

A robust error estimator and a residual-free error indicator for reduced basis methods

Yanlai Chen ^{a,*¹}, Jiahua Jiang ^{a,1}, Akil Narayan ^{b,2}

^a Department of Mathematics, University of Massachusetts Dartmouth, 285 Old Westport Road, North Dartmouth, MA 02747, USA

^b Scientific Computing and Imaging (SCI) Institute and Department of Mathematics, University of Utah, 72 S Campus Drive, Salt Lake City, UT 84112, USA



ARTICLE INFO

Article history:

Received 30 May 2018

Received in revised form 14 November 2018

Accepted 20 November 2018

Available online 10 December 2018

Keywords:

Reduced basis method

Greedy algorithms

Residual-based error estimation

Error stagnation

ABSTRACT

The Reduced Basis Method (RBM) is a rigorous model reduction approach for solving parameterized partial differential equations. It identifies a low-dimensional subspace for approximation of the parametric solution manifold that is embedded in high-dimensional space. A reduced order model is subsequently constructed in this subspace. RBM relies on residual-based error indicators or *a posteriori* error bounds to guide construction of the reduced solution subspace, to serve as a stopping criteria, and to certify the resulting surrogate solutions. Unfortunately, it is well-known that the standard algorithm for residual norm computation suffers from premature stagnation at the level of the square root of machine precision.

In this paper, we develop two alternatives to the standard offline phase of reduced basis algorithms. First, we design a robust strategy for computation of residual error indicators that allows RBM algorithms to enrich the solution subspace with accuracy beyond root machine precision. Secondly, we propose a new error indicator based on the Lebesgue function in interpolation theory. This error indicator does not require computation of residual norms, and instead only requires the ability to compute the RBM solution. This residual-free indicator is rigorous in that it bounds the error committed by the RBM approximation, but up to an uncomputable multiplicative constant. Because of this, the residual-free indicator is effective in choosing snapshots during the offline RBM phase, but cannot currently be used to certify error that the approximation commits. However, it circumvents the need for *a posteriori* analysis of numerical methods, and therefore can be effective on problems where such a rigorous estimate is hard to derive.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The fundamental reason that many model reduction approaches for parametric partial differential equations (PDE) are successful is that, for many PDE of interest, the solution manifold induced by the parametric variation has small and rapidly-decaying Kolmogorov n -width [1]. Among the reduction strategies that utilize this fact is the greedy approach to the Reduced Basis Method (RBM). It identifies a small set of representative points in parameter space, and obtains solution to the PDE at these points. The construction of this point set proceeds via a greedy algorithm that relies on an *a posteriori* error estimate

* Corresponding author.

E-mail addresses: yanlai.chen@umassd.edu (Y. Chen), jjiang@umassd.edu (J. Jiang), akil@sci.utah.edu (A. Narayan).

¹ Partially supported by National Science Foundation, USA grant DMS-1719698.

² Partially supported by NSF, USA DMS-1720416 and AFOSR, USA FA9550-15-1-0467.

for guidance. This solution ensemble on this small set is typically assembled from iterated queries to a potentially expensive existing solver.

The topic of this paper is the greedy approach for reduced basis methods. In such cases, there are *offline* and *online* stages for RBM algorithms [2–6], see also the recent monologue [7–9]. During the offline stage, significant computational effort is invested so that the online stage, when reduced order solutions for arbitrary parameter values are computed, can be efficient. During the offline stage, the parameter dependence is inspected and the greedy algorithm which the RBM methods mainly rely on judiciously selects a small number of parameter values on which the full-order, expensive PDE solver is employed to obtain so-called solution *snapshots*. During the online stage, a surrogate solution is efficiently computed for any new parameter value as a linear combination of these stored snapshots. The coefficients of this linear combination are computed via a reduced-order formulation of the PDE. This reduced solve can usually be completed with orders of magnitude less effort than a full PDE solve, and thus RBM achieves significant speedup when both the offline phase is not too expensive and when multiple online queries are utilized.

A critical component in RBM algorithms is the *a posteriori* error estimate, which dictates the adaptive sampling criterion in the offline greedy algorithm. This error estimate is the main concern of the current paper. The offline phase of the RBM algorithm finds a parameter value that maximizes the numerically-computed error estimate. Therefore, the accuracy (or the lack thereof) in this calculation dictates the accuracy of the RBM solution. Due to standard computational implementations of the RBM error estimate for elliptic PDE, the accuracy of this estimate stagnates around the level of the square root of the machine accuracy. Therefore, more accurate schemes for calculating the error estimate are necessary if one demands higher accuracy or when the query is close to the part of the parameter domain where resonances occur (and the stability constant approaches zero). To our knowledge, there are two previous attempts to resolve this issue [10,11] and [12]. The method in [10,11] employs an additional sampling of the parameter domain, potentially randomly, to generate a linear system to solve online for the stable calculation of the *a posteriori* error estimate. This is improved in [11] by the empirical interpolation method. These approaches increase the Offline and Online cost, and may suffer from ill-conditioning depending on the additional sampling. On the other hand, [12,13] presents a strategy that amends a direct computation of the *a posteriori* error estimate by rewriting it in a new form; the authors there show that their approach can circumvent stagnation errors due to floating-point arithmetic. Thus, the loss of accuracy of half of the digits caused by taking the square root is not new. But with this understanding, we devise a new approach for computing an *a posteriori* error estimate, and demonstrate that it can be used to circumvent loss of significance from floating-point arithmetic. Our method performs similarly to [12], but in cases where a matrix of residual vectors is rank-deficient, our approach is more efficient. We focus on the standard error criterion for selecting snapshots in this paper, that is, error in the solution, but note that goal-oriented strategies exist [14–16].

The main contributions of this paper are twofold, in the theoretical and algorithmic design of robust residual-based, and residual-free error estimates for the offline RBM phase. Our first contribution is to the standard, residual-based method. We design and test a novel computational strategy for residual-based RBM error estimators that is capable of delaying error stagnation until much closer to machine precision. This new strategy computes the residual-based error norms in different ways compared to standard RBM algorithms, but are just as efficient as those algorithms.

Our second contribution to the offline RBM phase is more general and falls into a residual-free category. The efficiency of the error estimate calculation directly determines that of RBM. When the parameter is high-dimensional then the requisite size of the training set is very large, and it is computationally infeasible to repeatedly maximize a residual-based error estimate over the training set. The situation is exacerbated when a standard residual-based RBM error estimate cannot be computed, such as for hyperbolic problems, or in convection-dominated convection–diffusion equations. To the best of our knowledge, computational stratagems in the RBM framework to tackle this problem are underdeveloped. We propose and test an error indication strategy that forgoes the residual norm calculation entirely and requires only the RBM coefficients. This new procedure is rigorous and applicable to any parameterized problem without requiring any *a posteriori* error analysis. However, the procedure cannot certify error due to the presence of a scaling constant that we have not been able to compute. Nevertheless, this new error estimate performs comparably to standard RBM algorithms for the examples that we have tried.

