Research Article

Multilevel a posteriori error estimator for greedy reduced basis algorithms

Marta Čertíková[1] · Liya Gaynutdinova[2] · Ivana Pultarová[2]

Received: 26 October 2019 / Accepted: 2 March 2020 / Published online: 12 March 2020

Abstract

The goal of reduced basis algorithms is to provide a relatively small set of functions which can serve as a basis for sufficiently accurate and fast numerical solution of a parametrized problem for any choice of parameters. Such methods are often employed in parameter identification problems detecting, for instance, material qualities in diffusion problems, elasticity, or in Maxwell equations, or in time dependent problems where time plays the role of the parameter. We deal with greedy reduced basis algorithms. An important part of these algorithms is to estimate the difference between the exact solution of a discretized problem and its projection onto the space spanned by a reduced basis. We introduce a new kind of the estimate, which is based on a multilevel splitting of a discretized solution space, and we compare it with a standard estimate based on bounds to coercivity and continuity constants. Two sided guaranteed bounds to the error can be obtained for both methods. Numerical complexity as well as memory consumption of both methods are comparable, while the multilevel method provides a more accurate spatial distribution of the error.

Keywords Reduced order modeling · Reduced basis algorithm · Greedy algorithm · A posteriori error estimate

Mathematics Subject Classification 65C20 · 65N15

1 Introduction

During recent decades, reduced order modeling has become a well-established tool in large scale numerical computing. Reduced basis (RB) methods turned out to be useful for reducing the computational complexity of parametrized (stochastic) or time dependent differential equations. Identification of parameters (so called inverse problems) usually needs a large number of solutions for many values of parameters [39]. An appropriate reduced model can significantly shorten solution time. Numerical methods for time dependent problems can employ sufficiently rich but small bases obtained from some solution history where the next time steps are to be solved. Within the RB methods, reduced order models are obtained during the preparatory *offline phase*, which allows us to efficiently compute relatively accurate solutions of the original-sized problems during the *online phase*. A large variety of methods providing reduced bases has been developed. The latest overview of the main issues concerning RB methods can be found in [31]. For example, in proper orthogonal decomposition methods [24] a singular value decomposition is used to choose the best subset from the whole set of possible discretized solutions. See, for example [41] for its adaptive form. A novelty method for a low-rank approximation to the solution of the stochastic Galerkin method is presented in [34]. In our paper, we deal with the so called *greedy reduced basis* (GRB) *algorithm* where

✉ Ivana Pultarová, ivana.pultarova@cvut.cz | [1]Department of Technical Mathematics, Faculty of Mechanical Engineering, Czech Technical University in Prague, Prague, Czech Republic. [2]Department of Mathematics, Faculty of Civil Engineering, Czech Technical University in Prague, Prague, Czech Republic.

SN Applied Sciences (2020) 2:614 | https://doi.org/10.1007/s42452-020-2409-9

the reduced basis is built up consecutively element by element [6, 20]. The choice of the function which has to be added to the current reduced basis depends on its contribution to the approximation quality of the new basis. Typically, such an element of the discretized solution manifold that achieves the largest distance from the span of the current reduced basis can be chosen. In practice, a sufficiently dense but finite set of *snapshots* is scanned instead of the whole manifold. Still, in every step of the GRB method it is necessary to check every element of this set which is computationally demanding. In our paper we focus on this part of the GRB algorithm. Especially, we introduce a new strategy for estimating the distance of a high-dimensional (*high-fidelity*) solution of the discretized parametric problem from the span of a current reduced basis. These error estimates are called *a posteriori error estimates*, see e.g. [18, 22], as they are only obtained after the inexact discretized solution is known. We should emphasize that here we consider the estimates of the distance between an approximate solution and the exact solution of the discretized problem contrary to many cases in literature where the name *a posteriori error estimate* serves for estimating the distance between an approximate and the exact solution of the underlying infinite-dimensional problem, cf. [1, 33].

The a posteriori error estimates are in fact estimates of algebraic errors of inexact discretized solutions. To obtain guaranteed bounds to these errors is in general a nontrivial problem; see, for example [33]. However, for special types of data, namely if the coefficient function of the problem is in the affine form, several approaches have been developed which provide guaranteed and feasible bounds to some norms of the error. The classical and mostly used estimates are based on estimating the dual norm of the residual scaled by (a bound to) the coercivity constant; see, for example [11, 18, 19, 38]. The complexity of the estimates does not depend on the size of the high fidelity problem; see, e.g. [18, 19]. However, the formula appears to be unstable in the case of large data or ill-conditioned problems. Rewriting the formula leads to better stability but increases memory consumption [11, 14, 15, 19]. Using some interpolation schemes [13–15] yields efficient but only approximate (not guaranteed) estimates. In recent paper [23], a hierarchical estimator has been introduced based on constructing some larger space than the reduced space and on a saturation assumption. Thus the obtained error bounds are not guaranteed as well. In all estimators mentioned so far, (a lower bound to) the coercivity constant must be employed. Obtaining it can be difficult in many applications; then the successive constrained method [26, 27] can appear as useful.

In this paper, the classical greedy RB algorithm is considered, but the a posteriori error estimate is new. We introduce a hierarchical estimate, the motivation for which comes from multilevel preconditioning methods, see, e.g. [3, 21]. There are several main differences between these the new and the classical estimators. While the classical upper bound estimator is defined as the H^1 dual norm of the residual scaled by the squared root of (a lower bound to) the coercivity constant, the new estimator approximates directly the energy norm of the error taking into account the actual values of parameters. This can be viewed as more physical. Moreover, the conjugate gradient method minimizes the energy norm of the error (on Krylov subspaces) as well. Of course, the dual norm of the residual is equivalent to the energy norm of the error, but for small and large coercivity and continuity constants, respectively, these two norms can provide rather different values. In some problems, accumulating the error in some small part of the solution domain can destroy validity of the solution, cf. [32]. The new estimator performs locally, thus the distribution of the error over the domain is well approximated. To gain all these benefits, we pay some extra effort during the offline phase; namely the complexity of the new estimator depends on the size of the high-fidelity problem. However, including adaptive enriching the training set for the RB algorithm, cf. [25], reduces the number of problems that have to be solved and makes the multilevel estimator competitive.

The goal of this paper is twofold. We suggest a new kind of a posteriori error estimates for the GRB algorithm and derive two-sided guaranteed bounds for the energy norm of the true error. The second goal of this paper is to present the GRB algorithm with both the standard and the new a posteriori error estimates not only in the form of general analytical formulas but also using linear algebra formulations to provide a clear connection between analytical and linear algebra objects. Our ideas are demonstrated on simple benchmark examples of parameter dependent elliptic differential equations. The outline of the paper is as follows. In the next section we describe the parameter dependent elliptic differential equation in the form which we are dealing with. In Sect. 3, the GRB method is described. In Sect. 4, the classical a posteriori error estimate (mean based, called MB) is described and the new one (multilevel, called ML) is introduced. In Sect. 5, the corresponding GBR algorithms are described and their computational and storage costs are discussed. The paper is concluded with numerical experiments and a short discussion.

2 Problem setting

In parameter dependent differential equations, the parameter space is usually a probability space equipped with a probability measure. Thus, let us start by introducing the classical probability framework for uncertainty quantification. The triple $\mathcal{T}_\Omega = (\Omega, \mathcal{F}, P)$ denotes a probability space where Ω is a sample space representing all possible outcomes, $\mathcal{F}$ is a σ-algebra in Ω and P is a probability measure on $\mathcal{F}$. We assume that the random input to the equation can be represented by a random vector $\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2, \dots \mathcal{X}_K)$, where $\mathcal{X}_k : \Omega \to \mathbb{R}$, $k = 1, \dots K$, are real-valued random variables defined in $\mathcal{T}_\Omega$.

Let us denote by $\Gamma \subseteq \mathbb{R}^K$ the image of $\mathcal{X}$, i.e. $\Gamma \subseteq \Pi_{k=1}^K \Gamma_k$, where each $\Gamma_k = \{\mathcal{X}_k(\omega), \omega \in \Omega\}$ is a measure space equipped with the measure $\rho_k(z)\,\mathrm{d}z$ induced by $\mathcal{X}_k$. Then, the system with a random input can be viewed as a parametric system which depends on the vector of parameters $\xi = ((\xi)_1, \dots (\xi)_K)$, $\xi \in \Gamma$. The values of ξ represent realizations of the random vector $\mathcal{X}$.

We solve the problem to find u dependent on $\boldsymbol{x} \in D$ and $\xi \in \Gamma$, $u(\cdot, \xi) \in W_0^{1,2}(D)$, such that

$$\begin{aligned}\int_D a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\\ = \int_D f(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \text{for all } v \in W_0^{1,2}(D),\ \xi \in \Gamma,\end{aligned} \tag{1}$$

where $D \subset \mathbb{R}^2$ is a bounded polygonal domain, $u(\boldsymbol{x}, \xi) = 0$ on $\delta D \times \Gamma$, $f \in L^2(D)$ and $a(\cdot, \xi) \in L^\infty(D)$ for $\xi \in \Gamma$. The gradient ∇ is considered with respect to the physical variable $\boldsymbol{x}$. In particular, we may search for some characteristics of the solution, like the mean $\mathrm{E}\, u(\boldsymbol{x}, \cdot)$ and the variance $\mathrm{Var}\, u(\boldsymbol{x}, \cdot)$ of $u(\boldsymbol{x}, \xi)$ for every $\boldsymbol{x} \in D$, or another quantity of interest. The coefficient $a(\boldsymbol{x}, \xi)$ in (1) is considered in the affine form

$$a(\boldsymbol{x}, \xi) = a_0(\boldsymbol{x}) + \sum_{k=1}^K (\xi)_k a_k(\boldsymbol{x}), \tag{2}$$

where $a_k(\boldsymbol{x}) \in L^\infty(D)$, $\xi = ((\xi)_1, \dots, (\xi)_K)$, $(\xi)_k \in \Gamma_k$. The function $a_0(\boldsymbol{x})$ may represent the mean value of $a(\boldsymbol{x}, \xi)$. We assume that there exist constants $0 < \alpha_1 \leq \alpha_2 < \infty$, such that

$$\alpha_1 \leq a(\boldsymbol{x}, \xi) \leq \alpha_2 \quad \text{for all } \boldsymbol{x} \in D,\ \xi \in \Gamma. \tag{3}$$

For the discretization of problem (1) with respect to the physical variable $\boldsymbol{x} \in D$, we employ the finite element (FE) method, see e.g. [9], with piece-wise bilinear basis functions $\psi_n(\boldsymbol{x})$, $n = 1, \dots, N$, using a grid of N inner nodes on D. Let us denote the N-dimensional span of these functions by V_N. This type of discretization is chosen for simplicity, the following ideas are also valid for other types of discretization. The discretized problem reads to find $u(\cdot, \xi) \in V_N$ such that

$$\begin{aligned}\int_D a(\boldsymbol{x}, \xi)\nabla u(\boldsymbol{x}, \xi)\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\\ = \int_D f(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \text{for all } v \in V_N,\ \xi \in \Gamma.\end{aligned} \tag{4}$$

Due to (2), the discretized problem (4) can be expressed in the matrix-vector form

$$\boldsymbol{A}(\xi)\boldsymbol{u}(\xi) := \left(\boldsymbol{A}_0 + \sum_{k=1}^K (\xi)_k \boldsymbol{A}_k\right)\boldsymbol{u}(\xi) = \boldsymbol{b}, \tag{5}$$

where $(\boldsymbol{A}_k)_{ns} = \int_D a_k(\boldsymbol{x})\nabla\psi_n(\boldsymbol{x})\nabla\psi_s(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, $\boldsymbol{b}_n = \int_D f(\boldsymbol{x})\psi_n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$, $k = 0, 1, \dots, K$, $n, s = 1, \dots, N$. The solution $\boldsymbol{u}(\xi) \in \mathbb{R}^N$ of (5) is the coefficient vector of the solution $u(\boldsymbol{x}, \xi) \in V_N$ of (4), which means that they are connected via $u(\boldsymbol{x}, \xi) = \sum_{n=1}^N \boldsymbol{u}_n(\xi)\psi_n(\boldsymbol{x})$.

Let us emphasize that instead of the usual solution space $W_0^{1,2}(D)$ equipped with the inner product

$$\int_D \nabla u(\boldsymbol{x})\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

and the related norm, we consider the same set of functions with the inner product and the norm defined as

$$(u, v)_V = \int_D a_0(\boldsymbol{x})\nabla u(\boldsymbol{x})\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \qquad \|u\|_V^2 = (u, u)_V, \tag{6}$$

respectively, and denote this vector space V. We assume that $\xi = (0, \dots, 0) \in \Gamma$, thus from (2) and (3) it follows that (6) are well defined.

In this paper, multi-valued objects, such as the vector ξ or the matrix $\boldsymbol{A}$, are denoted by bold letters. The order of any object in some sequence is marked by a subscript, e.g. $\xi_k, \boldsymbol{A}_k, \psi_k(\boldsymbol{x})$. To refer to a certain element of a multi-valued object, we use round brackets and subscripts. For example, $(\xi_j)_k$ is the k-th element of the vector ξ_j. We hope that all ambiguous terms, if they occur, can be understood from the context. Thus, for example, $\boldsymbol{A}(\xi)$ is a matrix depending on ξ, but $(\xi)_k\boldsymbol{A}$ is the matrix $\boldsymbol{A}$ multiplied by the scalar $(\xi)_k$. Some description of objects can be found in the place of superscripts, e.g. $\boldsymbol{u}^{\mathrm{RB}}(\xi)$ is the solution of a reduced problem parametrized by ξ. To emphasize the dependency on ξ, the notation such as $\boldsymbol{u}^{\mathrm{RB}}(\xi)$ can be used. But sometimes "(ξ)" is omitted for the sake of simplicity.

3 Reduced basis method

In order to provide a high-fidelity approximation of the solution of (1) for $\xi \in \Gamma$, the discretized problem (4)—or (5) in the matrix form—is generally required to be of a very large dimension. The solution manifold $\mathcal{M}_N = \{u(\boldsymbol{x}, \xi) \in V_N : \xi \in \Gamma\}$ of problem (4), however, can often be sufficiently approximated with a given error bound $\tau > 0$ by some *reduced space* $V_M \subset V_N$ with the dimension M much smaller than N. Any basis $\mathcal{U}_M = \{u_1(\boldsymbol{x}), \dots u_M(\boldsymbol{x})\}$ of V_M is called the *reduced basis*. When building such a basis during the offline phase we aim to find the smallest M and a set of M functions $\mathcal{U}_M$ such that for any $\xi \in \Gamma$ the distance of the solution $u(\boldsymbol{x}, \xi)$ of (4) from the span of $\mathcal{U}_M$ is less than τ. Let us emphasize that at least two meaningful metrics can be used. The first is generated by the metrics of V, i.e. $\|v\|_V^2 = \int_D a_0(\boldsymbol{x}) \nabla v(\boldsymbol{x})^2 \,\mathrm{d}\boldsymbol{x}$, and the second is the energy norm given by individual problems (4): $\|v\|_\xi^2 = \int_D a(\boldsymbol{x}, \xi) \nabla v(\boldsymbol{x})^2 \,\mathrm{d}\boldsymbol{x}$. See Sect. 3.1 for details.

