called *posterior distribution*, namely the distribution of μ_2 given the observations (with a probability density function denoted $\Pi(\mu_2|(x_i^{\mu_1})_{1\le i\le N_{\text{data}}^{\mu_1}})$ below). Of course, the posterior distribution for μ_2 is expected to depend on μ_1: for each μ_1, the aim is thus to sample the probability measure $\Pi(\mu_2|(x_i^{\mu_1})_{1\le i\le N_{\text{data}}^{\mu_1}})\,d\mu_2$, with

$$\Pi\big(\mu_2|(x_i^{\mu_1})_{1\le i\le N_{\text{data}}^{\mu_1}}\big) = (Z^{\mu_1})^{-1}\Pi\big((x_i^{\mu_1})_{1\le i\le N_{\text{data}}^{\mu_1}}|\mu_2\big)\,\text{Prior}(\mu_2)$$

where Z^{μ_1} is the normalization constant, and $\Pi((x_i^{\mu_1})_{1\le i\le N_{\text{data}}^{\mu_1}}|\mu_2)$ is the so-called likelihood function, namely the probability density function of the observations given the datas. One possible technique to sample the posterior distribution consists in drawing samples according to the prior distribution, and to weight each of them using the likelihood function, which depends on μ_1. With such a sampling technique, it is easy to draw correlated samples for various values of μ_1. Following Algorithm 1 in Sect. 4, it would thus be possible to build a reduced basis based on the sampling of the posterior distribution for some selected values of μ_1 (offline stage), in order to reduce the variance for the sampling of the posterior distribution for other generic values of μ_1 (online stage).

5.4 Relation to Functional Quantization

One computationally demanding stochastic context that defines a many-query framework is the approximation of the solution to a parametrized stochastic differential equation, for many values of the parameter. We already mentioned applications in finance and rheology in Sect. 4, where a variance reduction technique based on RB was proposed. Another idea consists in first computing precise discretizations of a few processes at some well-chosen parameter values, and then to use them for a faster computation of an approximation of the processes for other values of the parameter.

Functional quantization is an approach that has independently been developed along this line, see for instance [44, 67]. The idea of quantization is to approximate a square-integrable random variable with values in a Hilbert space by a random variable that takes a finite number of values, in an optimal way. In its simplest form, quantization deals with Gaussian random variables with values in $\mathbb{R}^d$, but it can also be applied to Gaussian processes. The numerical approach developed in [68] to solve stochastic differential equations is to first quantize the Brownian motion, and then to solve a collection of ordinary differential equations in order to recover approximations of the solutions to the stochastic differential equations as linear combinations of the ordinary differential equations solutions. Clearly, this approach for the discretization of stochastic differential equations has intimate connection with a RB approach. In particular, the computations are split into two parts: an offline step, which is computationally expensive, to quantize the Brownian motion, and then an online step to solve ordinary differential equations rather than stochastic differential equations.

In a setting where the stochastic differential equations are parametrized, a natural similar idea would be to quantize the solution to the stochastic differential equations for a few values of the parameter, and next to build the solution to the stochastic differential equation for another value of the parameter as a linear combination of these precomputed solutions.