All of the numerical examples in this manuscript investigate parametric PDE's that are relatively standard situations when RBM algorithms are known to perform well. We demonstrate for these cases that our strategies work well, and there is no methodological restriction that prevents our strategies from being applied in more general, difficult cases. However, we leave investigations of our approaches for computationally large-scale and more mathematically challenging parametric PDE's for future work.

The remainder of this paper is organized as follows. In Section 2, we provide a brief overview of the RBM framework and the typical setting it is successful for. Close attention is paid to the error estimate calculation, the focus of this paper. In Section 3, we detail our two approaches to tackle the afore-mentioned two challenges. Section 4 is devoted to numerical results corroborating the efficiency and accuracy gain of the proposed approaches. Finally, some concluding remarks are given in Section 5.

2. Reduced basis method: a brief overview

In this section, we present a brief overview of the Reduced Basis Method (RBM), in particular the error estimate and its implementation in the classical form. All of this is standard in the RBM literature. Therefore the reader familiar with RBM may skip this section, referring to Table 1 for our notation.

Table 1

Notation used throughout this article.

μ	Parameter in $\mathcal{D} \subseteq \mathbb{R}^p$
$u(\mu)$	Function-valued solution of a parameterized PDE
\mathcal{N}	Degrees of freedom (DoF) in PDE “truth” solver
$X^{\mathcal{N}}$	Truth solver solution space, having dimension \mathcal{N}
$u^{\mathcal{N}}(\mu)$	Truth solution (finite-dimensional)
N	Number of reduced basis snapshots, $N \ll \mathcal{N}$
μ^j	“Snapshot” parameter values, $j = 1, \dots, N$
S^N	Sample set $S^N = \{\mu^1, \dots, \mu^N\}$
$X_N^{\mathcal{N}}$	Span of $u^{\mathcal{N}}(\mu^k)$ for $k = 1, \dots, N$
$u_N^{\mathcal{N}}(\mu)$	Reduced basis solution, $u_N^{\mathcal{N}} \in X_N^{\mathcal{N}}$
$e_N(\mu)$	Reduced basis solution error, equals $u^{\mathcal{N}}(\mu) - u_N^{\mathcal{N}}(\mu)$
$\mathcal{E}_{\text{train}}$	Parameter training set, a finite subset of \mathcal{D}
$\Delta_N(\mu)$	Error estimate (upper bound) for $\ e_N(\mu)\ $
ϵ_{tol}	Error estimate stopping tolerance in greedy sweep

2.1. Problem setting

Let $\mathcal{D} \subset \mathbb{R}^p$ be the domain for a p -dimensional parameter, and let $\Omega \subset \mathbb{R}^d$ (for $d = 2$ or 3) be a bounded physical domain. Let X be a Hilbert space of functions on Ω . Given $\mu \in \mathcal{D}$, the goal is to compute $u(\mu) \in X$ satisfying

$$a(u(\mu), v; \mu) = f(v; \mu), \quad v \in X, \quad (2.1)$$

which corresponds to a parametric partial differential equation (pPDE) written in weak form; $a(\cdot, \cdot; \mu)$ is a bilinear form and f may encode forcing terms and/or boundary conditions. We assume $u \in X$ and that $H_0^1(\Omega) \subset X(\Omega) \subset H^1(\Omega)$, where

$$H^1(\Omega) = \left\{ w \in L^2(\Omega) \mid \int_{\Omega} \|\nabla w\|^2 \, dx < \infty \right\},$$

$$H_0^1(\Omega) = \left\{ w \in L^2(\Omega) \mid \int_{\Omega} \|\nabla w\|^2 \, dx < \infty \text{ and } w|_{\partial\Omega} = 0 \right\}.$$

We denote by $(\cdot, \cdot)_X$ the inner product associated with the space X , whose induced norm $\|\cdot\|_X = \sqrt{(\cdot, \cdot)_X}$ is equivalent to the usual $H^1(\Omega)$ norm. For μ -uniform well-posedness of (2.1), we assume that $a(\cdot, \cdot; \mu) : X \times X \rightarrow \mathbb{R}$ is continuous and uniformly coercive over X for all μ in \mathcal{D} , and that $f(\cdot)$ is a linear continuous functional over X for all μ . That is,

$$\gamma(\mu) := \sup_{w \in X} \sup_{v \in X} \frac{a(w, v; \mu)}{\|w\|_X \|v\|_X} < \infty, \quad \forall \mu \in \mathcal{D}, \quad (2.2a)$$

$$\alpha(\mu) := \inf_{w \in X} \frac{a(w, w; \mu)}{\|w\|_X^2} \geq \alpha_0 > 0, \quad \forall \mu \in \mathcal{D}, \quad (2.2b)$$

$$\sup_{u \in X} |f(u; \mu)| < \infty, \quad \forall \mu \in \mathcal{D}. \quad (2.2c)$$

As is common in the RBM literature [17], we assume that $a(\cdot, \cdot; \mu)$ is “affine” with respect to the parameter μ : i.e., there exist μ -dependent coefficient functions $\Theta_a^q : \mathcal{D} \rightarrow \mathbb{R}$ for $q = 1, \dots, Q_a$, and $\Theta_f^q : \mathcal{D} \rightarrow \mathbb{R}$ for $q = 1, \dots, Q_f$, and corresponding continuous μ -independent bilinear forms $a^q(\cdot, \cdot) : X \times X \rightarrow \mathbb{R}$ and linear forms $f^q(\cdot) : X \rightarrow \mathbb{R}$, respectively, such that

$$a(w, v; \mu) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(w, v), \quad f(w; \mu) = \sum_{q=1}^{Q_f} \Theta_f^q(\mu) f^q(w). \quad (2.3)$$

There are strategies for situations when the affine assumption is not satisfied, e.g., empirical interpolation [18]. These strategies generally replace a non-affine operator by an affine operator with sufficiently large Q_a and Q_f so that the solution of the affine problem well approximates the solution to the non-affine problem. Note, however, that such an approach can require large Q_a and/or Q_f and that RBM strategies can suffer substantially in computational efficiency in this case.

In order to compute solutions to (2.1) suppose that for μ fixed, a spatial discretization (e.g., a finite element solver) can be employed with $\mathcal{N} \gg 1$ degrees of freedom that computes an approximate solution $u^{\mathcal{N}}(\mu)$ to within an acceptable accuracy for every $\mu \in \mathcal{D}$. This approximate solution $u^{\mathcal{N}}$ is called the “truth” solution, and the associated spatial discretization and solver is called the “truth” solver.