In the matrix-vector form, the reduced basis can be represented by the matrix $\boldsymbol{U}_M \in \mathbb{R}^{N \times M}$ with columns $\{\boldsymbol{u}_1, \dots \boldsymbol{u}_M\}$ formed by the coefficient vectors with respect to the FE basis functions $\psi_1(\boldsymbol{x}), \dots, \psi_N(\boldsymbol{x})$ of the basis $\mathcal{U}_M = \{u_1(\boldsymbol{x}), \dots u_M(\boldsymbol{x})\}$. The name reduced basis will be used for $\mathcal{U}_M$ and for the coefficient vectors $\boldsymbol{U}_M$ as well. In the online phase, for each $\xi \in \Gamma$, the high-fidelity solution $\boldsymbol{u}(\xi)$ of (5) can now be approximated by $\boldsymbol{U}_M \boldsymbol{u}_M^{\mathrm{RB}}(\xi)$ where $\boldsymbol{u}_M^{\mathrm{RB}}(\xi)$ is the RB solution of the RB problem

$$\boldsymbol{A}^{\mathrm{RB}}(\xi) \boldsymbol{u}^{\mathrm{RB}}(\xi) := \left(\boldsymbol{A}_0^{\mathrm{RB}} + \sum_{k=1}^{K} (\xi)_k \boldsymbol{A}_k^{\mathrm{RB}} \right) \boldsymbol{u}^{\mathrm{RB}}(\xi) = \boldsymbol{b}^{\mathrm{RB}}, \qquad (7)$$

where $\boldsymbol{A}_k^{\mathrm{RB}} \in \mathbb{R}^{M \times M}$, $\boldsymbol{u}^{\mathrm{RB}}(\xi), \boldsymbol{b}^{\mathrm{RB}} \in \mathbb{R}^M$, $\boldsymbol{A}_k^{\mathrm{RB}} = \boldsymbol{U}_M^T \boldsymbol{A}_k \boldsymbol{U}_M$, $k = 0, 1, \dots, K$, and $\boldsymbol{b}^{\mathrm{RB}} = \boldsymbol{U}_M^T \boldsymbol{b}$. System (7) represents problem (5) restricted to the range of $\boldsymbol{U}_M$. Note that the range of $\boldsymbol{U}_N$ is equal to $\mathbb{R}^N$, the solution space of (5).

Evaluating the exact errors $\boldsymbol{u}(\xi) - \boldsymbol{U}_M \boldsymbol{u}_M^{\mathrm{RB}}(\xi)$ within the RB algorithm would unacceptably slow down the computation, and so only some estimates of the errors (called a posteriori error estimates) are employed. In the next three subsections we provide a definition of the error of the RB solution and its measurement, a detailed description of the greedy algorithm and we briefly comment on the choice of the set of snapshots $\boldsymbol{u}(\xi)$ associated to the choice of the *training set* of parameters $\xi \in \Gamma$.

3.1 Error of RB solution

Let us use the following notation. For all $u, v \in V$ let

$$\begin{aligned}
(u, v) &:= \int_D u(\boldsymbol{x}) v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \\
(u, v)_V &:= \int_D a_0(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \\
\|v\|_V^2 &:= \int_D a_0(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \\
(u, v)_\xi &:= \int_D a(\boldsymbol{x}, \xi) \nabla u(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \\
\|v\|_\xi^2 &:= \int_D a(\boldsymbol{x}, \xi) \nabla v(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}.
\end{aligned}$$

Note that in our setting

$$(u, v)_V = (u, v)_{\boldsymbol{0}} = \int_D a_0(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \nabla v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}, \quad u, v \in V.$$

Using the exact integration we have for the coefficient vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$ of functions $u, v \in V_N$

$$(u, v) = \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{u}, \quad (u, v)_V = \boldsymbol{v}^T \boldsymbol{A}_0 \boldsymbol{u}, \quad (u, v)_\xi = \boldsymbol{v}^T \boldsymbol{A}(\xi) \boldsymbol{u}$$

and corresponding dual norms can be expressed as

$$(u, v)_{V'} = \boldsymbol{v}^T \boldsymbol{A}_0^{-1} \boldsymbol{u}, \quad (u, v)_{\xi'} = \boldsymbol{v}^T (\boldsymbol{A}(\xi))^{-1} \boldsymbol{u},$$

where $\boldsymbol{M}$ is the mass matrix, $(\boldsymbol{M})_{jk} = \int_D \psi_j(\boldsymbol{x}) \psi_k(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$, $j, k = 1, \dots, N$, and $\boldsymbol{A}_0$ and $\boldsymbol{A}(\xi)$ are introduced in (5). Note that $\boldsymbol{b} = \boldsymbol{M}\boldsymbol{f}$, where $\boldsymbol{f}$ is the coefficient vector of a projection of the function $f \in L^2(D)$ to V_N: $(\boldsymbol{f})_j = \int_D \psi_j(\boldsymbol{x}) f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$, $j = 1, \dots, N$.

Let $u_N(\boldsymbol{x}, \xi) \in V_N$ be the exact solution of (4) and let $u_M(\boldsymbol{x}, \xi)$ be the RB solution in the reduced space $V_M \subset V_N$ for a given $\xi \in \Gamma$. Let $e_M(\boldsymbol{x}, \xi) = u_N(\boldsymbol{x}, \xi) - u_M(\boldsymbol{x}, \xi) \in V_N$ denote the error function of this RB solution. We measure the error in the *energy norm* generated by the underlying particular problem (4) and denote

$$E_M(\xi) = \|e_M(\boldsymbol{x}, \xi)\|_\xi^2 = \|u_N(\boldsymbol{x}, \xi) - u_M(\boldsymbol{x}, \xi)\|_\xi^2. \qquad (8)$$

In the matrix-vector form, let $\boldsymbol{u}_N(\xi) \in \mathbb{R}^N$ be the solution of (5) and let $\boldsymbol{u}_M(\xi) = \boldsymbol{U}_M \boldsymbol{u}^{\mathrm{RB}}(\xi)$ be the prolonged RB solution $\boldsymbol{u}^{\mathrm{RB}}(\xi) \in \mathbb{R}^M$ of (7). The coefficient vector of the error function $e_M(\boldsymbol{x}, \xi)$ is $\boldsymbol{e}_M(\xi) = \boldsymbol{u}_N(\xi) - \boldsymbol{u}_M(\xi)$. Denoting $\boldsymbol{r}_M(\xi) \in \mathbb{R}^N$ the residual vector of (5),

$$\boldsymbol{r}_M(\xi) = \boldsymbol{b} - \boldsymbol{A}(\xi) \boldsymbol{u}_M(\xi), \qquad (9)$$

we get

$$\boldsymbol{A}(\xi)\, \boldsymbol{e}_M(\xi) = \boldsymbol{A}(\xi)\, (\boldsymbol{u}_N(\xi) - \boldsymbol{u}_M(\xi)) = \boldsymbol{b} - \boldsymbol{A}(\xi) \boldsymbol{u}_M(\xi) = \boldsymbol{r}_M(\xi),$$

so the squared energy norm (8) of the error can be expressed as

$$\begin{aligned} E_M(\xi) &= \boldsymbol{e}_M(\xi)^T \boldsymbol{A}(\xi)\, \boldsymbol{e}_M(\xi) \\ &= (\boldsymbol{A}(\xi)\, \boldsymbol{e}_M(\xi))^T (\boldsymbol{A}(\xi))^{-1} \boldsymbol{A}(\xi)\, \boldsymbol{e}_M(\xi) \\ &= \boldsymbol{r}_M(\xi)^T\, (\boldsymbol{A}(\xi))^{-1} \boldsymbol{r}_M(\xi) \;=\; \|\boldsymbol{r}_M(\xi)\|^2_{\xi'}\,. \end{aligned} \tag{10}$$

3.2 Greedy algorithm

In the offline phase of building up the reduced basis, we employ the greedy algorithm [17, 18], which is based on two presumptions. First, that a good choice of the RB space is a space generated by some carefully chosen elements of the manifold $\mathcal{M}_N$ that are represented by M independent solutions $\{\boldsymbol{u}_1, \dots \boldsymbol{u}_M\}$ of (5) corresponding to a set of samples $T_M = \{\xi_1, \dots \xi_M\} \subset \Gamma$. Second, that we are able to characterize the whole manifold $\mathcal{M}_N$ sufficiently by a finite *trainig set* $T_J \subset \Gamma$, $J \gg M$, which has to be chosen in advance.

The solutions $\boldsymbol{u}(\xi_m)$ of (5) for all $\xi_m \in T_M$, $m = 1, \dots, M$, form the columns of the matrix $\boldsymbol{U}_M \in \mathbb{R}^{N\times M}$. The set $T_M \subset T_J$ is constructed by picking ξ from T_J one-by-one according to some criterion, and the matrix $\boldsymbol{U}_M$ is built column-by-column, correspondingly. The first element $\xi_1 \in T_J$ is usually chosen close to the "center" of Γ. Thus, $\boldsymbol{U}_1 = \boldsymbol{u}(\xi_1)$ and $T_1 = \{\xi_1\}$. Having $T_m = \{\xi_1, \dots \xi_m\}$ already selected and the corresponding matrix $\boldsymbol{U}_m$ constructed, the item ξ_{m+1} is considered to be such $\xi \in T_J$ that the corresponding solution $\boldsymbol{u}^{\mathrm{RB}}(\xi)$ of (7) maximizes the squared energy norm $E_m(\xi)$ of the error given by (10) for $M = m$. The process stops whenever $E_m(\xi) \le \tau$ for all $\xi \in T_J$. In practice, the columns of $\boldsymbol{U}_m$ are orthonormalized (in our numerical tests, orthonormalization with respect to the scalar product of $\mathbb{R}^N$ was used).

In other words, for $m = 1, 2, \dots, M$, the reduced problems (7) are solved for all currently remaining $\xi \in T_J \setminus T_m$, the obtained solutions $\boldsymbol{u}^{\mathrm{RB}}(\xi)$ are prolonged to the range of $\boldsymbol{U}_m$, $\boldsymbol{u}^{\mathrm{RB}}(\xi) \to \boldsymbol{U}_m \boldsymbol{u}^{\mathrm{RB}}(\xi)$, and the energy norms of the errors of the approximate solutions $\boldsymbol{U}_m \boldsymbol{u}^{\mathrm{RB}}(\xi)$ of (5) are compared. This means that $E_m(\xi)$ defined by (10) with M substituted by m has to be computed many times during the offline phase. Consequently, instead of computing the exact values of $E_m(\xi)$, some estimates of the errors are used, in which the inverse $\boldsymbol{A}(\xi)^{-1}$ is approximated. We focus on this in Sect. 4.

3.3 Choice of the training set

The training set $T_J \subset \Gamma$ of J elements should sufficiently represent the whole parameter space Γ and might allow efficient computation of the desired characteristics. To this end, various techniques are used to generate T_J such as the Monte Carlo method, or regular tensor grids over Γ, or sparse grids, see e.g. [30]. The coordinates of the points in T_J can be obtained, for example, as abscissas of some Gauss or Clenshaw–Curtis quadrature formulas. In this paper, the Monte Carlo method and the uniform probability distribution on Γ are considered. In general, the training set may be chosen according to the density distribution of the underlying probability space. A large training set T_J induces a lot of evaluations of (estimates of) $E_m(\xi)$. Starting with a relatively small training set and its adaptive enriching is proposed in [25]. This modification increases the efficiency of the methods of error estimates that we suggest, thus we employ this idea in our algorithms.

4 A posteriori error estimates

The key part of RB methods is estimating the quality of a current reduced basis or estimating which snapshot should be next included into the reduced basis. During the offline phase, formula (10) has to be evaluated or estimated for every $m = 1, 2, \dots, M$ and for all ξ remaining in the actual training set. Let us emphasize that the name *a posteriori error estimate* here and also, for example, in [17, 18, 22, 40], means the estimate of a difference (measured in some norm) between the solutions $u_M(\boldsymbol{x}, \xi) \in V_M \subset V_N$ and $u_N(\boldsymbol{x}, \xi) \in V_N$. These solutions are the $(\cdot,\cdot)_\xi$-orthogonal projections of the exact solution $u(\boldsymbol{x}, \xi) \in V$ of problem (1) onto V_M and V_N, respectively. However, a widely used meaning of the name *a posteriori error estimate* is connected to the estimate of the distance of some approximate solution $u_M(\boldsymbol{x}, \xi) \in V_M$ from the *exact* solution $u(\boldsymbol{x}, \xi) \in V$ of (1); see, for example, [1, 10]. Such an estimate cannot be obtained using only the discretized form of the problem. Some sophisticated construction is needed unless some other properties of the solutions are employed, such as in [10]. Before recalling the standard error estimate and introducing the new one in the next two subsections, we present the following lemma.

Lemma 4.1 *Let the matrices* $\boldsymbol{A}, \boldsymbol{M} \in \mathbb{R}^{N\times N}$ *be real, symmetric and positive definite. Suppose that there exist* $0 < c_1 \le c_2 < \infty$ *such that*

$$c_1 \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v} \le \boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} \le c_2 \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v} \quad \text{for all } \boldsymbol{v} \in \mathbb{R}^N. \tag{11}$$

Then

$$\frac{1}{c_2}\, \boldsymbol{v}^T \boldsymbol{M}^{-1} \boldsymbol{v} \le \boldsymbol{v}^T \boldsymbol{A}^{-1} \boldsymbol{v} \le \frac{1}{c_1}\, \boldsymbol{v}^T \boldsymbol{M}^{-1} \boldsymbol{v} \quad \text{for all } \boldsymbol{v} \in \mathbb{R}^N. \tag{12}$$

Moreover, the possible largest c_1 *and the smallest* c_2 *in* (11) *are equal to the minimal and maximal eigenvalue of* $\boldsymbol{M}^{-1}\boldsymbol{A}$*, respectively.*

Proof For any $\boldsymbol{v} \in \mathbb{R}^N$, inserting $\boldsymbol{v} = \boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{u}$ into (11), we have

$$c_1 \boldsymbol{u}^T\boldsymbol{u} \le \boldsymbol{u}^T\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{u} \le c_2 \boldsymbol{u}^T\boldsymbol{u} \quad \text{for all } \boldsymbol{u} \in \mathbb{R}^N, \tag{13}$$

so the eigenvalues of $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{M}^{-\frac{1}{2}}$ lie in $\langle c_1, c_2\rangle$. Then the spectra of the matrices $\boldsymbol{M}^{-1}\boldsymbol{A}$ and of $\boldsymbol{A}\boldsymbol{M}^{-1}$, both similar to $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{M}^{-\frac{1}{2}}$, lie in $\langle c_1, c_2\rangle$ as well. Then the spectra of the matrices $\boldsymbol{M}\boldsymbol{A}^{-1}$, $\boldsymbol{A}^{-1}\boldsymbol{M}$, and $\boldsymbol{M}^{\frac{1}{2}}\boldsymbol{A}^{-1}\boldsymbol{M}^{\frac{1}{2}}$ lie in $\langle \frac{1}{c_2}, \frac{1}{c_1}\rangle$, which yields (12). The last statement follows from the equivalence of (11) and (13) and from the similarity of $\boldsymbol{M}^{-1}\boldsymbol{A}$ and $\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{M}^{-\frac{1}{2}}$. □

4.1 Mean based a posteriori error estimate

A widely used estimate of the energy norm of error (10) is based on the dual norm of the residual

$$E_M^{\mathrm{MB}}(\xi) = \boldsymbol{r}_M^T(\xi)\,(\boldsymbol{A}_0)^{-1}\,\boldsymbol{r}_M(\xi) \; = \|\boldsymbol{r}_M(\xi)\|_{V'}^2\,, \tag{14}$$

see, e.g. [18], and it is called the *mean-based estimate* (MB) in this paper due to its relationship with the matrix $\boldsymbol{A}_0$ given by $a_0(x)$ which is often defined as the mean value of $a(x,\xi)$. Using (14) we can get guaranteed upper and lower bounds of (10). From coercivity and continuity conditions (3) for every $\xi \in \Gamma$ there exist $0 < \alpha_1^{\mathrm{cont}}(\xi) \le \alpha_2^{\mathrm{cont}}(\xi) < \infty$ such that

$$\alpha_1^{\mathrm{cont}}(\xi)\,\|v\|_V^2 \le (v,v)_\xi \le \alpha_2^{\mathrm{cont}}(\xi)\,\|v\|_V^2 \quad \text{for all } v \in V,\ \xi \in \Gamma, \tag{15}$$

where

$$\text{ess inf}_{\boldsymbol{x}\in D}\frac{a(\boldsymbol{x},\xi)}{a_0(\boldsymbol{x})} \le \alpha_1^{\mathrm{cont}}(\xi), \qquad \alpha_2^{\mathrm{cont}}(\xi) \le \text{ess sup}_{\boldsymbol{x}\in D}\frac{a(\boldsymbol{x},\xi)}{a_0(\boldsymbol{x})}.$$