References

1. Ainsworth M, Oden JT (2000) A posteriori error estimation in finite element analysis. Wiley-Interscience, New York
2. Almroth BO, Stern P, Brogan FA (1978) Automatic choice of global shape functions in structural analysis. AIAA J 16:525–528
3. Arouna B (2004) Robbins-Monroe algorithms and variance reduction in finance. J Comput Finance 7(2):35–62
4. Babuška I, Tempone R, Zouraris G (2005) Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. Comput Methods Appl Mech Eng 194:1251–1294
5. Babuška I, Nobile F, Tempone R (2007) A stochastic collocation method for elliptic partial differential equations with random input data. SIAM J Numer Anal 45(3):1005–1034
6. Barrault M, Nguyen NC, Maday Y, Patera AT (2004) An "empirical interpolation" method: Application to efficient reduced-basis discretization of partial differential equations. C R Acad Sci Paris, Sér I 339:667–672
7. Barrett A, Reddien G (1995) On the reduced basis method. Z Angew Math Mech 75(7):543–549
8. Becker R, Rannacher R (2001) An optimal control approach to a posteriori error estimation in finite element methods. Acta Numer 37:1–225
9. Binev P, Cohen A, Dahmen W, DeVore R, Petrova G, Wojtaszczyk P (2010) Convergence rate for greedy algorithms in reduced basis methods, submitted for publication (preprint available at www.sciencesmath-paris.fr/uploaded/dahmen.pdf)
10. Bonvin J, Picasso M (1999) Variance reduction methods for CONNFFESSIT-like simulations. J Non-Newton Fluid Mech 84:191–215
11. Boyaval S (2008) Reduced-basis approach for homogenization beyond the periodic setting. SIAM Multiscale Model Simul 7(1):466–494
12. Boyaval S (2009) Mathematical modeling and simulation for material science. PhD thesis, Université Paris Est
13. Boyaval S, Lelièvre T (2009) A variance reduction method for parametrized stochastic differential equations using the reduced basis paradigm. In: Zhang P (ed) Communication in mathematical sciences, Special issue "Mathematical issues on complex fluids"
14. Boyaval S, Le Bris C, Maday Y, Nguyen NC, Patera AT (2009) A reduced basis approach for variational problems with stochastic parameters: Application to heat conduction with variable Robin coefficient. Comput Methods Appl Mech Eng 198(41–44):3187–3206
15. Buffa A, Maday Y, Patera AT, Prud'homme C, Turinici G A priori convergence of the greedy algorithm for the parametrized reduced basis. M2AN (Math Model Numer Anal) (preprint available at http://augustine.mit.edu/methodology/papers/atp_M2AN_preprint_Nov09.pdf)
16. Burkardt J, Gunzburger MD, Lee HC (2006) POD and CVT-based reduced order modeling of Navier-Stokes flows. Comput Methods Appl Mech 196:337–355
17. Ciarlet Ph (1978) The finite element method for elliptic problems. North-Holland, Amsterdam
18. Deb MK, Babuška IM, Oden JT (2001) Solution of stochastic partial differential equations using Galerkin finite element techniques. Comput Methods Appl Mech Eng 190(48):6359–6372
19. Debusschere BJ, Najm HN, Pebay PP, Knio OM, Ghanem RG, Le Maître OP (2004) Numerical challenges in the use of polynomial chaos representations for stochastic processes. SIAM J Sci Comput 26(2):698–719
20. Deparis S (2009) Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach. SIAM J Numer Anal 46(4):2039–2067