The truth solution $u^{\mathcal{N}}$ is sought in an \mathcal{N} -dimensional subspace $X^{\mathcal{N}}$ (i.e., $\dim(X^{\mathcal{N}}) = \mathcal{N}$) that is a discretization of X , and (2.1) is discretized as

$$\begin{cases} \text{For } \mu \in \mathcal{D}, \text{ find the “truth” approximation } u^{\mathcal{N}}(\mu) \in X^{\mathcal{N}} \text{ such that} \\ a(u^{\mathcal{N}}, v; \mu) = f(v; \mu) \quad \forall v \in X^{\mathcal{N}}. \end{cases} \quad (2.4)$$

The other relevant quantities are defined according to the discretization. For example, the coercivity constant (2.2b) is approximated by its numerical counterpart $\alpha^{\mathcal{N}}(\mu) = \inf_{w \in X^{\mathcal{N}}} \frac{a(w, w; \mu)}{\|w\|_X^2}$, $\forall \mu \in \mathcal{D}$.

A naïve approach to computing solutions to (2.1) for many μ would be to query the truth solver many times, which is expensive under the $\mathcal{N} \gg 1$ assumption. RBM methods attempt to provide numerical solutions of (2.1) with accuracy comparable to $u^{\mathcal{N}}$, but with orders-of-magnitude less computational cost than the truth solver. The essential idea is to project the collection of discrete solutions $u^{\mathcal{N}}(\mu)$ for $\mu \in \mathcal{D}$ onto a low-dimensional representation, and then to efficiently compute this projected representation.

2.2. RBM framework

The reduced basis method is a reliable model reduction strategy that replaces a relatively expensive truth solver (2.4) with a less expensive surrogate. The best possible accuracy of the surrogate is governed by a theoretical quantity, the Kolmogorov N -width of the solution set $U^{\mathcal{N}}$, defined as

$$U^{\mathcal{N}} := \{u(\mu) \mid \mu \in \mathcal{D}\} \subset X^{\mathcal{N}}. \quad (2.5)$$

When the N -width of $U^{\mathcal{N}}$ decays quickly with respect to N , an RBM strategy is effective. Practitioners in advance identify a large but finite training set discretizing the parameter domain $\mathcal{E}_{\text{train}} \subset \mathcal{D}$, and a maximum dimension N_{\max} (usually $\ll \mathcal{N}$). An RBM algorithm then approximates the solution space by an N -dimensional subspace of $X^{\mathcal{N}}$, denoted by $X_N^{\mathcal{N}}$, with $N \leq N_{\max}$.

The surrogate for the truth discretization is denoted $u_N^{\mathcal{N}}(\mu)$ and is computed as a member of the reduced space $X_N^{\mathcal{N}}$. The space $X_N^{\mathcal{N}}$ is constructed in a hierarchical manner as the span of so-called “snapshots”, by hierarchically constructing a sample set $S^N = \{\mu^1, \dots, \mu^N\}$ from the training set $\mathcal{E}_{\text{train}}$ and solving (2.4) with $\mu = \mu^n$. Explicitly:

$$X_N^{\mathcal{N}} := \text{span}\{u^{\mathcal{N}}(\mu^n), 1 \leq n \leq N\}, \quad N = 1, \dots, N_{\max}. \quad (2.6)$$

Given $\mu \in \mathcal{D}$, we define the RBM solution $u_N^{\mathcal{N}}(\mu) \in X_N^{\mathcal{N}}$ as the solution to the following *reduced* problem

$$\begin{cases} \text{For } \mu \in \mathcal{D}, \text{ find the RB solution } u_N^{\mathcal{N}}(\mu) \in X_N^{\mathcal{N}} \subset X^{\mathcal{N}} \text{ such that} \\ a(u_N^{\mathcal{N}}, v; \mu) = f(v; \mu) \quad \forall v \in X_N^{\mathcal{N}}. \end{cases} \quad (2.7)$$

The truth system (2.4) is \mathcal{N} -dimensional, but the reduced system (2.7) is N -dimensional. When $N \ll \mathcal{N}$, this results in a significant computational savings. This saving is realized by precomputing and storing the parameter-independent components of the RB “stiffness” matrix which is decomposed via the affine assumption (2.3)

$$(a(v_m, v_n; \mu))_{N \times N} = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) (a^q(v_m, v_n))_{N \times N}. \quad (2.8)$$

The nested structure of $X_N^{\mathcal{N}}$ ($X_1^{\mathcal{N}} \subset X_2^{\mathcal{N}} \subset \dots \subset X_{N_{\max}}^{\mathcal{N}} \subset X^{\mathcal{N}}$) allows us to expand these parameter-independent components $(a^q(v_m, v_n))_{N \times N}$ by adding one more row and one more column each time a new sample location μ^i is selected and the new snapshot resolved.

2.3. Computational details

The Galerkin procedure in (2.7) can be transformed into an algorithm by using the following ansatz for the RBM solution:

$$u_N^{\mathcal{N}}(\mu) = \sum_{m=1}^N u_{Nm}^{\mathcal{N}}(\mu) u^{\mathcal{N}}(\mu^m). \quad (2.9)$$

Using this in (2.7) and choosing the test functions as $v = u^{\mathcal{N}}(\mu^{(n)})$ for $n = 1, \dots, N$, results in a solvable linear system for the unknown RB coefficients $\{u_{Nm}^{\mathcal{N}}(\mu)\}_{m=1}^N$, which defines $u_N^{\mathcal{N}}$. In practice, this strategy tends to produce ill-conditioned systems. To ameliorate this effect, practitioners usually choose an orthonormal basis for $X_N^{\mathcal{N}}$ for both trial and test functions:

$$u_N^{\mathcal{N}}(\mu) = \sum_{m=1}^N \hat{u}_{Nm}^{\mathcal{N}}(\mu) \xi_m, \quad \langle \xi_m, \xi_n \rangle_X = \delta_{n,m}, \quad \{\xi_m\}_{m=1}^N \subset X_N^{\mathcal{N}}. \quad (2.10)$$

The ξ_n are hierarchically computed by orthogonalizing the snapshots $u^{\mathcal{N}}(\mu^{(m)})$ each time S^N is updated. We note that computing the $\hat{u}_{Nm}^{\mathcal{N}}$ coefficients is mathematically equivalent to computing the $u_{Nm}^{\mathcal{N}}$ coefficients, as the two are related by a change of basis transformation: The computational representation (2.10) can be transformed into the representation (2.9) (and vice versa) through a linear transformation that we omit for brevity. The representation (2.10) is used in practical implementations of RBM algorithms (both in general and in the examples of this paper), but our discussion will use the formulation (2.9) because this formulation is more convenient to describe our residual-free error indicator in Section 3.3.

2.4. Selecting snapshots through the *a posteriori* error estimate

Here we describe the procedure for selecting the representative parameters μ^1, \dots, μ^N for the sample set S^N . This is an important task since these parameter choices define the reduced space (2.6). RBM adopts a greedy scheme to iteratively construct S^N , and leans on mathematical theory for the truth discretization (2.4). In particular, there exist efficiently-computable error estimates that quantify the discrepancy between the dimension- n RBM surrogate solution $u_n^N(\mu)$ and the truth solution $u^N(\mu)$. This error estimate is denoted Δ_n , and satisfies $\Delta_n(\mu) \geq \|u_n^N(\mu) - u^N(\mu)\|_{X^N}$. Assuming existence of this error estimate, the greedy procedure for constructing S^N is summarized in Algorithm 1.