Possibly, there may also exist uniform bounds $\bar{\alpha}_1^{\mathrm{cont}}$ and $\bar{\alpha}_2^{\mathrm{cont}}$ such that $0 < \bar{\alpha}_1^{\mathrm{cont}} \le \alpha_1^{\mathrm{cont}}(\xi) \le \alpha_2^{\mathrm{cont}}(\xi) \le \bar{\alpha}_2^{\mathrm{cont}} < \infty$ for all $\xi \in \Gamma$. If we consider the discretized problem only, $v \in V_N \subset V$, the inequalities (15) remain valid with possibly different constants,

$$\alpha_1^{\mathrm{disc}}(\xi)\,\boldsymbol{v}^T\boldsymbol{A}_0\boldsymbol{v} \le \boldsymbol{v}^T\boldsymbol{A}(\xi)\boldsymbol{v} \le \alpha_2^{\mathrm{disc}}(\xi)\,\boldsymbol{v}^T\boldsymbol{A}_0\boldsymbol{v}, \quad \text{for all } \boldsymbol{v} \in \mathbb{R}^N, \xi \in \Gamma, \tag{16}$$

where $\alpha_1^{\mathrm{cont}}(\xi) \le \alpha_1^{\mathrm{disc}}(\xi)$, $\alpha_2^{\mathrm{disc}}(\xi) \le \alpha_2^{\mathrm{cont}}(\xi)$. Obviously, cf. Lemma 4.1,

$$\alpha_1^{\mathrm{disc}}(\xi) = \lambda_{\min}(\boldsymbol{A}_0^{-1}\boldsymbol{A}(\xi)), \quad \alpha_2^{\mathrm{disc}}(\xi) = \lambda_{\max}(\boldsymbol{A}_0^{-1}\boldsymbol{A}(\xi)) \quad \text{for all } \xi \in \Gamma.$$

In our numerical experiments, there will exist uniform bounds $\bar{\alpha}_1^{\mathrm{disc}}$ and $\bar{\alpha}_2^{\mathrm{disc}}$ such that $\bar{\alpha}_1^{\mathrm{disc}} \le \alpha_1^{\mathrm{disc}}(\xi)$, $\alpha_2^{\mathrm{disc}}(\xi) \le \bar{\alpha}_2^{\mathrm{disc}}$ for all $\xi \in \Gamma$, where

$$\bar{\alpha}_1^{\mathrm{disc}} = \min_{\xi\in\Gamma}\lambda_{\min}(\boldsymbol{A}_0^{-1}\boldsymbol{A}(\xi)), \quad \bar{\alpha}_2^{\mathrm{disc}} = \max_{\xi\in\Gamma}\lambda_{\max}(\boldsymbol{A}_0^{-1}\boldsymbol{A}(\xi)).$$

Finally, we get

$$\begin{aligned} \bar{\alpha}_1^{\mathrm{cont}} &\le \bar{\alpha}_1^{\mathrm{disc}} \le \alpha_1^{\mathrm{disc}}(\xi), \\ \alpha_2^{\mathrm{disc}}(\xi) &\le \bar{\alpha}_2^{\mathrm{disc}} \le \bar{\alpha}_2^{\mathrm{cont}} \quad \text{for all } \xi \in \Gamma. \end{aligned} \tag{17}$$

Combining (16) and Lemma 4.1 we get two-sided guaranteed bounds to the squared energy norm (10) of the error

$$\frac{1}{\alpha_2^{\mathrm{disc}}(\xi)}\,\boldsymbol{r}_M(\xi)^T\boldsymbol{A}_0^{-1}\boldsymbol{r}_M(\xi) \le \boldsymbol{e}_M(\xi)^T\boldsymbol{A}(\xi)\boldsymbol{e}_M(\xi)^T \le \frac{1}{\alpha_1^{\mathrm{disc}}(\xi)}\,\boldsymbol{r}_M(\xi)^T\boldsymbol{A}_0^{-1}\boldsymbol{r}_M(\xi).$$

This can also be expressed using norms and dual norms as

$$\begin{aligned} \frac{\|\boldsymbol{r}_M(\xi)\|_{V'}^2}{\alpha_2^{\mathrm{cont}}(\xi)} \le \frac{\|\boldsymbol{r}_M(\xi)\|_{V'}^2}{\alpha_2^{\mathrm{disc}}(\xi)} &\le \|\boldsymbol{e}_M(\xi)\|_\xi^2 \\ \le \frac{\|\boldsymbol{r}_M(\xi)\|_{V'}^2}{\alpha_1^{\mathrm{disc}}(\xi)} &\le \frac{\|\boldsymbol{r}_M(\xi)\|_{V'}^2}{\alpha_1^{\mathrm{cont}}(\xi)}, \end{aligned} \tag{18}$$

cf. [8, (4-I.14)] or [18, (3.29)] for the upper bound.

In the GRB algorithm, instead of computing the constants $\alpha_1^{\mathrm{disc}}(\xi)$ and $\alpha_2^{\mathrm{disc}}(\xi)$ for every $\xi \in T_J$ separately, uniform lower and upper bounds of (17) can be used. However, if the variation of $a(\boldsymbol{x},\xi)$ with respect to ξ grows, these uniform bounds may become useless.

4.2 Multilevel a posteriori error estimate

The estimate of $E_M(\xi)$ suggested in this section is based on a hierarchy designed in the FE solution space. A full hierarchy is not necessary, only two or three levels can be sufficient. The theory of multilevel methods can be found in e.g. [3, 21]. An example of a specific FE setting (bilinear FE basis fubctions with rectangular supports) and the way of obtaining certain constants (γ, β_1, and β_2) and using them in a two-level error estimate can be found in Appendix. This scheme can be easily adapted to any other hierarchical FE setting. A multilevel method is then derived from the two-level scheme by recursive splitting of the coarse spaces which is described, e.g., in [28, 37]. Therefore, let us only consider a general hierarchical two-level splitting of the solution space V_N into two subspaces, the coarse and the fine one; see the definition in "Appendix". Let us denote the coefficient vectors of the function $u \in V_N$ with respect to the original FE space by $\boldsymbol{u} \in \mathbb{R}^N$ and with respect to the hierarchical two-level basis by

$$\boldsymbol{u}^{\mathrm{ML}} = \begin{pmatrix} \boldsymbol{u}^{\mathrm{ML,C}} \\ \boldsymbol{u}^{\mathrm{ML,F}} \end{pmatrix} \in \mathbb{R}^N, \quad \boldsymbol{P}\boldsymbol{u}^{\mathrm{ML}} = \boldsymbol{u}, \tag{19}$$

where we use the transformation matrix $\boldsymbol{P} \in \mathbb{R}^{N\times N}$ which is specified in "Appendix". Any system of linear equations $\boldsymbol{Au} = \boldsymbol{b}$ with respect to the original FE basis can be transformed into the system

$$\begin{aligned} \boldsymbol{A}^{\mathrm{ML}}\boldsymbol{u}^{\mathrm{ML}} &= \begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}} & \boldsymbol{A}^{\mathrm{ML,CF}} \\ \boldsymbol{A}^{\mathrm{ML,CF}^T} & \boldsymbol{A}^{\mathrm{ML,F}} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{\mathrm{ML,C}} \\ \boldsymbol{u}^{\mathrm{ML,F}} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{b}^{\mathrm{ML,C}} \\ \boldsymbol{b}^{\mathrm{ML,F}} \end{pmatrix} = \boldsymbol{b}^{\mathrm{ML}} \end{aligned} \tag{20}$$

where we have

$$\boldsymbol{A}^{\mathrm{ML}}\boldsymbol{u}^{\mathrm{ML}} = \boldsymbol{P}^T\boldsymbol{A}\boldsymbol{P}\boldsymbol{u}^{\mathrm{ML}} = \boldsymbol{P}^T\boldsymbol{b} = \boldsymbol{b}^{\mathrm{ML}}.$$

The squared energy norm (10) of the error can be expressed in this hierarchical splitting as

$$\begin{aligned} E_M(\xi) &= \boldsymbol{e}_M^{\mathrm{ML}}(\xi)^T\boldsymbol{A}^{\mathrm{ML}}(\xi)\boldsymbol{e}_M^{\mathrm{ML}}(\xi) \\ &= \boldsymbol{r}_M^{\mathrm{ML}}(\xi)^T\boldsymbol{A}^{\mathrm{ML}}(\xi)^{-1}\boldsymbol{r}_M^{\mathrm{ML}}(\xi), \end{aligned} \tag{21}$$

where $\boldsymbol{P}\boldsymbol{e}_M^{\mathrm{ML}}(\xi) = \boldsymbol{e}_M(\xi)$ and $\boldsymbol{P}\boldsymbol{r}_M^{\mathrm{ML}}(\xi) = \boldsymbol{r}_M(\xi)$, see Lemma 7.1 in "Appendix". Due to Lemma 4.1 applied to (39) in "Appendix", $\boldsymbol{A}^{\mathrm{ML}}(\xi)^{-1}$ in (21) can be approximated using

$$\begin{aligned} &\frac{1}{1+\gamma}\boldsymbol{v}^T\begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}}(\xi)^{-1} & 0 \\ 0 & \boldsymbol{A}^{\mathrm{ML,F}}(\xi)^{-1} \end{pmatrix}\boldsymbol{v} \\ &\quad\le \boldsymbol{v}^T\boldsymbol{A}^{\mathrm{ML}}(\xi)^{-1}\boldsymbol{v} \\ &\quad\le \frac{1}{1-\gamma}\boldsymbol{v}^T\begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}}(\xi)^{-1} & 0 \\ 0 & \boldsymbol{A}^{\mathrm{ML,F}}(\xi)^{-1} \end{pmatrix}\boldsymbol{v} \end{aligned} \tag{22}$$

for all $\boldsymbol{v} \in \mathbb{R}^N$, where $\gamma \in \langle 0, 1)$ can be easily obtained for all hierarchical FE spaces and does not depend on the discretization parameter and on $\xi \in \Gamma$ provided all functions $a_k(\boldsymbol{x})$ are piecewise constant or mildly varying on D; see the details in "Appendix". Thus we have to solve only two smaller systems of equations. Moreover, the coarse subspace can be further decomposed into coarse and fine subspaces and thus the system with the matrix $\boldsymbol{A}^{\mathrm{ML,C}}(\xi)$ can be further approximated by a two-by-two block diagonal matrix.

The matrix $\boldsymbol{A}^{\mathrm{ML,F}}(\xi)$ can be equivalently substituted by its diagonal. An example of such an equivalence is given in "Appendix". More precisely, let $\boldsymbol{D}^{\mathrm{ML,F}}(\xi)$ be the diagonal of $\boldsymbol{A}^{\mathrm{ML,F}}(\xi)$, then there exist constants $0 < \beta_1 \le 1 \le \beta_2 < \infty$ such that

$$\beta_1\ \boldsymbol{v}^T\boldsymbol{D}^{\mathrm{ML,F}}(\xi)\boldsymbol{v} \le \boldsymbol{v}^T\boldsymbol{A}^{\mathrm{ML,F}}(\xi)\boldsymbol{v} \le \beta_2\ \boldsymbol{v}^T\boldsymbol{D}^{\mathrm{ML,F}}(\xi)\boldsymbol{v}$$

for all $\boldsymbol{v} \in \mathbb{R}^N$. Thus using Lemma 7.1 in Appendix, we finally get the *multilevel* (ML) guaranteed error bounds

$$E_M^{\mathrm{ML},1}(\xi) \le \boldsymbol{e}_M^{\mathrm{ML}}(\xi)^T\boldsymbol{A}^{\mathrm{ML}}(\xi)\boldsymbol{e}_M^{\mathrm{ML}}(\xi) \le E_M^{\mathrm{ML},2}(\xi), \tag{23}$$

where

$$\begin{aligned} E_M^{\mathrm{ML},1}(\xi) =& \frac{1}{(1+\gamma)}\ \boldsymbol{r}_M^{\mathrm{ML,C}}(\xi)^T\boldsymbol{A}^{\mathrm{ML,C}}(\xi)^{-1}\boldsymbol{r}_M^{\mathrm{ML,C}}(\xi) \\ &+ \frac{1}{(1+\gamma)\beta_2}\ \boldsymbol{r}_M^{\mathrm{ML,F}}(\xi)^T\boldsymbol{D}^{\mathrm{ML,F}}(\xi)^{-1}\boldsymbol{r}_M^{\mathrm{ML,F}}(\xi) \end{aligned}$$

and

$$\begin{aligned} E_M^{\mathrm{ML},2}(\xi) =& \frac{1}{(1-\gamma)}\ \boldsymbol{r}_M^{\mathrm{ML,C}}(\xi)^T\boldsymbol{A}^{\mathrm{ML,C}}(\xi)^{-1}\boldsymbol{r}_M^{\mathrm{ML,C}}(\xi) \\ &+ \frac{1}{(1-\gamma)\beta_1}\ \boldsymbol{r}_M^{\mathrm{ML,F}}(\xi)^T\boldsymbol{D}^{\mathrm{ML,F}}(\xi)^{-1}\boldsymbol{r}_M^{\mathrm{ML,F}}(\xi), \end{aligned}$$

and the multilevel central error estimate

$$\begin{aligned} E_M^{\mathrm{ML}}(\xi) =& \boldsymbol{r}_M^{\mathrm{ML,C}}(\xi)^T\boldsymbol{A}^{\mathrm{ML,C}}(\xi)^{-1}\boldsymbol{r}_M^{\mathrm{ML,C}}(\xi) \\ &+ \boldsymbol{r}_M^{\mathrm{ML,F}}(\xi)^T\boldsymbol{D}^{\mathrm{ML,F}}(\xi)^{-1}\boldsymbol{r}_M^{\mathrm{ML,F}}(\xi), \end{aligned} \tag{24}$$

where

$$E_M^{\mathrm{ML},1}(\xi) \le E_M^{\mathrm{ML}}(\xi) \le E_M^{\mathrm{ML},2}(\xi) \quad \text{for all } \xi \in \Gamma.$$

The constants γ, β_1 and β_2 can be quantified or estimated under many practical and theoretical (not only) FE settings. In our numerical experiments in Sect. 6, we will use bilinear FE basis functions with rectangular supports, and thus either $\gamma^2 \le \frac{3}{8} \approx 0.6124^2$ if $a(\boldsymbol{x}, \xi)$ is constant on the coarse elements, or $\gamma^2 \le 0.4109 \approx 0.6410^2$ if $a(\boldsymbol{x}, \xi)$ is mildly varying inside the coarse elements (up to 20%), and $\beta_1 = \frac{1}{4}$ and $\beta_2 = \frac{7}{4}$; see the details in Appendix. For triangular or tetrahedral elements and piecewise linear basis functions [4, 7, 16], $\gamma^2 \in \langle \frac{3}{8}, \frac{3}{4} \rangle$ depending on the shape of elements and iso/anisotropy of the operator. A hierarchy can be also built from different polynomial degrees [16, 36]. Note that a uniform guaranteed upper bound to γ and the bounds to β_1 and β_2 can be obtained from a simple algebra after rectricting the problem on a reference element [5, 16, 36]. The presented two-level method of substituting $\boldsymbol{A}^{-1}$ by the inverse of a block 2×2 matrix is a corner stone of a multilevel algorithm. The idea of recursive splitting of the coarse spaces and the way how to substitute $\boldsymbol{A}^{-1}$ are given in [3, 21, 28, 37]. As a result, the bounds analogous to $\frac{1}{1+\gamma}$ and $\frac{1}{1-\gamma}$ in (22) are 1 and λ, respectively, where

$$\lambda = \frac{1}{2\sqrt{1-\gamma^2}-1}$$

provided $\gamma^2 < \frac{3}{4}$. For the case $\gamma^2 \geq \frac{3}{4}$, the multilevel method can still be used but the bounds are not uniform with respect to the number of levels any more. Note that the computation is faster but the bounds to the true error become less tight with growing number of levels.