21. Devore RA (1993) Constructive approximation. Acta Numer 7:51–150
22. Fink JP, Rheinboldt WC (1983) On the error behavior of the reduced basis technique for nonlinear finite element approximations. Z Angew Math Mech 63(1):21–28
23. Fox RL, Miura H (1971) An approximate analysis technique for design calculations. AIAA J 9(1):177–179
24. Ghanem RG, Spanos PD (1991) Stochastic finite elements: A spectral approach. Dover, New York. Revised edition, Springer, New York (2003)
25. Golub G, van Loan C (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, London
26. Grepl MA, Patera AT (2005) A Posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. M2AN (Math Model Numer Anal) 39(1):157–181
27. Grepl MA, Maday Y, Nguyen NC, Patera AT (2007) Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. M2AN (Math Model Numer Anal) 41(2):575–605. doi:10.1051/m2an:2007031
28. Gunzburger MD (1989) Finite element methods for viscous incompressible flows. Academic Press, New York
29. Haasdonk B, Ohlberger M (2008) Reduced basis method for finite volume approximations of parametrized linear evolution equations. M2AN (Math Model Numer Anal) 42(3):277–302
30. Haasdonk B, Ohlberger M (2008) Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. In: Proc 5th international symposium on finite volumes for complex applications, June 08–13, 2008, Aussois, France, pp 471–478
31. Hammersley J, Handscomb D (eds) (1964) Monte Carlo methods. Chapman and Hall Ltd, London
32. Huynh DBP, Patera AT (2007) Reduced-basis approximation and a posteriori error estimation for stress intensity factors. Int J Numer Methods Eng 72(10):1219–1259. doi:10.1002/nme.2090
33. Huynh DBP, Rozza G, Sen S, Patera AT (2007) A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. C R Math Acad Sci Paris 345:473–478
34. Huynh DBP, Knezevic DJ, Chen Y, Hesthaven JS, Patera AT (2010) A natural-norm successive constraint method for inf-sup lower bounds. Comput Methods Appl Mech Eng 199:1963–1975. doi:10.1016/j.cma.2010.02.011
35. Karhunen K (1946) Zur spektraltheorie stochastischer prozesse. Ann Acad Sci Fenn 37
36. Kloeden P, Platen E (2000) Numerical solution of stochastic differential equations. Springer, Berlin
37. Knezevic DJ, Patera AT (2010) A certified reduced basis method for the Fokker-Planck equation of dilute polymeric fluids: FENE dumbbells in extensional flow. SIAM J Sci Comput 32(2):793–817. doi:10.1137/090759239
38. Kunisch K, Volkwein S (2002) Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J Numer Anal 40(2):492–515
39. Le Bris C (2005) Systèmes multi-échelles : Modélisation & simulation. Mathématiques & applications, vol 47. Springer, Berlin
40. Le Bris C, Lelièvre T (2009) Multiscale modelling of complex fluids: A mathematical initiation. In: Engquist B, Lötstedt P, Runborg O (eds) Multiscale modeling and simulation in science series. LNCSE, vol 66. Springer, Berlin, pp 49–138
41. Lee Lin MY (1991) Estimation of the error in the reduced-basis method solution of differential algebraic equations. SIAM J Numer Anal 28:512–528
42. Lienhard JH IV, Lienhard JH V, (2002) A heat transfer textbook. Phlogiston Press, Cambridge
43. Loève M (1978) Probability theory, vol I–II. Springer, New York
44. Luschy H, Pagès G (2002) Functional quantization of Gaussian processes. J Funct Anal 196:486–531