Algorithm 1 Greedy algorithm for constructing S^N and X_n^N .

- 1: Input: training set $\mathcal{E}_{\text{train}}$, an accuracy tolerance ε_{tol} , maximum RB dimension N_{max} .
- 2: Randomly select the first sample $\mu^1 \in \mathcal{E}_{\text{train}}$, and set $n = 1$ and $\varepsilon = 2\varepsilon_{\text{tol}}$.
- 3: Obtain truth solution $u^N(\mu^1)$, and set $X_1^N = \text{span}\{u^N(\mu^1)\}$.
- 4: **while** ($\varepsilon > \varepsilon_{\text{tol}}$ and $n < N_{\text{max}}$) **do**
- 5: **for** each $\mu \in \mathcal{E}_{\text{train}}$ **do**
- 6: Obtain RBM solution $u_n^N(\mu) \in X_n^N$ and error estimate $\Delta_n(\mu)$
- 7: **end for**
- 8: $\mu^{n+1} = \underset{\mu \in \mathcal{E}_{\text{train}}}{\text{argmax}} \Delta_n(\mu)$, $\varepsilon = \Delta_n(\mu^{n+1})$.
- 9: Augment the sample set $S^{n+1} = S^n \cup \{\mu^{n+1}\}$ and the RB space $X_{n+1}^N = X_n^N \oplus \{u^N(\mu^{n+1})\}$.
- 10: Set $n \leftarrow n + 1$.
- 11: **end while**

The design and efficient implementation of the error bound Δ_n is usually accomplished with *a posteriori* error estimates from the truth discretization. Mathematical rigor of this estimate is crucial for the accuracy of the reduced basis solution. The main novelties of this paper are replacements of the “classical” *a posteriori* error estimate with alternatives that are either more efficient or have enhanced accuracy properties. We finish this section by describing how Δ_n is evaluated in the classical fashion.

The error between the reduced basis surrogate solution and the truth discretization is $e_N(\mu) := u^N(\mu) - u_N^N(\mu) \in X^N$. Unfortunately, this is not computable directly without knowledge of u^N , which we want to avoid computing. However, the linearity of $a(\cdot, \cdot; \mu)$ results in

$$a(e_N(\mu), v; \mu) = r_N(v; \mu) \quad \forall v \in X^N, \quad (2.11)$$

with the residual $r_N(v; \mu) \in (X^N)'$ (the dual of X^N) defined as $f(v; \mu) - a(u_N^N(\mu), v; \mu)$. The Riesz representation theorem and a variational inequality imply that $\|e_N(\mu)\|_{X^N} \leq \frac{\|r_N(\cdot; \mu)\|_{(X^N)'}}{\alpha^N(\mu)}$, where $\alpha^N(\mu) = \inf_{w \in X^N} \frac{a(w, w; \mu)}{\|w\|_X^2}$ is the stability (coercivity) constant for the elliptic bilinear form a . Therefore, we can define an *a posteriori* RBM error estimator as

$$\Delta_N(\mu) = \frac{\|r_N(\cdot; \mu)\|_{(X^N)'}}{\alpha_{LB}^N(\mu)} \geq \|e_N(\mu)\|_{X^N}. \quad (2.12)$$

Here $\alpha_{LB}^N(\mu)$ is a lower bound for $\alpha^N(\mu)$ which is expensive to compute directly for all μ . However, approaches for computationally efficient estimation of the stability factor $\alpha_{LB}^N(\mu)$ has been undergoing vast development in [19,20] and recently in [21–23]. Furthermore, an offline–online decomposition is also exploited to speed up the evaluation of $\alpha_{LB}^N(\mu)$.

The remaining ingredient for efficient computation of the *a posteriori* error estimation is the evaluation of $\|r_N(\cdot; \mu)\|_{(X^N)'}$, which is the chief concern of this manuscript. The following discussion details how this is achieved in a standard RB implementation. The Riesz representation theorem states that we can calculate functions $C^{\tilde{q}} \in X^N$ and $\mathcal{L}_m^q \in X^N$, for $1 \leq \tilde{q} \leq Q_f$, $1 \leq m \leq N$, $1 \leq q \leq Q_a$, such that

$$\begin{cases} (C^{\tilde{q}}, v)_{X^N} = f^{\tilde{q}}(v)_{X^N} & \forall v \in X^N \\ (\mathcal{L}_m^q, v)_{X^N} = a^q(u^N(\mu^m), v) & \forall v \in X^N. \end{cases} \quad (2.13)$$

With the availability of $C^{\tilde{q}}$ and \mathcal{L}_m^q , the classical implementation of RBM then adopts an offline–online decomposition for $\|r_N(\cdot; \mu)\|_{(X^N)'}$. Combining (2.11), (2.9), and (2.3), we have

$$\begin{aligned} \|r_N(\cdot; \mu)\|_{(X^N)' }^2 &= \sum_{\tilde{q}_1=1}^{Q_f} \sum_{\tilde{q}_2=1}^{Q_f} \Theta_f^{\tilde{q}_1}(\mu) \Theta_f^{\tilde{q}_2}(\mu) (C^{\tilde{q}_1}, C^{\tilde{q}_2})_{X^N} + \\ &\quad \sum_{q=1}^{Q_a} \sum_{m=1}^N \Theta_a^q(\mu) u_{Nm}^N \left\{ \sum_{q'=1}^{Q_a} \sum_{m'=1}^N \Theta_a^{q'}(\mu) u_{Nm'}^N (\mathcal{L}_m^q, \mathcal{L}_{m'}^{q'})_{X^N} \right\} - 2 \sum_{q=1}^{Q_a} \sum_{m=1}^N \sum_{\tilde{q}=1}^{Q_f} \Theta_a^q(\mu) u_{Nm}^N (\mathcal{L}_m^q, C^{\tilde{q}})_X. \end{aligned} \quad (2.14)$$

Most of the quantities above can be precomputed and stored explicitly from the form of the pPDE. Therefore, the offline stage is devoted to calculating and storing

$$(\mathcal{C}^{\tilde{q}_1}, \mathcal{C}^{\tilde{q}_2})_{X^N}, (\mathcal{C}^{\tilde{q}_1}, \mathcal{L}_m^q)_{X^N}, (\mathcal{L}_m^q, \mathcal{L}_{m'}^{q'})_{X^N}, \quad 1 \leq m, m' \leq N_{\text{RB}}, \quad 1 \leq \tilde{q}_1, \tilde{q}_2 \leq Q_f, \quad 1 \leq q, q' \leq Q_a.$$

During the online stage, given any parameter μ , we only need to evaluate $\Theta_a^q(\mu)$, $\Theta_f^{\tilde{q}}(\mu)$, $u_{Nm}^N(\mu)$, $1 \leq m \leq N$, and perform the addition and subtraction according to (2.14). However, the coefficients u_{Nm}^N are the RBM expansion coefficients from (2.9), and therefore evaluation of (2.14) at each μ also requires computation of the RB solution u_N^N . We denote this numerical approximation of $\Delta_n(\mu)$ in (2.12) by $\varepsilon_1(\mu; n)$.