5 Algorithms

The purpose of this part is to present the transcription of the methods suggested in the previous sections into a pseudo-code which may help to understand the details of the algorithms. First, we introduce the general GRB algorithm. Then we present the algorithms for two different methods of the a posteriori error estimation: mean-based (MB) estimates and multilevel (ML) estimates, which are described in Sects. 4.1 and 4.2 , respectively. These two methods are described in the form of modifications of the GRB algorithm.

5.1 GRB algorithm

The following algorithm represents a typical greedy algorithm for the construction of a reduced basis, described in Sect. 3. The algorithm is general in the sense that the error estimator is not specified yet.

GRB algorithm:

Input: The training set T_J and the constant $\tau > 0$ which serves as a threshold for the acceptable (squared) norm of the error. The training set T_J is considered to be divided into $L \geq 1$ nonoverlapping sets (levels): $T_J := T^{\text{lev }1} \cup T^{\text{lev }2} \dots \cup T^{\text{lev }L}$.

Output: The $N \times M$ matrix $\boldsymbol{U}_M$, the columns of which form the reduced basis ($\boldsymbol{U}_M$ is constructed column-by-column).

```
1. Set $T := T^{\text{lev }1}$, the first level of $T_J$; set $l := 1$.
     Take $\xi_1 \in T$ which is close to the center of $\Gamma$.
     Solve (5) with $\xi = \xi_1$; denote the solution $\boldsymbol{u}_1$.
     Set $\boldsymbol{U}_1 := \boldsymbol{u}_1$ and $T_1 := \{\xi_1\}$.
     Set $T := T \setminus \{\xi_1\}$.
     Set $m := 1$.
2. For $j = 1, 2, \dots, |T|$:
     Find the solution of (7) with $\xi = \xi_j \in T$; denote the solution by $\boldsymbol{u}_{m,j}$.
     Compute $E_{m,j}$ – an estimate (of the square) of some suitable norm of $\boldsymbol{e}_{m,j}$,
     the error vector, which is defined as $\boldsymbol{e}_{m,j} = \boldsymbol{A}(\xi)^{-1}\boldsymbol{b} - \boldsymbol{U}_m\boldsymbol{u}_{m,j}$,
     where $\boldsymbol{U}_m\boldsymbol{u}_{m,j}$ is a prolongation of $\boldsymbol{u}_{m,j}$.
3. If $\max_{j=1,\dots,|T|} E_{m,j} < \tau$ and $l = L$ then $M := m$ and stop.
     If $\max_{j=1,\dots,|T|} E_{m,j} < \tau$ and $l < L$ then go to step 4.
     Set $\xi_{m+1} := \xi_{j_{\max}}$, where $j_{\max} := \operatorname{argmax}_{j=1,\dots,|T|} E_{m,j}$.
     Solve (5) with $\xi = \xi_{m+1}$, and denote the solution by $\boldsymbol{u}_{m+1}$.
     Set $\boldsymbol{U}_{m+1} := [\boldsymbol{U}_m, \boldsymbol{u}_{m+1}]$ and $T_{m+1} := T_m \cup \{\xi_{m+1}\}$.
     Orthonormalize $\boldsymbol{U}_{m+1}$.
4. Set $T_{\text{remove}} := \{\xi_j \in T;\, E_{m,j} < \tau\}$.
     Set $T := T \setminus T_{\text{remove}}$.
     Enlarge $T$ by adding the next level of values of $\xi$, if applicable (for example, if $|T|$ became less than some chosen threshold and $l < L$), i.e. set $T := T \cup T^{\text{lev }l}$, $l := l + 1$.
     Set $m := m + 1$ and go to step 2.
```

We compare two special cases of the GRB algorithm, namely the GRB-MB algorithm and the GRB-ML algorithm below, which in step 2 of the GRB algorithm use the mean-based squared error estimate E_m^{MB} and the multilevel estimate E_m^{ML}, respectively, as described in the previous sections.

5.2 GRB-MB algorithm

The estimate $E_{m,j}^{\text{MB}} := E_m^{\text{MB}}(\xi_j)$ of (14) of the squared energy norm (10) is used. In order to effectively compute $E_{m,j}^{\text{MB}}$, auxiliary vectors $\boldsymbol{w}_{k,m} \in \mathbb{R}^N$ are successively computed, where $\boldsymbol{w}_{k,m} = (\boldsymbol{A}_0)^{-1}\boldsymbol{A}_k\boldsymbol{u}_m, k = 0, \dots, K, m = 1, \dots, M$. These vectors are used for the computation of $E_{m,j}^{\text{MB}}$ in the following manner. The residual of (5) used in (14) can be expressed as

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}(\xi_j)\boldsymbol{U}_m\boldsymbol{u}_{m,j} = \boldsymbol{b} - \left(\sum_{k=0}^{K}(\xi_j)_k\boldsymbol{A}_k\right)\boldsymbol{U}_m\boldsymbol{u}_{m,j}$$

$$= \boldsymbol{b} - \sum_{k=0}^{K}(\xi_j)_k\left(\boldsymbol{A}_k\boldsymbol{U}_m\right)\boldsymbol{u}_{m,j},$$

and its inverse image with respect to $\boldsymbol{A}_0$ as

$$(\boldsymbol{A}_0)^{-1}\boldsymbol{r} = (\boldsymbol{A}_0)^{-1}\boldsymbol{b} - \sum_{k=0}^{K}(\xi_j)_k\sum_{i=1}^{m}(\boldsymbol{u}_{m,j})_i\boldsymbol{w}_{k,i}. \tag{25}$$

From relationship (25), the formula for the error estimate (14) can be obtained as

$$\begin{aligned} E^{\mathrm{MB}}_{m,j} &= \boldsymbol{r}^T(\boldsymbol{A}_0)^{-1}\boldsymbol{r} = \boldsymbol{r}^T(\boldsymbol{A}_0)^{-1}\boldsymbol{A}_0\,(\boldsymbol{A}_0)^{-1}\boldsymbol{r} \\ &= \left((\boldsymbol{A}_0)^{-1}\boldsymbol{r}\right)^T\boldsymbol{A}_0\left((\boldsymbol{A}_0)^{-1}\boldsymbol{r}\right) \\ &= \left(\boldsymbol{b}^T(\boldsymbol{A}_0)^{-1} - \sum_{k=0}^{K}(\xi_j)_k\sum_{i=1}^{m}(\boldsymbol{u}_{m,j})_i\boldsymbol{w}^T_{k,i}\right) \\ &\quad \boldsymbol{A}_0\left((\boldsymbol{A}_0)^{-1}\boldsymbol{b} - \sum_{k=0}^{K}(\xi_j)_k\sum_{i=1}^{m}(\boldsymbol{u}_{m,j})_i\boldsymbol{w}_{k,i}\right) \\ &= \boldsymbol{b}^T(\boldsymbol{A}_0)^{-1}\boldsymbol{b} - 2\sum_{k=0}^{K}\sum_{i=1}^{m}(\xi_j)_k(\boldsymbol{u}_{m,j})_i\boldsymbol{b}^T\boldsymbol{w}_{k,i} \\ &\quad + \sum_{k=0}^{K}\sum_{r=0}^{K}\sum_{i=1}^{m}\sum_{s=1}^{m}(\xi_j)_r(\xi_j)_k(\boldsymbol{u}_{m,j})_s(\boldsymbol{u}_{m,j})_i\boldsymbol{w}^T_{r,s}\boldsymbol{A}_0\boldsymbol{w}_{k,i}\,, \end{aligned} \tag{26}$$

cf. [17]. The inner products $\boldsymbol{b}^T\boldsymbol{w}_{k,i}$ and $\boldsymbol{w}^T_{r,s}\boldsymbol{A}_0\boldsymbol{w}_{k,i}$, $r,k = 0,\dots,K$, $s,i = 1,\dots,m$, are independent of ξ, they are successively computed, stored and reused.

GRB-MB algorithm (refinement of the GRB algorithm)
Input, Output: As in the GRB algorithm.

1. As in the GRB algorithm, plus:
   Set $\boldsymbol{W}_{0,1} = \boldsymbol{u}_1$ and $(\xi_j)_0 = 1$, $j = 1,\dots,J$ (in order to simplify the notation later).
   Set $\boldsymbol{W}_{k,1} = (\boldsymbol{A}_0)^{-1}\boldsymbol{A}_k\boldsymbol{u}_1$, $k = 1,\dots,K$, and compute and store the inner products
   $\boldsymbol{b}^T\boldsymbol{w}_{k,1}$ and $\boldsymbol{w}^T_{k,1}\boldsymbol{A}_0\boldsymbol{w}_{r,1}$, $r,k = 0,\dots,K$.
2. As in the GRB algorithm. The estimate $E_{m,j} \equiv E^{\mathrm{MB}}_{m,j}$ is computed using (26).
3. As in the GRB algorithm, plus whenever a new column $\boldsymbol{u}_{m+1}$ is added to $\boldsymbol{U}_m$ in the GRB algorithm, compute vectors $\boldsymbol{w}_{k,m+1} = (\boldsymbol{A}_0)^{-1}\boldsymbol{A}_k\boldsymbol{u}_{m+1}$, $k = 0,\dots,K$, and use them for the computation of the inner products $\boldsymbol{b}^T\boldsymbol{w}_{k,m+1}$ and $\boldsymbol{w}^T_{k,m+1}\boldsymbol{A}_0\boldsymbol{w}_{r,s} = \boldsymbol{w}^T_{k,m+1}\boldsymbol{A}_r\boldsymbol{u}_s$, $r,k = 0,\dots,K$, $s = 1,\dots,m+1$. Store these inner products.
4. As in the GRB algorithm.

5.3 GRB-ML algorithm

The estimate $E^{\mathrm{ML}}_{m,j} := E^{\mathrm{ML}}_m(\xi_j)$ of (24) of the squared energy norm (10) is used.

The residual $\boldsymbol{r}^{\mathrm{ML,C}}_M(\xi)$ in (24) represents the residual $\boldsymbol{r}_M(\xi)$ from (10), expressed in the hierarchical basis. It is obtained using $\boldsymbol{P}$ from (19) as $\boldsymbol{r}^{\mathrm{ML,C}}_M(\xi) = \boldsymbol{P}^T\boldsymbol{r}_M(\xi)$.

The matrices $\boldsymbol{A}^{\mathrm{ML,C}}(\xi)$ and $\boldsymbol{D}^{\mathrm{ML,F}}(\xi)$ in (24) come from the hierarchical splitting (20) applied to equation (5), which leads to the transformed matrix

$$\boldsymbol{A}^{\mathrm{ML}}(\xi) = \boldsymbol{P}^T\boldsymbol{A}(\xi)\,\boldsymbol{P} = \begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}}(\xi) & \boldsymbol{A}^{\mathrm{ML,CF}}(\xi) \\ {\boldsymbol{A}^{\mathrm{ML,CF}}}^T(\xi) & \boldsymbol{A}^{\mathrm{ML,F}}(\xi) \end{pmatrix},$$

with $\boldsymbol{D}^{\mathrm{ML,F}}(\xi)$ being the diagonal of $\boldsymbol{A}^{\mathrm{ML,F}}(\xi)$. From the affine property of ξ we have

$$\begin{aligned} \boldsymbol{A}^{\mathrm{ML,C}}(\xi) &= \boldsymbol{A}^{\mathrm{ML,C}}_0 + \sum_{k=1}^{K}(\xi)_k\boldsymbol{A}^{\mathrm{ML,C}}_k, \\ \boldsymbol{D}^{\mathrm{ML,F}}(\xi) &= \boldsymbol{D}^{\mathrm{ML,F}}_0 + \sum_{k=1}^{K}(\xi)_k\boldsymbol{D}^{\mathrm{ML,F}}_k. \end{aligned} \tag{27}$$

Alternatively, all vectors and matrices can be considered with respect to the hierarchical basis during the whole algorithm. Then, no transformation with $\boldsymbol{P}$ is needed.

GRB-ML algorithm (refinement of the GRB algorithm)
Input, Output: As in the GRB algorithm.

1. As in the GRB algorithm, plus:
   Choose the hierarchy and prepare the transformation $\boldsymbol{P}$ from (19).
   Assemble and store the matrices $\boldsymbol{A}^{\mathrm{ML,C}}_k$, $k = 0,\dots K$, from (27).
2. As in the GRB algorithm. The estimate $E_{m,j} \equiv E^{\mathrm{ML}}_{m,j}$ of (24) is computed in three steps:

   2.1 Compute the residual of (5) as $\boldsymbol{r} = \boldsymbol{b} - \sum_{k=0}^{K}(\xi_j)_k\left(\boldsymbol{A}_k\boldsymbol{U}_m\right)\boldsymbol{u}_{m,j}$.
   Transform and split it to coarse and fine residuals as

   $$\boldsymbol{r}^{\mathrm{ML}} = \boldsymbol{P}^T\boldsymbol{r} = \begin{pmatrix} \boldsymbol{r}^{\mathrm{ML,C}} \\ \boldsymbol{r}^{\mathrm{ML,F}} \end{pmatrix}.$$

   2.2 Assemble the matrices $\boldsymbol{A}^{\mathrm{ML,C}}(\xi_j)$ and $\boldsymbol{D}^{\mathrm{ML,F}}(\xi_j)$ using (27).
   Solve the coarse problem $\boldsymbol{A}^{\mathrm{ML,C}}(\xi_j)\,\boldsymbol{e}^{\mathrm{C}} = \boldsymbol{r}^{\mathrm{ML,C}}$.
   Compute $\boldsymbol{e}^{\mathrm{F}} = (\boldsymbol{D}^{\mathrm{ML,F}}(\xi_j))^{-1}\boldsymbol{r}^{\mathrm{ML,F}}$.

   2.3 Compute the estimate as $E^{\mathrm{ML}}_{m,j} = \boldsymbol{r}^T\boldsymbol{P}\begin{pmatrix} \boldsymbol{e}^{\mathrm{C}} \\ \boldsymbol{e}^{\mathrm{F}} \end{pmatrix} = (\boldsymbol{r}^{\mathrm{ML}})^T\begin{pmatrix} \boldsymbol{e}^{\mathrm{C}} \\ \boldsymbol{e}^{\mathrm{F}} \end{pmatrix}$.

3., 4. As in the GRB algorithm.

5.4 Computational and storage cost

Both the GRB-MB and GRB-ML algorithms have the same basic structure of the GRB algorithm, so that we discuss just the differences between the algorithms: additional memory and computational cost. In evaluating the computational cost, we concentrate on the solution of linear systems only, which represents the main part of the cost.