45. Maday Y (2006) Reduced-basis method for the rapid and reliable solution of partial differential equations. In: Proceedings of international conference of mathematicians, Madrid. European Mathematical Society, Zurich
46. Maday Y, Patera AT, Turinici G (2002) A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. J Sci Comput 17(1–4):437–446
47. Maday Y, Nguyen NC, Patera AT, Pau G (2009) A general, multipurpose interpolation procedure: the magic points. Commun Pure Appl Anal 8(1):383–404
48. Matthies HG, Keese A (2005) Galerkin methods for linear and nonlinear elliptic stochastic pdes. Comput Methods Appl Mech Eng 194:1295–1331
49. Melchior M, Öttinger HC (1995) Variance reduced simulations of stochastic differential equations. J Chem Phys 103:9506–9509
50. Milstein GN, Tretyakov MV (2006) Practical variance reduction via regression for simulating diffusions. Technical Report MA-06-19, School of Mathematics and Computer Science, University of Leicester
51. Nagy DA (1979) Modal representation of geometrically nonlinear behaviour by the finite element method. Comput Struct 10:683–688
52. Newton NJ (1994) Variance reduction for simulated diffusions. SIAM J Appl Math 54(6):1780–1805
53. Nguyen NC (2007) A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. J Comput Phys 227:9807–9822
54. Nguyen NC, Veroy K, Patera AT (2005) Certified real-time solution of parametrized partial differential equations. In: Yip S (ed) Springer, Berlin, pp 1523–1558
55. Nguyen NC, Rozza G, Huynh DBP, T Patera A (2009) Reduced basis approximation and a posteriori error estimation for parametrized parabolic pdes; application to real-time Bayesian parameter estimation. In: Biegler, Biros, Ghattas, Heinkenschloss, Keyes, Mallick, Tenorio, van Bloemen Waanders, Willcox (eds) Computational methods for large scale inverse problems and uncertainty quantification. Wiley, Berlin
56. Nguyen NC, Rozza G, Patera AT (2009) Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers equation. Calcolo 46(3):157–185
57. Noor AK (1981) Recent advances in reduction methods for nonlinear problems. Comput Struct 13:31–44
58. Noor AK (1982) On making large nonlinear problems small. Comput Methods Appl Mech Eng 34:955–985
59. Noor AK, Peters JM (1980) Reduced basis technique for nonlinear analysis of structures. AIAA J 18(4):455–462
60. Noor AK, Peters JM (1983) Multiple-parameter reduced basis technique for bifurcation and post-buckling analysis of composite plates. Int J Numer Methods Eng 19:1783–1803
61. Noor AK, Peters JM (1983) Recent advances in reduction methods for instability analysis of structures. Comput Struct 16:67–80
62. Noor AK, Balch CD, Shibut MA (1984) Reduction methods for non-linear steady-state thermal analysis. Int J Numer Methods Eng 20:1323–1348
63. Noor AK, Peters JM, Andersen CM (1984) Mixed models and reduction techniques for large-rotation nonlinear problems. Comput Methods Appl Mech Eng 44:67–89
64. Nouy A (2007) A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. Comput Methods Appl Mech Eng 196(45–48):4521–4537
65. Nouy A (2008) Generalized spectral decomposition method for solving stochastic finite element equations: Invariant subspace problem and dedicated algorithms. Comput Methods Appl Mech Eng 197(51–52):4718–4736
66. Öttinger HC, van den Brule BHAA, Hulsen MA (1997) Brownian configuration fields and variance reduced CONNFFESSIT. J Non-Newton Fluid Mech 70(30):255–261
67. Pagès G, Printems J (2005) Functional quantization for numerics with an application to option pricing. Monte Carlo Methods Appl 11(4):407–446
68. Pagès G, Sellami A (2008) Convergence of multi-dimensional quantized SDE's. http://arxiv.org/abs/0801.0726
69. Patera AT, Rønquist EM (2007) Reduced basis approximations and a posteriori error estimation for a Boltzmann model. Comput Methods Appl Mech Eng 196:2925–2942
70. Patera AT, Rozza G Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. Copyright MIT, 2006–2007. To appear in MIT Pappalardo monographs in mechanical engineering
71. Peterson JS (1989) The reduced basis method for incompressible viscous flow calculations. SIAM J Sci Stat Comput 10(4):777–786
72. Pinkus A (1985) n-widths in approximation theory. Springer, Berlin
73. Porsching TA (1985) Estimation of the error in the reduced basis method solution of nonlinear equations. Math Comput 45(172):487–496
74. Porsching TA, Lee Lin MY (1987) The reduced-basis method for initial value problems. SIAM J Numer Anal 24:1277–1287
75. Prud'homme C, Rovas D, Veroy K, Maday Y, Patera AT, Turinici G (2002) Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bounds methods. J Fluids Eng 124(1):70–80
76. Quarteroni A (2009) Numerical models for differential problems, vol 2. Springer, Berlin
77. Rheinboldt WC (1981) Numerical analysis of continuation methods for nonlinear structural problems. Comput Struct 13(1–3):103–113
78. Rheinboldt WC (1993) On the theory and error estimation of the reduced basis method for multi-parameter problems. Nonlinear Anal Theory Methods Appl 21(11):849–858
79. Rozza G, Huynh DBP, Patera AT (2008) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations—application to transport and continuum mechanics. Arch Comput Methods Eng 15(3):229–275. doi:10.1007/s11831-008-9019-9
80. Schwab C, Todor RA (2006) Karhunen-loève approximation of random fields by generalized fast multipole methods. J Comput Phys 217(1):100–122
81. Sen S (2008) Reduced-basis approximation and a posteriori error estimation for many-parameter heat conduction problems. Numer Heat Transf Part B, Fundam 54(5)
82. Sen S, Veroy K, Huynh DBP, Deparis S, Nguyen NC, Patera AT (2006) "Natural norm" a posteriori error estimators for reduced basis approximations. J Comput Phys 217:37–62
83. Strang G, Fix GJ (1973) An analysis of the finite element method. Prentice-Hall, New York
84. Temlyakov VN (2008) Nonlinear methods of approximation. Found Comput Math 3:33–107
85. Veroy K, Rovas D, Patera AT (2002) A Posteriori error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: "convex inverse" bound conditioners. ESAIM Control Optim Calc Var 8:1007–1028
86. Veroy K, Prud'homme C, Rovas DV, Patera AT (2003) A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In: Proceedings of the 16th AIAA computational fluid dynamics conference. Paper 2003-3847