3. Novel approaches for error quantification

As shown by Algorithm 1, the classical RBM computes the maximum of the error estimator $\Delta_n(\cdot)$ over $\mathcal{E}_{\text{train}}$ in a brute-force manner. $\mathcal{E}_{\text{train}}$ is large especially for a high-dimensional parameter domain. Moreover, this maximization procedure must be done for every $n = 1, \dots, N$. For these two reasons, the process of selecting snapshots is usually the computational bottleneck of RBM algorithms. Another observation is that the dual norm of the residual is computed as a square root of its square which is evaluated as the difference between a sum of two positive terms and a third term. This is prone to loss of significance and one suspects that errors smaller than the square root of machine precision are not computable using the form (2.14). This supposition is borne out in numerical results for RBM.

In this section, we provide two novel approaches to mitigate the deficiencies of classical RBM residual estimation. We begin first by discussing why the classical approach for evaluation of the error estimate via the formula (2.14) is not ideal when implemented in finite-precision arithmetic. We follow this by presenting our robust residual-based error estimate, which directly evaluates $\|r_N(\cdot; \mu)\|_{(X^N)^*}$ without computing its squared value; this allows the new method to achieve errors much smaller than root machine precision. Finally, we present our residual-free approach that uses a surrogate for the residual that circumvents requirement of computing $\alpha_{LB}(\cdot)$. Thus the second approach is applicable to pPDE where no rigorous *a posteriori* error estimate is available. Our preliminary analysis and numerical experiments suggest that the residual-free method is promising, but rigorous theoretical analysis demonstrating its utility in the RBM setting is the subject of ongoing work.

3.1. Finite-precision limitations for residual norm evaluation

The formula (2.14) is an expanded quadratic form for the expression

$$\|r_N(\cdot; \mu)\|_{(X^N)^*}^2 = \|a(u_N^N - u^N, \cdot; \mu)\|_{(X^N)^*}^2 = \|a(u_N^N, \cdot; \mu) - f(\cdot; \mu)\|_{(X^N)^*}^2 \quad (3.1)$$

At the root of the floating-point stagnation is evaluation of the above quadratic form via the formula

$$(a - b)^2 = a^2 - 2ab + b^2.$$

Indeed, the right-hand side expansion of the above equation is (2.14). A simple floating point error analysis reveals the following lemma, where we use ϵ to denote the machine precision, and $fl(\cdot)$ to denote the floating point representation of a number.

Lemma 3.1. *In the case of $b = a + \mathcal{O}(\epsilon)$, we have that*

$$fl(a^2 - 2ab + b^2) = \mathcal{O}(\epsilon),$$

while

$$fl((a - b)^2) = \mathcal{O}(\epsilon^2).$$

Proof. Assume $fl(x) = x(1 + C\epsilon)$, where C is an ϵ -independent constant that does depend on x . We have $|C| \leq 1$, and in the sequel we use C_j for $j = 1, 2, \dots$, to denote various such constants. Since $a - b = \mathcal{O}(\epsilon)$, we have

$$\begin{aligned} fl(a^2 - 2ab + b^2) &= (a^2(1 + C_2\epsilon) - 2ab(1 + C_3\epsilon) + b^2(1 + C_4\epsilon))(1 + C_1\epsilon) \\ &= (a - b)^2 + ((a - b)^2C_1 + a^2C_2 - 2abC_3 + b^2C_4)\epsilon + (\mathcal{O}(\epsilon))^2 \\ &= (\mathcal{O}(\epsilon))^2 + \mathcal{O}(\epsilon) \\ &= \mathcal{O}(\epsilon). \end{aligned}$$

showing the first equation in the Lemma's conclusion. The second equation is straightforward since $fl((fl(a) - fl(b))^2) = (a(1 + C_a\epsilon) - b(1 + C_b\epsilon))^2(1 + C\epsilon) = \epsilon^2(1 + (aC_a + bC_b))^2(1 + C\epsilon) = \mathcal{O}(\epsilon^2)$. \square

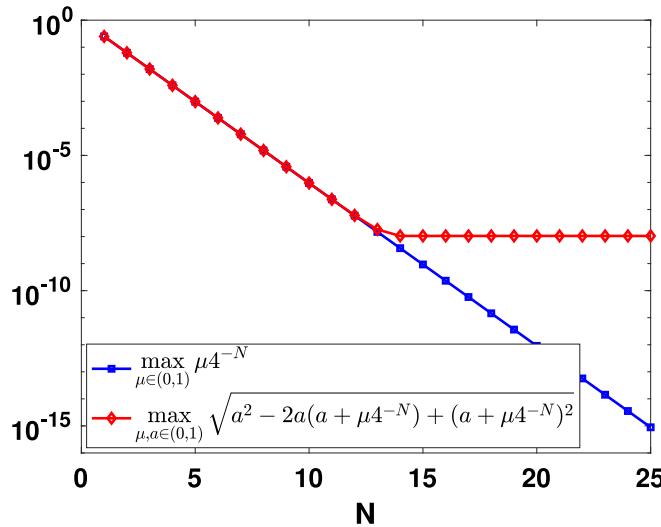


Fig. 1. The loss of accuracy for the formula $(a-b)^2 = a^2 - 2ab + b^2$, with a randomly chosen from $(0, 1)$ and $b = a + \mu 4^{-N}$. Implemented in double precision, we have $\sqrt{\epsilon} \approx 10^{-8}$. We see that the $a^2 - 2ab + b^2$ formula stagnates around this level.

The technical conclusion of this lemma can be communicated visually via Fig. 1. In this Figure, we introduce a parameter $\mu \in (0, 1)$ to emulate the RBM setting, and set $b = a + \mu 4^{-N}$. We numerically compute the μ -maximum of the expressions $\sqrt{(a-b)^2} = \mu 4^{-N}$ and $\sqrt{a^2 - 2ab + b^2}$. The figure results demonstrate that the expression $\sqrt{a^2 - 2ab + b^2}$, representing the direct RBM estimate (2.14), stagnates around $\sqrt{\epsilon}$, which is about 10^{-8} in IEEE double-precision. Direct usage of $\sqrt{(a-b)^2}$ does not suffer from this stagnation. This motivates the need for a robust approach to handle this case. One approach is described in [12], and we describe a new method in the next two sections that also circumvents this issue.