GRB-MB algorithm

Additional memory needed: about $\frac{3}{2}KM + \frac{1}{2}(KM)^2$ floating point numbers for $\boldsymbol{b}^T\boldsymbol{w}_{k,i}$ and $\boldsymbol{w}^T_{r,s}\boldsymbol{A}_0\boldsymbol{w}_{k,i}$, $r,k = 0,\dots,K$, $s,i = 1,\dots,M$.

For each vector $\boldsymbol{u}_m$, $m = 1, \dots, M$, of the reduced basis, the computation of vectors $\boldsymbol{w}_{k,m}$ represents the solution of K systems of the size N with the (sparse) matrix $\boldsymbol{A}_0$.

GRB-ML algorithm

Additional memory needed: about $(K+1) \cdot N_C$ floating point numbers for $\boldsymbol{A}_k^{\mathrm{ML,C}}$, $k = 0, 1, \dots, K$, where N_C is the size of the coarse basis (matrices $\boldsymbol{A}_k^{\mathrm{ML,C}}$ are sparse).

For each vector $\boldsymbol{u}_m$, $m = 1, \dots, M$, of the reduced basis: for each ξ_j in the (actual) training set, the computation of the coarse part of the error $\boldsymbol{e}^{\mathrm{C}}$ represents the solution of the system of the size N_C with the (sparse) matrix $\boldsymbol{A}^{\mathrm{ML,C}}(\xi_j)$.

6 Numerical experiments

In our experiments, we consider $D = (0,1) \times (0,1)$ with a uniform discretization grid and piece-wise bilinear FE functions. We consider two benchmark problems denoted by (P1) and (P2) motivated by [18]. In problem (P1), the coefficient function (2) is defined as

$$\begin{aligned} a_0(\boldsymbol{x}) =& 1, \quad a_k(\boldsymbol{x}) = \frac{1}{2(m_{k,1}^2 + m_{k,2}^2)} \sin(m_{k,1}\pi(\boldsymbol{x})_1) \\ & \sin(m_{k,2}\pi(\boldsymbol{x})_2), \quad k = 1, \dots, K, \end{aligned} \tag{28}$$

where $(m_{k,1}, m_{k,2}) \subset \mathbb{N}^2$ is the k-th pair in the ordered sequence of pairs of integers. The elements in the sequence are ordered according to their Euclidean norm. Thus, the first seven pairs are (1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (1, 3), (2, 3). The data of problem (P2) mimic random porosity of the media. The domain D is uniformly divided into $m \times m$ rectangles, $m = 2, 3, 4, \dots$. For a given m, $K = m^2$, the coefficient function (2) is defined as

$$a_0(\boldsymbol{x}) = 1, \quad a_k(\boldsymbol{x}) = c\, \chi_k, \quad k = 1, \dots, K, \tag{29}$$

where χ_k is the characteristic function associated with the k-th rectangle. We choose $c = 0.5$ in Example 6.4 and $c = 0.95$ in Examples 6.1 and 6.6 . In all examples, we use $\Gamma = \langle -1, 1 \rangle^K$ and a training set generated randomly using the pseudo-random generator function. In all experiments, we present the results for one particular choice of the random training set, but almost identical results are obtained for other choices. The probability density $\rho(\xi)$ is constant in our experiments and equals $\frac{1}{|\Gamma|}$. By an error we mean the deviation of some approximate discretized solution from the exact discretized solution. In our experiments, we compare the true energy norm of the error and its two estimates: the mean based (MB) estimate and the multilevel (ML) estimate. In addition, we compare the memory consumption and, instead of the solution time, which strongly depends on implementation, we compare the numbers of steps of the conjugate gradient (CG) method performed during the GRB-MB or GRB-ML computation. The CG method is used for solving any problem of the size N or the coarse problem. All of them have sparse matrices. We compute the numbers of the CG steps in three different stages: in the MB estimate where the auxiliary vectors $\boldsymbol{w}_{k,i}$, $k = 0, \dots, K$, $i = 1, \dots, M$, are computed, (cf. Sect. 5.2), in the ML estimates where the coarse problems with matrices $\boldsymbol{A}^{\mathrm{ML,C}}(\xi)$ are solved (cf. Sect. 5.3), and in the common part of the GRB algorithm where the new vectors $\boldsymbol{u}^m$ of the reduced basis are computed (cf. Sect. 5.1). Since the coarse systems solved for the ML estimates are smaller than the original problems (5), we use a scaling factor to get a realistic comparison of the total work of the MB and ML methods. The scaling factor corresponds to the coarsening scale which is $\frac{N_C}{N} \approx \frac{1}{4}$ in our examples. All experiments were performed in MATLAB [29] on a personal computer. The threshold for the squared energy norm of the error in all experiments is $\tau = 10^{-6}$. The CG method is finished whenever the relative residual decrease reaches 10^{-8}. The quality of the resulting reduced basis after finishing the GRB algorithm is measured by the averaged squared norm of the exact error

$$E_M^{\mathrm{av}} = \frac{1}{|T_J|} \sum_{\xi_j \in T_J} E_M(\xi_j) \approx \int_\Gamma E_M(\xi) \rho(\xi)\, \mathrm{d}\xi, \tag{30}$$

which we compute for the GRB-MB and GRB-ML methods separately. For these respective methods we also evaluate and compare the *central* estimates of E_M^{av},

$$\begin{aligned} E_M^{\mathrm{av,MB}} =& \frac{1}{|T_J|} \sum_{\xi_j \in T_J} E_M^{\mathrm{MB}}(\xi^j) \\ =& \frac{1}{|T_J|} \sum_{\xi_j \in T_J} \boldsymbol{r}_M(\xi_j)^T \boldsymbol{A}_0^{-1} \boldsymbol{r}_M(\xi_j) \end{aligned} \tag{31}$$

$$\begin{aligned} E_M^{\mathrm{av,\,ML}} =& \frac{1}{|T_J|} \sum_{\xi_j \in T_J} E_M^{\mathrm{ML}}(\xi_j) \\ =& \frac{1}{|T_J|} \sum_{\xi_j \in T_J} \Big(\boldsymbol{r}_M^{\mathrm{ML,C}}(\xi_j)^T \boldsymbol{A}^{\mathrm{ML,C}}(\xi_j)^{-1} \boldsymbol{r}_M^{\mathrm{ML,C}}(\xi_j) \\ & + \boldsymbol{r}_M^{\mathrm{ML,F}}(\xi_j)^T D(\xi_j)^{-1} \boldsymbol{r}_M^{\mathrm{ML,F}}(\xi_j) \Big), \end{aligned} \tag{32}$$

using (18) and (24), respectively. The maximal squared energy norm of the exact error over T_J is denoted by $E_M^{\max}$, and the maximal MB and ML estimates are denoted by $E_M^{\mathrm{max,MB}}$ and $E_M^{\mathrm{max,ML}}$, respectively. In addition to this, we compare the guaranteed averaged lower and upper bounds to the squared exact errors. For the MB estimates, the bounds are obtained from (18) where either the uniform constants $\bar{\alpha}_1^{\mathrm{disc}} \le \bar{\alpha}_2^{\mathrm{disc}}$ are used yielding

$$L_M^{\mathrm{av,MB}} = \frac{1}{\bar{\alpha}_2^{\mathrm{disc}}|T_J|}\sum_{\xi_j \in T_J} \boldsymbol{r}_M(\xi_j)^T \boldsymbol{A}_0^{-1} \boldsymbol{r}_M(\xi_j)$$

$$U_M^{\mathrm{av,MB}} = \frac{1}{\bar{\alpha}_1^{\mathrm{disc}}|T_J|}\sum_{\xi_j \in T_J} \boldsymbol{r}_M(\xi_j)^T \boldsymbol{A}_0^{-1} \boldsymbol{r}_M(\xi_j),$$

or the non-uniform constants $\alpha_1^{\mathrm{disc}}(\xi) \le \alpha_2^{\mathrm{disc}}(\xi)$ are used yielding more accurate bounds

$$l_M^{\mathrm{av,MB}} = \frac{1}{|T_J|}\sum_{\xi_j \in T_J} \frac{1}{\alpha_2^{\mathrm{disc}}(\xi_j)} \boldsymbol{r}_M(\xi_j)^T \boldsymbol{A}_0^{-1} \boldsymbol{r}_M(\xi_j)$$

$$u_M^{\mathrm{av,MB}} = \frac{1}{|T_J|}\sum_{\xi_j \in T_J} \frac{1}{\alpha_1^{\mathrm{disc}}(\xi_j)} \boldsymbol{r}_M(\xi_j)^T \boldsymbol{A}_0^{-1} \boldsymbol{r}_M(\xi_j),$$

however, evaluating $\alpha_1^{\mathrm{disc}}(\xi)$ and $\alpha_2^{\mathrm{disc}}(\xi)$ for every training sample may rather slow down the computation, unless they are precomputed for all $\xi \in T_J$. For the ML estimates, the bounds

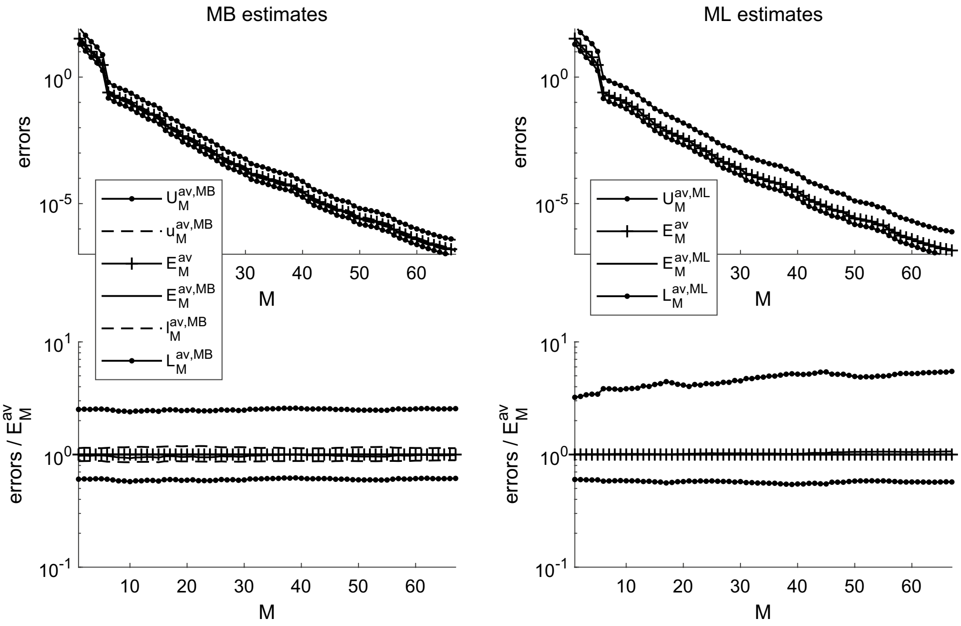

Fig. 1 Averaged exact errors (solid lines with crosses), their estimates (plain solid lines) and guaranteed lower and upper bounds (solid lines with dots for MB and ML; dashed lines for MB) of Example 6.1, (P1) problem for MB (left) and ML (right)

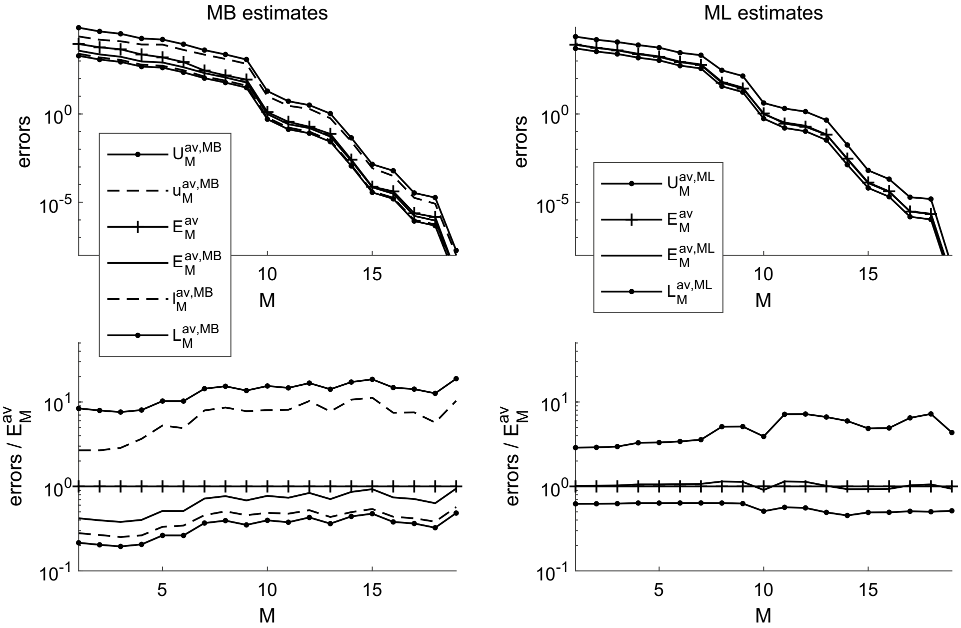

Fig. 2 Averaged exact errors (solid lines with crosses), their estimates (plain solid lines) and guaranteed lower and upper bounds (solid lines with dots for MB and ML; dashed lines for MB) of Example 6.1, (P2) problem for MB (left) and ML (right)

$$L_M^{\mathrm{av,ML}} = \frac{1}{|T_J|} \sum_{\xi_j \in T_J} E_M^{\mathrm{ML},1}(\xi_j),$$

$$U_M^{\mathrm{av,ML}} = \frac{1}{|T_J|} \sum_{\xi_j \in T_J} E_M^{\mathrm{ML},2}(\xi_j)$$

are obtained from (23) using $\beta_1 = \frac{1}{4}$, $\beta_2 = \frac{7}{4}$, and $\gamma = 0.641$ in problem (P1) and $\gamma = \sqrt{\frac{3}{8}}$ in problem (P2).

Example 6.1 In this example, we illustrate the accuracy of the MB and ML a posteriori error estimates in the GRB method. We consider $N = (2^5 - 1)^2$ and the problem (P1) with $K = 6$, and the problem (P2) with $K = 4$ and $c = 0.95$ in (29). We generate 200 random samples and use three levels containing 30, 60 and 110 random samples. In Figs. 1 (P1 problem) and 2 (P2 problem), we can see the graphical plots of the decay of the averaged squared energy norms of the errors and of their estimates. On the horizontal axis, there are sizes M of individual reduced bases designed by the GBR algorithms using the MB or ML a posteriori error estimates. On the vertical axis, there are averaged over $\xi \in T_J$ exact squared energy norms of the errors E_M^{av} (30) (solid line with crosses) and their estimates and guaranteed lower and upper bounds. In both plots, the solid lines without markers denote the *central* error estimates $E_M^{\mathrm{av,MB}}$ and $E_M^{\mathrm{av,ML}}$ defined by (31) and (32), respectively. In the left plot, the solid lines with dots denote the averaged guaranteed lower and upper bounds $L_M^{\mathrm{av,MB}}$ and $U_M^{\mathrm{av,MB}}$. The dashed lines denote the guaranteed lower and upper bounds $l_M^{\mathrm{av,MB}}$ and $u_M^{\mathrm{av,MB}}$. In the right plot, the solid lines with dots denote the guaranteed lower and upper bounds $L_M^{\mathrm{av,ML}}$ and $U_M^{\mathrm{av,ML}}$. In Figs. 1 and 2 in the bottom pairs of plots, we can see the ratios of the computed quantities with respect to the exact error E_M^{av}. We can see that in these examples, the MB or ML estimates yield the same final dimensions of the reduced bases and almost the same decay rates of errors. In both problems, $l_M^{\mathrm{av,MB}}$ and $u_M^{\mathrm{av,MB}}$ provide the best guaranteed two-sided estimates. However, the constants $\alpha_1^{\mathrm{disc}}(\xi)$ and $\alpha_2^{\mathrm{disc}}(\xi)$ are costly to obtain. In (P1) problem, the guaranteed MB bounds obtained for the uniform constants $\bar{\alpha}_1^{\mathrm{disc}} = 0.4875$ and $\bar{\alpha}_2^{\mathrm{disc}}1.5125$ are more accurate than the guaranteed ML bounds. In (P2) problem, the guaranteed ML bounds are more accurate than the guaranteed MB bounds obtained for the uniform constants $\bar{\alpha}_1^{\mathrm{disc}} = 0.05$ and $\bar{\alpha}_2^{\mathrm{disc}}1.95$. The central ML estimates $E_M^{\mathrm{av,ML}}$ are slightly more accurate than the central MB estimates $E_M^{\mathrm{av,MB}}$ in (P2) problem.