3.2. A new evaluation of the residual norm

An intuitive explanation of our approach is as follows: Fix $u_N^{\mathcal{N}}$ and consider the right-hand side expression in (3.1). We define $f(\cdot; \mu) = f_{\parallel}(\cdot; \mu) + f_{\perp}(\cdot; \mu)$, where $f_{\parallel}(\cdot; \mu)$ is a component of $f(\cdot; \mu)$ that is parallel to $a(u_N^{\mathcal{N}}, \cdot; \mu)$ and $f_{\perp}(\cdot; \mu)$ is a component that is perpendicular to $a(u_N^{\mathcal{N}}, \cdot; \mu)$. Therefore, Eq. (3.1) can be rewritten as

$$\|r_N(\cdot; \mu)\|_{(X^{\mathcal{N}})'}^2 = \| [a(u_N^{\mathcal{N}}, \cdot; \mu) - f_{\parallel}(\cdot; \mu)] + f_{\perp}(\cdot; \mu) \|_{(X^{\mathcal{N}})'}^2.$$

Our improvement to the standard evaluation (2.14) computes the residual norm above by the following observation: Since the two separated terms under the norm are perpendicular, we can separate them via the Pythagorean theorem. We now present the details; writing the residual norm as a norm in $X^{\mathcal{N}}$ instead of in the dual $(X^{\mathcal{N}})'$, we have

$$\|r_N(\cdot; \mu)\|_{(X^{\mathcal{N}})'} = \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) C^q - \sum_{q=1}^{Q_a} \sum_{m=1}^N \Theta_a^q(\mu) u_{Nm}^{\mathcal{N}} \mathcal{L}_m^q \right\|_{X^{\mathcal{N}}}. \quad (3.2)$$

Assume that $\mathcal{N} > Q_a N$, which is a reasonable assumption in the RBM framework. We introduce the following subspace in the Hilbert space $X^{\mathcal{N}}$:

$$V_N := \text{span} \left\{ \mathcal{L}_1^1, \dots, \mathcal{L}_1^{Q_a}, \dots, \mathcal{L}_N^1, \dots, \mathcal{L}_N^{Q_a} \right\},$$

Defining V_N^{\perp} as the $X^{\mathcal{N}}$ -orthogonal complement of V_N , then let P_N be the $X^{\mathcal{N}}$ -orthogonal projection onto V_N , and let P_N^{\perp} be the orthogonal projection onto V_N^{\perp} .

We are now ready to state a preliminary result:

Lemma 3.2. *The dual norm of the residual (3.2) has the form*

$$\|r_N(\cdot; \mu)\|_{(X^{\mathcal{N}})'} = \sqrt{ \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) P_N^{\perp} C^q \right\|_{X^{\mathcal{N}}}^2 + \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) P_N C^q - \sum_{q=1}^{Q_a} \sum_{m=1}^N \Theta_a^q(\mu) u_{Nm}^{\mathcal{N}} \mathcal{L}_m^q \right\|_{X^{\mathcal{N}}}^2 }. \quad (3.3)$$

Proof. The result is a fairly straightforward computation in least-squares problems. We have

$$\begin{aligned} r_N(\cdot, \mu) &= \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathcal{C}^q - \sum_{q=1}^{Q_a} \sum_{m=1}^N \Theta_a^q(\mu) u_{Nm}^{\mathcal{N}} \mathcal{L}_m^q \\ &= \sum_{q=1}^{Q_f} \Theta_f^q(\mu) P_N^{\perp} \mathcal{C}^q + \left(\sum_{q=1}^{Q_f} \Theta_f^q(\mu) P_N \mathcal{C}^q - \sum_{q=1}^{Q_a} \sum_{m=1}^N \Theta_a^q(\mu) u_{Nm}^{\mathcal{N}} \mathcal{L}_m^q \right). \end{aligned}$$

Note that the first term is an element of P_N^{\perp} and the second term in parenthesis is an element of P_N . The conclusion follows from the Pythagorean theorem. \square

This Lemma yields a computational procedure that ameliorates finite-precision loss of significance in numerical implementations.

3.2.1. Implementation and offline–online decomposition

In this section we treat elements of $X^{\mathcal{N}}$ as Euclidean vectors in $\mathbb{R}^{\mathcal{N}}$ and identify the norm $\|\cdot\|_{X^{\mathcal{N}}}$ with the standard ℓ^2 norm $\|\cdot\|$. When the norm $\|\cdot\|_{X^{\mathcal{N}}}$ is different than the ℓ^2 norm, the below discussion would proceed by inserting Gramian square root matrices in appropriate places so that the resulting weighted ℓ^2 norm equals the norm on $X^{\mathcal{N}}$.

The discretized version of $\mathcal{L}_m^q \in X^{\mathcal{N}}$, is $\vec{\mathcal{L}}_m^q \in \mathbb{R}^{\mathcal{N}}$, a $\mathcal{N} \times 1$ vector. Similarly, we let $\vec{\mathcal{C}}^q \in \mathbb{R}^{\mathcal{N}}$ denote the vector representation of \mathcal{C}^q . We use the $\vec{\mathcal{L}}_m^q$ vectors to define a matrix \mathcal{B} and its associated column-pivoted reduced QR factorization:

$$\begin{aligned} \mathcal{B} &= \left(\vec{\mathcal{L}}_1^1, \dots, \vec{\mathcal{L}}_1^{Q_a}, \dots, \vec{\mathcal{L}}_N^1, \dots, \vec{\mathcal{L}}_N^{Q_a} \right) \in \mathbb{R}^{\mathcal{N} \times Q_a N}, \\ \mathcal{B}\mathcal{Z} &= \mathcal{Q}\mathcal{R}, \quad \mathcal{Q} \in \mathbb{R}^{\mathcal{N} \times \text{rank}(\mathcal{B})}, \quad \mathcal{R} \in \mathbb{R}^{\text{rank}(\mathcal{B}) \times Q_a N}, \end{aligned}$$

where $\mathcal{Z} \in \mathbb{R}^{Q_a N \times Q_a N}$ is a permutation matrix obtained from the column-pivoted reduced QR factorization above. Define $\vec{c}(\mu)$ the column vector

$$\vec{c}(\mu) = (\Theta_a^1(\mu) u_{N1}^{\mathcal{N}}, \dots, \Theta_a^{Q_a}(\mu) u_{N1}^{\mathcal{N}}, \dots, \Theta_a^1(\mu) u_{NN}^{\mathcal{N}}, \dots, \Theta_a^{Q_a}(\mu) u_{NN}^{\mathcal{N}})^T \in \mathbb{R}^{Q_a N}.$$

Finally, define $V_N \subset X^{\mathcal{N}}$ as the column space of \mathcal{Q} , along with some associated projection matrices:

$$V_N := \text{range}(\mathcal{Q}), \quad \mathcal{P}_N = \mathcal{Q}\mathcal{Q}^T, \quad \mathcal{P}_N^{\perp} = I - \mathcal{P}_N =: \mathcal{W}\mathcal{W}^T.$$