Example 6.2 We consider problem (P1), $N = 127^2 = (2^7 - 1)^2$, $K = 1, 3, 6, 12$. We generate 200 random samples divided into three levels containing 30, 60 and 110 random samples. The numerical results are found in Table 1 for the GRB-MB and GRB-ML methods. We present dimensions M of the resulting reduced bases and the averaged exact errors E_M^{av} and the estimates $E_M^{\mathrm{av,MB}}$ and $E_M^{\mathrm{av,ML}}$. For the GRB-MB method we present two lower and two upper averaged guaranteed error bounds $L_M^{\mathrm{av,MB}}$, $l_M^{\mathrm{av,MB}}$, $U_M^{\mathrm{av,MB}}$, and $u_M^{\mathrm{av,MB}}$. For the GRB-ML method, we present

Table 1 Errors and their estimates in Example 6.2 for problem (P1) for $N = (2^7 - 1)^2 = 16129$ and 200 random samples with 3 levels

MB: K	M	E_M^{av}	$L_M^{\mathrm{av,MB}}$	$l_M^{\mathrm{av,MB}}$	$E_M^{\mathrm{av,MB}}$	$u_M^{\mathrm{av,MB}}$	$U_M^{\mathrm{av,MB}}$	E_M^{max}	$E_M^{\mathrm{max,MB}}$
1	6	2.9e−11	2.3e−11	2.7e−11	2.9e−11	3.0e−11	3.9e−11	8.2e−11	8.5e−11
3	45	6.4e−8	4.3e−8	5.6e−8	6.2e−8	7.2e−8	1.1e−7	8.1e−7	9.1e−7
6	107	1.3e−7	7.9e−8	1.1e−7	1.3e−7	1.5e−7	3.3e−7	9.6e−7	8.6e−7
12	170	8.4e−8	4.7e−8	7.5e−8	8.5e−8	9.5e−8	4.3e−7	9.5e−7	9.3e−7
ML: K	M	E_M^{av}	$L_M^{\mathrm{av,ML}}$		$E_M^{\mathrm{av,ML}}$		$U_M^{\mathrm{av,ML}}$	E_M^{max}	$E_M^{\mathrm{max,ML}}$
1	6	2.9e−11	1.7e−11		2.8e−11		9.6e−11	8.2e−11	8.1e−11
3	45	6.4e−8	3.9e−8		6.4e−8		1.9e−7	8.1e−7	8.1e−7
6	109	9.2e−8	5.5e−8		9.2e−8		2.9e−7	7.7e−7	7.8e−7
12	170	8.4e−8	5.0e−8		8.4e−8		2.9e−7	9.6e−7	9.6e−7

Table 2 Memory and CG steps in Example 6.2 for problem (P1) for $N = (2^7 - 1)^2 = 16129$ and 200 random samples with 3 levels

K	MB mem	CG	Comm. mem.	CG	ML mem.	CG	Comm. mem.	CG
1	1.4e2	1.1e3	2.6e5	1.1e3	4.0e4	6.9e3	2.6e5	1.1e3
3	3.2e4	3.7e4	1.0e6	1.4e4	7.9e4	4.5e4	1.0e6	1.4e4
6	5.6e5	1.7e5	2.3e6	3.4e4	1.4e5	1.6e5	2.3e6	3.5e4
12	4.9e6	5.3e5	3.8e6	5.6e4	2.6e5	3.5e5	3.8e6	5.6e4

$L_M^{\mathrm{av,ML}}$ and $U_M^{\mathrm{av,ML}}$. The last two columns contain the maximal achieved error over $\xi \in T_J$ denoted by $E_M^{\max}$, and the maximal estimates obtained either from the MB or ML estimates denoted by $E_M^{\max,\mathrm{MB}}$ and $E_M^{\max,\mathrm{ML}}$, respectively. The MB bounds $l_M^{\mathrm{av,MB}}$ and $u_M^{\mathrm{av,MB}}$ are very accurate. The uniform averaged guaranteed bounds for ML and MB seem comparable. In Table 2, we present how many 8-byte numbers need to be allocated (mem.) and how many (rescaled) CG steps are performed (CG) for the MB and ML methods and in the common part of the GRB algorithm (comm.) We can see that the memory allocated for the auxiliary arrays in the MB method is smaller than that allocated for the set of matrices $\boldsymbol{A}_k$, $k = 0, 1, \ldots, K$, if K is relatively small, otherwise the ML method may need less memory than the MB method.

Example 6.3 We consider problem (P1), $K = 6$, $N = (2^{\exp} - 1)^2$, $\exp = 5, 6, 7, 8$. We generate 200 random samples and use three levels containing 30, 60 and 110 random samples. In Tables 3 and 4 , we present analogous results as in Example 6.2 and Tables 1 and 2 . The reduced bases obtained for the MB and ML error estimates are of almost the same sizes and qualities.

Example 6.4 In this example, we compare enriching the training set during the offline phase with considering the whole training set from the beginning of the GRB method. We solve problem (P2) with $N = 14161$, $K = 9$, and $c = 0.5$

Table 3 Errors and their estimates in Example 6.3 for problem (P1) for $K = 6$ and 200 random samples with 3 levels for $N = (2^{\exp} - 1)^2$

MB: exp	M	E_M^{av}	$L_M^{\mathrm{av,MB}}$	$l_M^{\mathrm{av,MB}}$	$E_M^{\mathrm{av,MB}}$	$u_M^{\mathrm{av,MB}}$	$U_M^{\mathrm{av,MB}}$	$E_M^{\max}$	$E_M^{\max,\mathrm{MB}}$
5	68	1.4e−7	8.7e−8	1.3e−7	1.4e−7	1.6e−7	3.6e−7	8.4e−7	8.6e−7
6	87	1.4e−7	8.5e−8	1.2e−7	1.4e−7	1.6e−7	3.5e−7	8.4e−7	8.6e−7
7	107	1.3e−7	7.9e−8	1.1e−7	1.3e−7	1.5e−7	3.3e−7	9.6e−7	8.6e−7
8	125	1.2e−7	7.7e−8	1.1e−7	1.2e−7	1.4e−7	3.2e−7	8.9e−7	9.0e−7
ML: exp	M	E_M^{av}	$L_M^{\mathrm{av,ML}}$		$E_M^{\mathrm{av,ML}}$		$U_M^{\mathrm{av,ML}}$	$E_M^{\max}$	$E_M^{\max,\mathrm{ML}}$
5	68	1.4e−7	8.1e−8		1.5e−7		7.7e−7	8.4e−7	8.9e−7
6	87	1.5e−7	8.9e−8		1.5e−7		5.8e−7	8.4e−7	8.5e−7
7	109	9.2e−8	5.5e−8		9.2e−8		2.9e−7	7.7e−7	7.8e−7
8	125	1.1e−7	6.5e−8		1.1e−7		3.2e−7	7.4e−7	7.4e−7

Table 4 Memory and CG steps in Example 6.3 for problem (P1) for $K = 6$ and 200 random samples with 3 levels for $N = (2^{\exp} - 1)^2$

Exp	MB mem	CG	Comm. mem.	CG	ML mem.	CG	Comm. mem.	CG
5	2.2e5	2.8e4	9.9e4	5.3e3	7.9e3	2.2e4	9.9e4	5.3e3
6	3.7e5	7.0e4	4.8e5	1.4e4	3.4e4	6.0e4	4.8e5	1.4e4
7	5.6e5	1.7e5	2.3e6	3.4e4	1.4e5	1.6e5	2.3e6	3.5e4
8	7.7e5	4.0e5	1.0e7	8.2e4	5.6e5	4.6e5	1.0e7	8.2e4

Table 5 Errors and their estimates in Example 6.4 for problem (P2), $K = 9$, $N = 14161$, GRB method with Monte Carlo set of 400 samples with 1, 2, or 5 levels of adaptivity

MB: levels	M	E_M^{av}	$L_M^{\mathrm{av,MB}}$	$l_M^{\mathrm{av,MB}}$	$E_M^{\mathrm{av,MB}}$	$u_M^{\mathrm{av,MB}}$	$U_M^{\mathrm{av,MB}}$	$E_M^{\max}$	$E_M^{\max,\mathrm{MB}}$
1	94	1.5e−7	1.5e−8	1.1e−7	1.5e−7	2.6e−7	1.5e−6	1.1e−6	9.4e−7
2	98	7.0e−8	6.8e−9	4.9e−8	6.8e−8	1.2e−7	6.8e−7	8.9e−7	8.0e−7
5	99	5.5e−8	5.4e−9	3.9e−8	5.4e−8	9.8e−8	5.4e−7	7.3e−7	7.5e−7
ML: levels	M	E_M^{av}	$L_M^{\mathrm{av,ML}}$		$E_M^{\mathrm{av,ML}}$		$U_M^{\mathrm{av,ML}}$	$E_M^{\max}$	$E_M^{\max,\mathrm{ML}}$
1	93	1.8e−7	9.6e−8		1.7e−7		8.3e−7	9.1e−7	9.1e−7
2	99	5.8e−8	3.2e−8		5.7e−8		2.5e−7	7.0e−7	6.5e−7
5	99	4.9e−8	2.7e−8		4.8e−8		2.2e−7	5.1e−7	4.9e−7

Table 6 Memory and CG steps in Example 6.4 for problem (P2), $K = 9$, $N = 14161$, GRB method with Monte Carlo set of 400 samples with 1, 2, or 5 levels of adaptivity

Levels	MB mem	CG	Comm. mem.	CG	ML mem.		Comm. mem.	CG
1	7.2e5	2.0e5	1.3e6	3.7e4	1.7e5	1.4e6	1.3e6	3.6e4
2	7.8e5	2.1e5	1.4e6	3.7e4	1.7e5	2.9e5	1.4e6	3.8e4
5	7.9e5	2.1e5	1.4e6	3.8e4	1.7e5	1.5e5	1.4e6	3.8e4

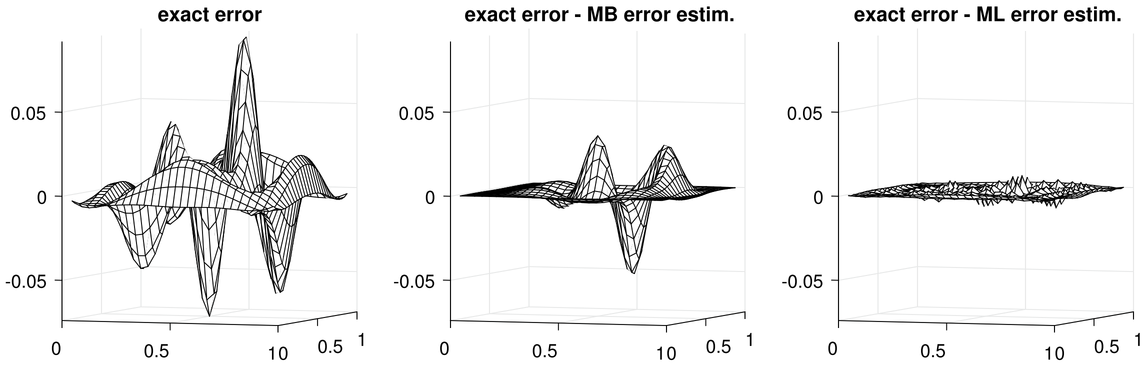

Fig. 3 Exact error (left) of a solution of Example 6.5 for $M = 15$ and $\xi = (-0.9, 0.5, 0.7)$. The difference between the exact error function and the MB estimate (middle), and between the exact error function and the ML estimate (right)

in (29). The training set T_J contains $J = 400$ samples. The samples are either all considered and searched from the beginning of the GRB algorithm, or there are two or five subsets (levels) of samples which are gradually involved: the first subset is considered from the beginning and the next ones are added whenever the rest of the current training set contains less than seven elements. If we use two levels, they contain 100 and 300 samples, respectively; if we use five levels, they contain 30, 60, 60, 100 and 150 samples, respectively. In Tables 5 and 6 , we can see that the gradually enriched training sets significantly reduce the number of CG steps of the GRB-ML algorithm. However, the dimension of the resulting reduced basis slightly grows. The other outputs and the parameters of GRB-MB and GRB-ML remain comparable.

Table 7 Differences between the estimated and true errors in Example 6.6: the means (aver) and the standard deviations (SD) of the differences in the central node of D

RB dim	MB aver	SD	ML aver	SD
10	−2.8e1	6.3e1	−1.1e−1	5.7e−1
20	−2.8e0	8.9e0	−1.1e-1	4.5e-1
30	−1.8e−1	1.1e0	−4.4e−2	2.8e−1
40	−2.2e−2	8.8e−2	−9.1e−3	6.5e−2

Example 6.5 An example of the error function and its MB and ML estimates can be found in Fig. 3 for problem (P1) with $K = 3$ and $N = 29^2$ and with the omitted factor 1/2 at a_1, a_2, a_3. For a reduced basis with $M = 15$ elements obtained by the GRB-MB method for the random sample $\xi = (-0.9, 0.5, 0.7)$, we obtain the residual $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}(\xi)\boldsymbol{U}_{15}\boldsymbol{u}_{15}^{\mathrm{RB}}(\xi)$. We can compare the exact error vector $\boldsymbol{e} = \boldsymbol{A}(\xi)^{-1}\boldsymbol{r}$, the MB approximate error vector $\boldsymbol{e}^{\mathrm{MB}} = \boldsymbol{A}_0^{-1}\boldsymbol{r}$ and the ML approximate error vector

$$\boldsymbol{P}\boldsymbol{e}^{\mathrm{ML}} = \boldsymbol{P}\begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}}(\xi) & 0 \\ 0 & \boldsymbol{D}^{\mathrm{ML,F}}(\xi) \end{pmatrix}^{-1}\boldsymbol{P}^T\boldsymbol{r}$$

reshaped into two-dimensional arrays: $\boldsymbol{e}$ (left), the differences $\boldsymbol{e} - \boldsymbol{e}^{\mathrm{MB}}$ (middle) and $\boldsymbol{e} - \boldsymbol{P}\boldsymbol{e}^{\mathrm{ML}}$ (right). In general, for growing norms of ξ the differences between the two estimates are amplified and the ML estimates are more accurate.

Example 6.6 In this example, we study the localization of the estimates of the error functions obtained by the MB and ML methods. We consider problem (P2) with $N = 59^2$, $K = 9$ and $c = 0.95$ in (29). We examine $M = 10, 20, 30$, and 40, and consider the set T^{test} of 300 random test samples, $T^{\mathrm{test}} \subset \Gamma$, $T^{\mathrm{test}} \neq T_J$. For all $\xi \in T^{\mathrm{test}}$ we compute the solutions $\boldsymbol{u}_M^{\mathrm{RB}}(\xi)$ of (7), and test the exact errors of (5) $\boldsymbol{e}_M(\xi) = \boldsymbol{A}(\xi)^{-1}\boldsymbol{b} - \boldsymbol{U}_M\boldsymbol{u}_M^{\mathrm{RB}}(\xi)$ and the approximate errors obtained either by the MB or by the ML estimates, $\boldsymbol{e}_M^{\mathrm{MB}}(\xi) = \boldsymbol{A}_0^{-1}(\boldsymbol{b} - \boldsymbol{A}(\xi)\boldsymbol{U}_M\boldsymbol{u}_M^{\mathrm{RB}}(\xi))$ and

$$\boldsymbol{P}\boldsymbol{e}_M^{\mathrm{ML}}(\xi) = \boldsymbol{P}\begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}}(\xi) & 0 \\ 0 & \boldsymbol{D}^{\mathrm{ML,F}}(\xi) \end{pmatrix}^{-1} \boldsymbol{P}^T(\boldsymbol{b} - \boldsymbol{A}(\xi)\boldsymbol{U}_M\boldsymbol{u}_M^{\mathrm{RB}}(\xi)), \tag{33}$$

respectively. The averages over $\xi \in T^{\mathrm{test}}$ and the standard deviations of the differences of certain nodal values $(\boldsymbol{e}_M(\xi))_p - (\boldsymbol{e}_M^{\mathrm{MB}}(\xi))_p$ are displayed in Table 7. The same quantities are computed for $(\boldsymbol{e}_M(\xi))_p - (\boldsymbol{P}\boldsymbol{e}_M^{\mathrm{ML}}(\xi))_p$. We chose $p = 1741 = (59^2 + 1)/2$, which is the number of the node in the center of D, $\boldsymbol{v}_p = (0.5, 0.5)$. We can see that for the reduced bases of smaller dimensions M, the approximate error functions obtained by the ML method are better localized (the standard deviation is smaller) than the estimates obtained from the MB method. For richer reduced bases (larger M), the methods provide comparable results.

Several observations can be made from our experiments:

1. Both the MB and ML error estimates provide guaranteed bounds to the exact energy norm of the error of approximate solutions. When using the uniform bounds $\bar{\alpha}_1^{\mathrm{disc}}$ and $\bar{\alpha}_2^{\mathrm{disc}}$ in the MB method (cf. Sect. 4.1), the MB and ML methods yield comparable bounds. When using the individually computed $\alpha_1^{\mathrm{disc}}(\xi)$ and $\alpha_2^{\mathrm{disc}}(\xi)$ for each $\xi \in \Gamma$, the MB bounds are more accurate but the computation may be time consuming. Moreover, when the function $a_0(\boldsymbol{x})$ does not correspond to the mean of $a(\boldsymbol{x}, \xi)$ in (2), or if the variation of $a(\boldsymbol{x}, \xi)$ is large, the MB guaranteed lower and upper error bounds may become useless. The ML estimate provides an error function which better locally corresponds to the exact error function on D, see Examples 6.2 and 6.5 .
2. We encountered the numerical instability of evaluating formula (26), reported in [11, 14, 15, 23], even after reordering the terms trying to avoid the summation of large and small numbers. Therefore, for large K and M we stored and worked with the matrices $\boldsymbol{A}_0^{-1}\boldsymbol{A}_k\boldsymbol{U}_m$, $k = 1, \dots, K$, $m = 1, \dots, M$, instead of the scheme described in Sect. 5.2; cf. [17, Section 3.5.2].
3. In all numerical experiments we observed the exponential decay of the error with respect to the size of the reduced basis, cf. [10, 18].

7 Discussion

In this paper, we deal with the GRB algorithm and suggest a new a posteriori error estimate (called ML), which is based on a multilevel splitting of the FE solution space and can serve as an alternative to the estimate (called here MB) widely used in the GRB methods. While the MB estimates are based on the dual norm of the residual scaled by the coercivity constant, the ML estimate is obtained from a slightly modified energy norm of the error. We discuss and compare these two methods and introduce some numerical experiments. For both estimates, we can have guaranteed lower and upper bounds. In addition, the MB estimate allows for tight guaranteed bounds which, however, depend on constants that may be difficult to evaluate. On the other hand, the ML method yields error functions which are better localized than those obtained from the MB method, especially if the variance of the data is big. The number of operations and the achieved accuracy depend on the problem. For a large number of unknowns and a small number of random variables, the MB methods seems more efficient. If the number of random variables grows, the methods become comparable in terms of the memory and CG steps. There are some modifications which can be applied in the ML method. For example, if the size of the coarse problem is still large, further coarsening of the coarse space can be applied. Gradually enriching the training set instead of considering all test samples from the start of the GRB algorithm also improves the relative efficiency of the ML method. The efficiency of the MB method depends on the affine form of the coefficient function. For the ML method, the coefficient function is not required to be in the affine form; thus in the case of the log-normal distribution of the data the ML estimates can be obtained with almost the same effort as for the affine data provided the coefficient function $a(\boldsymbol{x}, \xi)$ is piece-wise constant or mildly varying on subdomains of D. The ML estimates are directly applicable to any kind of elliptic problems since the associated weak forms generate (semi)norms. Errors of solutions of advection-diffusion-reaction problems can be measured using norms associated to the symmetric part of related non-symmetric bilinear forms [2, 12]. Then the ML error estimates can be used here. In nonlinear elasticity problems, for example, solved by Newton's method, the ML estimates can be used in every step. Moreover, due to mild varying of the system matrices during the steps of Newton's method, bounds to γ can be obtained easily. We believe that in some special problems, e.g. with varying but piecewise constant material data, an appropriate implementation can further improve the efficiency. Our future research will focus on implementing matrix free computation and matrix free forms of the obtained formulas at least for problems with regular grids.

Acknowledgements The authors thank the anonymous referees for helpful comments and suggestions.

Funding This research was supported by the Grant Agency of the Czech Republic under the No. 18-04262S, by the Grant Agency of the Czech Technical University in Prague under the No. SGS19/002/OHK1/1T/11, and partially supported by the Center of Advanced Applied Sciences and the European Regional Development Fund, Project No. CZ.02.1.01/0.0/0.0/16_019/0000778.

Compliance with ethical standards

Conflict of Interest. The authors declare that they have no conflict of interest.

Appendix

We derive some pieces of theory supplementing Sect. 4.2. It is based on the multi-level (or *hierarchical*) splitting of the solution FE space. Some of the first theoretical results were presented in [43]. For the sake of simplicity, the general ideas will be illustrated on the case of bilinear FE elements on the uniform rectangular discretization grid. Let us consider a rectangular domain $D = (0,1) \times (0,1)$ and a model problem: to find $u \in H_0^1(D)$ such that

$$\int_D a(\boldsymbol{x})\nabla u(\boldsymbol{x})\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_D f(\boldsymbol{x})v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad \text{for all } v \in H_0^1(D), \tag{34}$$

$u(\boldsymbol{x}) = 0$ on the boundary of D. Let us consider a uniform rectangular discretization grid in D with $N_1 \times N_2 = N$ inner nodes $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N \in D$, where $N_1 = 2N_{C,1} + 1, N_2 = 2N_{C,2} + 1$. The grid defines $(N_1 + 1)(N_2 + 1)$ (fine) rectangles in D. Let $\psi_k(\boldsymbol{x}), k = 1, \dots, N$, be bilinear FE basis functions which are continuous on D; $\psi_j(\boldsymbol{v}_k) = \delta_{jk}$, where δ_{jk} is the Kronecker delta symbol. Let $h_1 = \frac{1}{N_1+1}$ and $h_2 = \frac{1}{N_2+1}$. Let $a(\boldsymbol{x}) \in L^\infty(D)$ be constant on every element and suppose that there exist $0 < \alpha_1 \le \alpha_2 < \infty$, such that $\alpha_1 \le a(\boldsymbol{x}) \le \alpha_2$ for all $\boldsymbol{x} \in D$. Using the notation

$$(u,v)_a = \int_D a(\boldsymbol{x})\nabla u(\boldsymbol{x})\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{35}$$

let the matrix $\boldsymbol{A}$ be the stiffness matrix of problem (34), $(\boldsymbol{A})_{rs} = (\psi_s, \psi_r)_a$. Denote by V_N the span of $\psi_1, \dots, \psi_N$. Let us consider a uniform coarser discretization of $N_{C,1} \times N_{C,2} = N_C$ *coarse* inner nodes; $h_1^C = 2h_1$ and $h_2^C = 2h_2$ and denote by V_{N_C} the space of the bilinear FE functions associated with this coarse discretization of N_C nodes $\{\boldsymbol{v}_1^C, \dots, \boldsymbol{v}_{N_C}^C\} \subset \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\}$. Let W_{N_C} be such a space that

$$V_N = V_{N_C} \oplus W_{N_C}.$$

The space W_{N_C} is the span of such $N - N_C$ FE basis functions of V_N which are not associated with any coarse node (i.e. have zero values at all coarse nodes). For any function $u \in V_N$ let $\boldsymbol{u}$ be the coefficient vector with respect to the FE basis of V_N, $u(\boldsymbol{x}) = \sum_{k=1}^N \boldsymbol{u}_k\psi_k(\boldsymbol{x})$; and let $\boldsymbol{u}^{\mathrm{ML}}$,

$$\boldsymbol{u}^{\mathrm{ML}} = \begin{pmatrix} \boldsymbol{u}^{\mathrm{ML,C}} \\ \boldsymbol{u}^{\mathrm{ML,F}} \end{pmatrix}, \quad \boldsymbol{P}\boldsymbol{u}^{\mathrm{ML}} = \boldsymbol{u}, \tag{36}$$

be the coefficient vector with respect to the bases of V_{N_C} (first N_C entries) and W_{N_C} (last $N - N_C$ entries). Note that for linear or bilinear FE functions, the coarse coefficients $\boldsymbol{u}^{\mathrm{ML,C}}$ are equal to the nodal values of u in the coarse nodes, $\boldsymbol{u}_k^{\mathrm{ML,C}} = u(\boldsymbol{v}_k^C)$, but $\boldsymbol{u}^{\mathrm{ML,F}}$ are not equal to any nodal values of u in general any more. The individual columns of the transformation matrix $\boldsymbol{P}$ of (36) are thus formed by the values of individual basis functions of V_{N_C} (first N_C columns) and of W_{N_C} (last $N - N_C$ columns) in individual nodes $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$ (each row of the matrix $\boldsymbol{P}$ corresponds to one node). Having obtained $\boldsymbol{u}$ and $\boldsymbol{v}$, as the coefficient vectors of u and v, respectively, we get

$$(u,v)_a = \boldsymbol{v}^T\boldsymbol{A}\boldsymbol{u} = \boldsymbol{v}^{\mathrm{ML}^T}\boldsymbol{A}^{\mathrm{ML}}\boldsymbol{u}^{\mathrm{ML}},$$

where

$$\boldsymbol{P}^T\boldsymbol{A}\boldsymbol{P} = \boldsymbol{A}^{\mathrm{ML}} = \begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}} & \boldsymbol{A}^{\mathrm{ML,CF}} \\ \boldsymbol{A}^{\mathrm{ML,CF}^T} & \boldsymbol{A}^{\mathrm{ML,F}} \end{pmatrix}. \tag{37}$$

Note that all three matrices $\boldsymbol{A}^{\mathrm{ML}}$, $\boldsymbol{A}^{\mathrm{ML,C}}$, and $\boldsymbol{A}^{\mathrm{ML,F}}$ are positive definite. In fact $\boldsymbol{A}^{\mathrm{ML,C}}$ and $\boldsymbol{A}^{\mathrm{ML,F}}$ can be obtained by the restriction of (35) to V_{N_C} and W_{N_C}, respectively. Since V_{N_C} and W_{N_C} are finite-dimensional and the only common function of V_{N_C} and W_{N_C} is the zero function, there exists $\gamma \in \langle 0, 1)$ such that

$$\left(\boldsymbol{v}^{\mathrm{ML,C}^T}\boldsymbol{A}^{\mathrm{ML,CF}}\boldsymbol{v}^{\mathrm{ML,F}}\right)^2 \le \gamma^2\, \boldsymbol{v}^{\mathrm{ML,C}^T}\boldsymbol{A}^{\mathrm{ML,C}}\boldsymbol{v}^{\mathrm{ML,C}}\, \boldsymbol{v}^{\mathrm{ML,F}^T}\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v}^{\mathrm{ML,F}} \tag{38}$$

for all $\boldsymbol{v} \in \mathbb{R}^N$. The inequality (38) is called the Cauchy–Schwarz–Bunyakowski (CBS) strengthened inequality [3, 21]. Usually, we aim to get as small γ fulfilling (38) as possible. For example, for bilinear FE functions and a uniform mesh grid and for the coefficient function $a(\boldsymbol{x}, \xi)$ which is constant on every coarse element (the element corresponding to the coarse space that is composed of 2×2 fine elements) we get $\gamma^2 \le \frac{3}{8} \approx 0.6124^2$, see e.g. [16]. For a triangular grid and for piece-wise linear FEs or for a nonuniform grid and bilinear FEs, we can get $\gamma^2 \leq \frac{3}{4}$, cf. [4, 35]. It is important that a sharp uniform upper bound to the CBS constant γ can be obtained by restricting (38) to a single coarse element independently of the discretization density, see e.g. [3]. Of course, we aim to find its smallest upper bound. Thus, for example, for bilinear FE functions, the upper bound to γ is the square root of the largest singular value of a certain 4×5 matrix [36]. Accordingly, we may derive an upper bound to γ for such a case when $a(\boldsymbol{x})$ varies on coarse elements (but is still constant on fine elements). The theory and the technique of obtaining γ can be found e.g. in [3, 16, 21, 36]. If, for example, the function $a(\boldsymbol{x}, \xi)$ varies up to 20% on the coarse elements but is piece-wise constant on each fine element, then $\gamma^2 \leq 0.4109 \approx 0.6410^2$. If the possible jumps of $a(\boldsymbol{x}, \xi)$ within the coarse elements grow, the constant γ tends to one. However, in our experiments and also in some real life problems we may assume that either $a(\boldsymbol{x}, \xi)$ is constant on every coarse element or the variation of $a(\boldsymbol{x}, \xi)$ with respect to $\boldsymbol{x} \in D$ is mild and does not exceed a certain threshold. In our numerical experiments, we assume that the coefficient function is constant on every fine element and that either it is constant on the coarse elements or it varies in such a manner that there are not larger jumps within any coarse element than 20%.

From (38), we get

$$\begin{aligned}(1-\gamma)\,\boldsymbol{v}^T \begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}} & 0 \\ 0 & \boldsymbol{A}^{\mathrm{ML,F}} \end{pmatrix} \boldsymbol{v} \\ \leq \boldsymbol{v}^T \boldsymbol{A}^{\mathrm{ML}} \boldsymbol{v} \leq (1+\gamma)\,\boldsymbol{v}^T \begin{pmatrix} \boldsymbol{A}^{\mathrm{ML,C}} & 0 \\ 0 & \boldsymbol{A}^{\mathrm{ML,F}} \end{pmatrix} \boldsymbol{v}\end{aligned} \tag{39}$$

for all $\boldsymbol{v} \in \mathbb{R}^N$. Indeed, denoting $\boldsymbol{v}_1 = \boldsymbol{v}^{\mathrm{ML,C}}$ and $\boldsymbol{v}_2 = \boldsymbol{v}^{\mathrm{ML,F}}$, using the symmetry and positive semi-definiteness of the matrices $\boldsymbol{A}^{\mathrm{ML,C}}$ and $\boldsymbol{A}^{\mathrm{ML,F}}$, and using $2xy \leq x^2 + y^2$, we get

$$\begin{aligned}\boldsymbol{v}^T\boldsymbol{A}^{\mathrm{ML}}\boldsymbol{v} =& \boldsymbol{v}_1^T\boldsymbol{A}^{\mathrm{ML,C}}\boldsymbol{v}_1 + \boldsymbol{v}_2^T\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v}_2 + 2\boldsymbol{v}_1^T\boldsymbol{A}^{\mathrm{ML,CF}}\boldsymbol{v}_2 \\ \leq& \boldsymbol{v}_1^T\boldsymbol{A}^{\mathrm{ML,C}}\boldsymbol{v}_1 + \boldsymbol{v}_2^T\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v}_2 + 2\gamma\sqrt{\boldsymbol{v}_1^T\boldsymbol{A}^{\mathrm{ML,C}}\boldsymbol{v}_1\, \boldsymbol{v}_2^T\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v}_2} \\ \leq& \boldsymbol{v}_1^T\boldsymbol{A}^{\mathrm{ML,C}}\boldsymbol{v}_1 + \boldsymbol{v}_2^T\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v}_2 + \gamma(\boldsymbol{v}_1^T\boldsymbol{A}^{\mathrm{ML,C}}\boldsymbol{v}_1 + \boldsymbol{v}_2^T\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v}_2) \\ =& (1+\gamma)(\boldsymbol{v}_1^T\boldsymbol{A}^{\mathrm{ML,C}}\boldsymbol{v}_1 + \boldsymbol{v}_2^T\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v}_2),\end{aligned}$$

which yields the second inequality of (39). The first inequality can be obtained analogously.