We remark that we allow the column space of \mathcal{Q} to be less than the number of columns of \mathcal{B} through, e.g. a rank-revealing QR factorization. In cases where this rank deficiency is utilized, this makes our algorithm more online-efficient than that of [12], where a Gram–Schmidt step with re-iteration is used on the full dimension of the column space. With V_N^{\perp} as the $\mathbb{R}^{\mathcal{N}}$ -orthogonal complement of V_N , then \mathcal{P}_N orthogonally projects onto V_N , and \mathcal{P}_N^{\perp} orthogonally projects onto V_N^{\perp} . The columns of $\mathcal{W} \in \mathbb{R}^{\mathcal{N} \times (Q_a N - \text{rank}(\mathcal{B}))}$ are formed from any orthonormal basis for V_N^{\perp} . We can now state conclusion of Lemma 3.2 in terms of vectors and matrices:

Theorem 3.3. *The dual norm of the residual (3.2) can be evaluated by*

$$\|r_N(\cdot, \mu)\|_{(X^{\mathcal{N}})^*}^2 = \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathcal{W}^T \vec{\mathcal{C}}^q \right\|^2 + \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathcal{Q}^T \vec{\mathcal{C}}^q - \mathcal{R} \mathcal{Z}^T \vec{c}(\mu) \right\|^2. \quad (3.4)$$

Proof. The second term in this result equals the second term in (3.3) due to the fact that $\mathcal{Q}^T \mathcal{Q} = I$ and that \mathcal{Q}^T is an isometric map on the range of \mathcal{Q} :

$$\begin{aligned} \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathcal{P}_N \mathcal{C}^q - \sum_{q=1}^{Q_a} \sum_{m=1}^N \Theta_a^q(\mu) u_{Nm}^{\mathcal{N}} \mathcal{L}_m^q \right\|_{X^{\mathcal{N}}}^2 &= \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathcal{P}_N \vec{\mathcal{C}}^q - \mathcal{B} \vec{c}(\mu) \right\|^2 \\ &= \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathcal{Q} \mathcal{Q}^T \vec{\mathcal{C}}^q - \mathcal{Q} \mathcal{R} \mathcal{Z}^T \vec{c}(\mu) \right\|^2 \\ &= \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathcal{Q}^T \vec{\mathcal{C}}^q - \mathcal{R} \mathcal{Z}^T \vec{c}(\mu) \right\|^2 \end{aligned}$$

That the first term in the result equals the first term in (3.3) is the result of a similar computation but using \mathcal{W} instead of \mathcal{Q} :

$$\begin{aligned} \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) \mathcal{P}_N^\perp \vec{c}^q \right\|^2 &= \left\| \mathcal{W} \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) \mathcal{W}^T \vec{c}^q \right\|^2 \\ &= \left\| \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) \mathcal{W}^T \vec{c}^q \right\|^2. \quad \square \end{aligned}$$

The theorem above immediately reveals an offline–online decomposition: The $\boldsymbol{\mu}$ -independent parts of the formula involve the matrix products

$$\mathcal{W}^T \vec{c}^q, \quad \mathcal{Q}^T \vec{c}^q, \quad \mathcal{R} \mathcal{Z}^T,$$

none of which have any dimensions dependent on \mathcal{N} , but are simply dependent on N and Q_a . Thus, these matrices may be precomputed and stored independent of $\boldsymbol{\mu}$. One $\boldsymbol{\mu}$ -dependent component involves the affine coefficients $\Theta_f^q(\boldsymbol{\mu})$, which are scalar and thus easy to compute. Finally, the coefficients $\vec{c}(\boldsymbol{\mu})$ can be computed explicitly via the affine coefficients $\Theta_a^q(\boldsymbol{\mu})$ and from the RBM approximation $u_N^{\mathcal{N}}$ in (2.9). Since computing RBM approximation is \mathcal{N} -independent, the entire residual norm computation via (3.4) is \mathcal{N} -independent in the online phase. We denote $\Delta_N(\boldsymbol{\mu})$ in (2.12) with residual norm computed via this new approach by $\mathcal{E}_2(\boldsymbol{\mu}; n)$.

3.3. A residual-free error indicator

Much of the RBM algorithm is dependent on rigor of the inequality in (2.12). However, in practical situations one may not have access to such a computable *a posteriori* error estimator. This happens, for instance, with sufficiently complicated nonlinear pPDE's for which mathematical analysis is difficult or infeasible. In other situations, a rigorous error estimate like (2.12) may exist, but is not easily computable in a \mathcal{N} -independent fashion due to nonlinearity of the pPDE. In this case, the offline parameter selection portion of the RBM algorithm may be so expensive as to outweigh any computational saving gained during the online phase.

In either of the cases above, one still hopes to use an efficient model-order reduction strategy like RBM, but with an understanding that mathematically rigorous error certification may be lost.

A strategy for devising an error estimate for such a case is the subject of this section. While presumably one always has access to the pPDE residual, one could then use $\|r_N(\cdot; \boldsymbol{\mu})\|_{\mathcal{N}}$ as an error estimator that is not mathematically rigorous. However, this estimator may also not be \mathcal{N} -independent, so that such an estimator is both non-rigorous and expensive.

The alternative we suggest is as follows: recall expression (2.9) that expresses the RBM approximation $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ in terms of the snapshots $u^{\mathcal{N}}(\boldsymbol{\mu}^n)$, $n = 1, \dots, N$, via the expansion coefficients $\{u_{Nm}^{\mathcal{N}}(\boldsymbol{\mu})\}_{m=1}^N$. As indicated, these expansion coefficients are functions of $\boldsymbol{\mu}$, and play role of basis functions. Furthermore, they satisfy

$$u_{Nm}^{\mathcal{N}}(\boldsymbol{\mu}^n) = \delta_{n,m}, \tag{3.5}$$

where $\delta_{n,m}$ is the Kronecker delta. The above property is a direct consequence of condition (2.7) that defines the RBM solution. Therefore, the coefficient functions $u_{Nm}^{\mathcal{N}}(\cdot)$ are actually cardinal Lagrange interpolants associated to the space of functions defined by their span. Note that, given any $\boldsymbol{\mu}$, the cost of evaluation of these cardinal functions does *not* depend on \mathcal{N} , since these coefficients are computed from the RBM solution.

We now rephrase the essential portion of the offline RBM phase: given the current parameter values $\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N$ along with the current subspace of parameter-dependent functions

$$\text{span} \{u_{N1}^{\mathcal{N}}(\cdot), \dots, u_{NN}^{\mathcal{N}}(\cdot)\},$$

can we compute the next parameter value $\boldsymbol{\mu}^{N+1}$? Abstractly, this can be interpreted as an interpolation problem to find a nested sequence of interpolation points $\boldsymbol{\mu}^j, \boldsymbol{\mu}^{j+1}, \dots$, associated to a nested sequence of function spaces.