Let us consider a single node v_k surrounded by four small rectangles where the values of $a(\boldsymbol{x})$ are $a_1, \dots, a_4$. The matrix stencil of $\boldsymbol{A}$ for such a node (or for the corresponding function $\psi_k(\boldsymbol{x})$) is then an appropriate multiple of

$$\begin{pmatrix} -\frac{a_3}{2} & -\frac{a_3}{4} - \frac{a_4}{4} & -\frac{a_4}{2} \\ -\frac{a_1}{4} - \frac{a_3}{4} & a_1 + a_2 + a_3 + a_4 & -\frac{a_2}{4} - \frac{a_4}{4} \\ -\frac{a_1}{2} & -\frac{a_1}{4} - \frac{a_2}{4} & -\frac{a_2}{2} \end{pmatrix}. \tag{40}$$

In other words, the appropriately ordered entries of (40) are found in the k-th row of $\boldsymbol{A}$. By construction, the matrix $\boldsymbol{A}^{\mathrm{ML,F}}$ arises from $\boldsymbol{A}$ just by omitting the columns and rows with the same numbers as the coarse nodes have. Since every node $\boldsymbol{v}_j$, $j = 1, \dots, N$, has eight neighboring nodes, after omitting from $\boldsymbol{A}$ the columns and rows corresponding to the coarse nodes $\boldsymbol{v}_k^{\mathrm{C}}$, $k = 1, \dots, N_{\mathrm{C}}$, we get $\boldsymbol{A}^{\mathrm{ML,F}}$ the stencil of which can be three-fold:

$$\begin{pmatrix} -\frac{a_3}{2} & -\frac{a_3}{4} - \frac{a_4}{4} & -\frac{a_4}{2} \\ 0 & a_1 + a_2 + a_3 + a_4 & 0 \\ -\frac{a_1}{2} & -\frac{a_1}{4} - \frac{a_2}{4} & -\frac{a_2}{2} \end{pmatrix},$$

$$\begin{pmatrix} -\frac{a_3}{2} & 0 & -\frac{a_4}{2} \\ -\frac{a_1}{4} - \frac{a_3}{4} & a_1 + a_2 + a_3 + a_4 & -\frac{a_2}{4} - \frac{a_4}{4} \\ -\frac{a_1}{2} & 0 & -\frac{a_2}{2} \end{pmatrix}$$

or

$$\begin{pmatrix} 0 & -\frac{a_3}{4} - \frac{a_4}{4} & 0 \\ -\frac{a_1}{4} - \frac{a_3}{4} & a_1 + a_2 + a_3 + a_4 & -\frac{a_2}{4} - \frac{a_4}{4} \\ 0 & -\frac{a_1}{4} - \frac{a_2}{4} & 0 \end{pmatrix}.$$

Note that the sum of the off-diagonal elements is either $-\frac{3}{4}(a_1 + a_2 + a_3 + a_4)$ or $-\frac{1}{2}(a_1 + a_2 + a_3 + a_4)$. Denote by $\boldsymbol{D}^{\mathrm{ML,F}}$ the diagonal part of $\boldsymbol{A}^{\mathrm{ML,F}}$. Then, from the Gerschgorin theorem [42], the spectrum of $(\boldsymbol{D}^{\mathrm{ML,F}})^{-1}\boldsymbol{A}^{\mathrm{ML,F}}$ lies in the interval $\langle \frac{1}{4}, \frac{7}{4} \rangle$. In other words,

$$\begin{aligned}\frac{1}{4}\,\boldsymbol{v}^T\boldsymbol{D}^{\mathrm{ML,F}}\boldsymbol{v} \leq& \boldsymbol{v}^T\boldsymbol{A}^{\mathrm{ML,F}}\boldsymbol{v} \\ \leq& \frac{7}{4}\,\boldsymbol{v}^T\boldsymbol{D}^{\mathrm{ML,F}}\boldsymbol{v} \quad \text{for all } \boldsymbol{v} \in \mathbb{R}^{N-N_{\mathrm{C}}}.\end{aligned} \tag{41}$$

For non-uniform grids or for different types of FE functions, (the uniform estimates of) the constants in (41) can be obtained in a similar manner. Let us summarize the above considerations in the following lemma.

Lemma 7.1 *Let the matrix* $\boldsymbol{A}$ *be the stiffness matrix obtained for the operator* (35) *and for the bilinear FE functions. Let us consider a two-level splitting with properties* (36) *and* (37)*. Let the strengthened CBS inequality* (38) *with the constant* $\gamma \in \langle 0, 1)$ *hold. Denote some error vector by* $\boldsymbol{e}$ *and the residual vector by* $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{e}$*; and let* $\boldsymbol{A}^{\mathrm{ML}}\boldsymbol{e}^{\mathrm{ML}} = \boldsymbol{r}^{\mathrm{ML}}$*, where* $\boldsymbol{r}^{\mathrm{ML}} = \boldsymbol{P}^T\boldsymbol{r}$ *and* $\boldsymbol{e} = \boldsymbol{P}\boldsymbol{e}^{\mathrm{ML}}$*. Then*

$$\boldsymbol{r}^T\boldsymbol{A}^{-1}\boldsymbol{r} = \boldsymbol{e}^T\boldsymbol{A}\boldsymbol{e} = \boldsymbol{e}^{\mathrm{ML}^T}\boldsymbol{A}^{\mathrm{ML}}\boldsymbol{e}^{\mathrm{ML}} = \boldsymbol{r}^{\mathrm{ML}^T}\boldsymbol{A}^{\mathrm{ML}^{-1}}\boldsymbol{r}^{\mathrm{ML}}, \tag{42}$$

and for all $\boldsymbol{v}^{\mathrm{ML}} \in \mathbb{R}^N$ *there hold the inequalities*

$$\begin{aligned}\boldsymbol{v}^{\mathrm{ML}^T}\boldsymbol{A}^{\mathrm{ML}^{-1}}\boldsymbol{v}^{\mathrm{ML}} \leq& \frac{1}{1-\gamma}\Big(\boldsymbol{v}^{\mathrm{ML,C}^T}\boldsymbol{A}^{\mathrm{ML,C}^{-1}}\boldsymbol{v}^{\mathrm{ML,C}} \\ &+4\,\boldsymbol{v}^{\mathrm{ML,F}^T}\boldsymbol{D}^{\mathrm{ML,F}^{-1}}\boldsymbol{v}^{\mathrm{ML,F}}\Big)\end{aligned} \tag{43}$$

and

$$\frac{1}{1+\gamma}\left(\boldsymbol{v}^{\mathrm{ML,C}^T}\boldsymbol{A}^{\mathrm{ML,C}^{-1}}\boldsymbol{v}^{\mathrm{ML,C}} + \frac{4}{7}\boldsymbol{v}^{\mathrm{ML,F}^T}\boldsymbol{D}^{\mathrm{ML,F}^{-1}}\boldsymbol{v}^{\mathrm{ML,F}}\right) \le \boldsymbol{v}^{\mathrm{ML}^T}\boldsymbol{A}^{\mathrm{ML}^{-1}}\boldsymbol{v}^{\mathrm{ML}}. \tag{44}$$

Proof Equalities (42) follow directly from the definitions of the objects involved. Inequalities (43) and (44) follow from Lemma 4.1 applied to (39) and then to (41). □

Analogously to the introduced technique, more than two levels of hierarchy can be considered. Moreover, the coarse elements can be composed of 3 × 3 fine elements (instead of the 2 × 2 fine elements) yielding the coarsening ratio $\frac{N}{N_C} \approx 9$ (instead of $\frac{N}{N_C} \approx 4$).

References

1. Ainsworth M, Oden JT (2011) A posteriori error estimation in finite element analysis. Wiley, Hoboken
2. Araya R, Poza AH, Stephan EP (2005) A hierarchical a posteriori error estimate for an advection–diffusion–reaction problem. Math Models Methods Appl Sci 15(7):1119–1139
3. Axelsson O (1996) Iterative solution methods. Cambridge University Press, Cambridge
4. Axelsson O, Blaheta R (2004) Two simple derivations of universal bounds for the C.B.S. inequality constant. Appl Math 49(1):57–72
5. Axelsson O, Padiy A (1999) On the additive version of the algebraic multilevel iteration method for anisotropic elliptic problems. SIAM J Sci Comput 20:1807–1830
6. Binev P, Cohen A, Dahmen W, DeVore R, Petrova G, Wojtaszczyk P (2011) Convergence rates for greedy algorithms in reduced basis methods. SIAM J Math Anal 43(3):1457–1472
7. Blaheta R, Margenov S, Neytcheva M (2005) Robust optimal preconditioners for non-conforming finite element systems. Numer Linear Algebra Appl 12:495–514
8. Boyaval S (2009) Mathematical modelling and numerical simulation in materials science. Université Paris-Est, theses
9. Brenner SC, Scott LR (2002) The mathematical theory of finite element methods, 2nd edn. Springer, New York
10. Buffa A, Maday Y, Patera AT, Prud'homme Ch, Turinici G (2012) A priori convergence of the greedy algorithm for the parametrized reduced basis method. ESAIM Math Model Numer Anal 46:595–603
11. Buhr A, Engwer C, Ohlberger M, Rave S (2014) A numerically stable a posteriori error estimator for reduced basis approximations of elliptic equations. In: 11th World congress on computational mechanics, WCCM 2014, 5th European conference on computational mechanics, ECCM 2014 and 6th European conference on computational fluid dynamics, ECFD 2014. pp 4094–4102
12. Canuto C (2006) Enhanced coercivity for pure advection and advection–diffusion problems. J Sci Comput 28:223–244
13. Canuto C, Tonn T, Urban K (2009) A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear PDEs. SIAM J Numer Anal 47(3):2001–2022
14. Casenave F (2012) Accurate a posteriori error evaluation in the reduced basis method. C R Acad Sci Paris Ser I 350:539–542
15. Casenave F, Ern A, Lelièvre T (2014) Accurate and online-efficient evaluation of the a posteriori error bound in the reduced basis method. ESAIM Math Model Numer Anal Modél Math Anal Numér 48(1):207–229
16. Crowder AJ, Powell CE (2018) CBS constants and their role in error estimation for stochastic Galerkin finite element methods. J Sci Comput 77:1030–1054
17. Chen P, Quarteroni A, Rozza G (2014) Comparison between reduced basis and stochastic collocation methods for elliptic problems. J Sci Comput 59(1):187–216
18. Chen P, Quarteroni A, Rozza G (2017) Reduced basis mehod for uncertainty quantification. SIAM/ASA J Uncertain Quantif 5(1):813–869
19. Chen Y, Jiang J, Narayan A (2019) A robust error estimator and a residual-free error indicator for reduced basis methods. Comput Math Appl 77(7):1963–1979
20. Dahmen W, Plesken Ch, Welper G (2014) Double greedy algorithms: reduced basis methods for transport dominated problems. ESAIM Math Model Numer Anal 48(3):623–663
21. Eijkhout V, Vassilevski P (1991) The role of the strengthened C.B.S.-inequality in multi-level methods. SIAM Rev 33:405–419
22. Giraldi L, Litvinenko A, Liu D, Matthies HG, Nouy A (2014) To be or not to be intrusive? The solution of parametric and stochastic equations-the "plain vanilla" Galerkin case. SIAM J Sci Comput 36(6):A2720–A2744
23. Hain S, Ohlberger M, Radic M, Urban K (2019) A hierarchical a posteriori error estimator for the reduced basis method. Adv Comput Math 45:2191–2214
24. Hesthaven JS, Rozza G, Stamm B (2016) Certified reduced basis methods for parametrized partial differential equations. Springer briefs in mathematics. Springer, Berlin
25. Hesthaven JS, Stamm B, Shang S (2014) Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. ESAIM Math Model Numer Anal Modél Math Anal Numér Tome 48(1):259–283
26. 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 345(8):473–478
27. 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(29–32):1963–1975
28. Kraus J, Margenov S (2009) Robust algebraic multilevel methods and algorithms, radon series on computational and applied mathematics 5. RICAM, Walter de Gruyter, Germany
29. MATLAB software. https://de.mathworks.com/products/matlab.html
30. Nobile F, Tempone R, Webster CG (2008) A sparse grid stochastic collocation method for partial differential equations with random input data. SIAM J Numer Anal 46(5):2309–2345
31. Ohlberger M, Rave S (2016) Reduced basis methods: success, limitations and future challenges. In: Handlovičová A, Ševčovič D (eds) Proceedings of Algoritmy 2016: 20-th conference on scientific computing, Vysoké Tatry. Bratislava, pp 1–12. ArXiv:1511.02021v2 [math.NA] 8Jan2016
32. Papež J, Liesen J, Strakoš Z (2014) Distribution of the discretization and algebraic error in numerical solution of partial differential equations. Linear Algebra Appl 449:89–114
33. Papež J, Strakoš Z, Vohralík M (2018) Estimating and localizing the algebraic and total numerical errors using flux reconstructions. Numer Math 138:681–721

34. Powell CE, Silvester D, Simoncini V (2017) An efficient reduced basis solver for stochastic Galerkin equations. SIAM J Sci Comput 39(1):A141–A163
35. Pultarová I (2005) The strengthened CBS inequality constant for second order elliptic partial differential operator and for hierarchical bilinear finite element functions. Appl Math 50(3):323–329
36. Pultarová I (2009) Preconditioning and a posteriori error estimates using h- and p-hierarchical finite elements with rectangular supports. Numer Linear Algebra Appl 16(5):415–430
37. Pultarová I (2017) Block and multilevel preconditioning for stochastic Galerkin problems with lognormally distributed parameters and tensor product polynomials. Int J Uncertain Quantif 7:441–462
38. Quarteroni A, Rozza G, Manzoni A (2011) Certified reduced basis approximation for parametrized partial differential equations and applications. J Math Ind 1(3):1–49
39. Robert ChP (2016) The metropolis-hastings algorithm. Wiley StatsRef: statistics reference online. Wiley, Hoboken
40. Rozza G, Huynh D B P, Patera A T (2008) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Arch Comput Methods Eng 15:229
41. Ullmann S, Rotkvic M, Lang J (2016) POD-Galerkin reduced-order modeling with adaptive finite element snapshots. J Comput Phys 325:244–258
42. Varga RS (2000) Matrix iterative analysis. Springer series in computational mathematics 27. Springer, Berlin and Heidelberg
43. Yserentant H (1986) On the multilevel splitting of finite element spaces. Numer Math 49(4):379–412

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