We consider one potential solution to this problem, inspired by concepts in polynomial approximation. We take the next point $\boldsymbol{\mu}^{N+1}$ as the point that maximizes a function of these cardinal interpolants:

$$\tilde{\Delta}_N(\boldsymbol{\mu}) = \left(\sum_{m=1}^N |u_{Nm}^{\mathcal{N}}(\boldsymbol{\mu})| \right). \tag{3.6}$$

The above function is simply the Lebesgue function from interpolation theory (i.e., the norm of an interpolation operator). Note that evaluation of this function depends only on the RBM solution, and does not directly involve computation of residual norms, nor does it require mathematically rigorous *a posteriori* error estimates. We show in Lemma 3.4 that $\tilde{\Delta}_N$ does indeed match the behavior of $e_N(\boldsymbol{\mu})$; it is therefore quite useful in selecting RBM parameter values to compute snapshots. However, the relationship between e_N and $\tilde{\Delta}_N$ involves a multiplicative scaling constant that is in general an uncomputable best approximation error. Since we cannot compute this scaling constant, we cannot certify the error committed by parameter

values picked with this method. However, we show in our numerical results section that choosing parameter samples via greedy maximization of (3.6) empirically produces results comparable to using the RBM error estimate (2.12). Our ongoing work seeks to “certify” the surrogate error generated via this approach. A simple approach for approximating a lower bound for ϵ_N can be $\|u^N(\mu^{N+1}) - u_N^N(\mu^{N+1})\|_X$, which is a computable quantity requiring no additional PDE simulations in the RBM context. Naturally, the effectiveness of this and related approaches will need to be carefully studied in the future.

We emphasize that this procedure is similar to, but distinct from, empirical interpolation procedures [18]. In empirical interpolation, one essentially has an $(N+1)$ -dimensional space with N points, and uses the discrepancy between the $(N+1)$ -dimensional space and the N points to pick the $(N+1)$ st point. We cannot do this here since the $(N+1)$ -dimensional space depends explicitly on the sought point μ^{N+1} . Thus, our strategy using the objective (3.6) circumvents the needs for identification of the higher-dimensional space.

3.3.1. Characterization of $\tilde{\Delta}_N$

We now motivate the choice of $\tilde{\Delta}_N(\mu)$ as an error indicator. We introduce the space of X^N -valued functions in $L^\infty(\mathcal{D})$:

$$L^\infty(\mathcal{D}, X^N) = \left\{ u : \mathcal{D} \rightarrow X^N \mid \|u\|_{L^\infty(\mathcal{D}, X^N)} < \infty \right\}, \quad \|u\|_{L^\infty(\mathcal{D}, X^N)} := \sup_{\mu \in \mathcal{D}} \|u(\mu)\|_{X^N}.$$

A subspace of particular interest is those functions in $L^\infty(\mathcal{D}, X^N)$ whose μ -variation is prescribed by the cardinal functions u_{Nm}^N :

$$U_N = \left\{ u = \sum_{m=1}^N y_m u_{Nm}^N(\mu) \mid y_1, \dots, y_N \in X^N \right\} \subset L^\infty(\mathcal{D}, X^N).$$

Note that the RBM solution is in U_N , and the truth solution is in $L^\infty(\mathcal{D}, X^N)$:

$$u_N^N \in U_N, \quad u^N \in L^\infty(\mathcal{D}, X^N).$$

The former is true by inspection of (2.9), and the latter is true because of the truth discretization versions of the uniform ellipticity and continuity assumptions (2.2a) – (2.2c). The following result then applies.

Lemma 3.4. *Let $\|e_N(\mu)\|_{X^N} = \|u_N^N(\mu) - u^N(\mu)\|_{X^N}$ be the RBM error committed at parameter value μ . Then*

$$\|e_N(\mu)\|_{X^N} \leq (1 + \tilde{\Delta}_N(\mu)) \epsilon_N(u^N),$$

where ϵ_N is the best approximation error.

$$\epsilon_N(u^N) := \inf_{v \in U_N} \|u^N - v\|_{L^\infty(\mathcal{D}, X^N)}.$$

Proof. The result is an exercise in a pointwise version of Lebesgue’s Lemma for projective approximations. We first define two operators. The first, $P_N : L^\infty(\mathcal{D}, X^N) \rightarrow U_N$, is the interpolative projection operator defined by

$$P_N v = \sum_{m=1}^N v(\mu^m) u_{Nm}^N(\mu), \quad v \in L^\infty(\mathcal{D}, X^N).$$

The second operator is $\delta_\mu : L^\infty(\mathcal{D}, X^N) \rightarrow X^N$, corresponding to point-evaluation at $\mu \in \mathcal{D}$:

$$\delta_\mu v = v(\mu), \quad v \in L^\infty(\mathcal{D}, X^N).$$

Note that

$$P_N u^N(\mu) = \sum_{m=1}^N u^N(\mu^m) u_{Nm}^N(\mu) = u_N^N(\mu).$$

Now let v be any element in U_N , so that $P_N v = v$. Then

$$\begin{aligned} \|e_N(\mu)\|_{X^N} &= \|u_N^N(\mu) - u^N(\mu)\|_{X^N} \leq \|u_N^N(\mu) - v(\mu)\|_{X^N} + \|v(\mu) - u^N(\mu)\|_{X^N} \\ &= \|P_N[u^N - v](\mu)\|_{X^N} + \|u^N(\mu) - v(\mu)\|_{X^N} \\ &= \|\delta_\mu P_N[u^N - v]\|_{X^N} + \|u^N(\mu) - v(\mu)\|_{X^N} \\ &\leq [1 + \|\delta_\mu P_N\|] \|u^N - v\|_{L^\infty(\mathcal{D}, X^N)}, \end{aligned} \tag{3.7}$$

where $\|\delta_\mu P_N\|$ is the induced operator norm. We can directly compute

$$\begin{aligned}\|\delta_\mu P_N\| &= \sup_{\|w\|_{L^\infty(\mathcal{D}, X^N)}=1} \|\delta_\mu P_N w\|_{X^N} \\ &= \sup_{\|w\|_{L^\infty(\mathcal{D}, X^N)}=1} \left\| \sum_{m=1}^N u_{Nm}^N(\mu) w(\mu^m) \right\|_{X^N} \\ &\leq \sum_{m=1}^N |u_{Nm}^N(\mu)| \sup_{\|w\|_{L^\infty(\mathcal{D}, X^N)}=1} \|w(\mu^m)\|_{X^N} \\ &\leq \sum_{m=1}^N |u_{Nm}^N(\mu)| = \tilde{\Delta}_N(\mu).\end{aligned}$$

Using the above in (3.7) and infimizing over v yields

$$\|e_N(\mu)\|_{X^N} \leq [1 + \tilde{\Delta}(\mu)] \inf_{v \in U_N} \|v - u^N\|_{L^\infty(\mathcal{D}, X^N)} = [1 + \tilde{\Delta}_N(\mu)] \epsilon_N(u^N). \quad \square$$

Lemma 3.4 states that, relative to the best approximant from U_N , the RBM solution u_N^N commits a μ -pointwise error that scales monotonically with $\tilde{\Delta}_N(\mu)$. This is, essentially, a generalization of Lebesgue's Lemma in approximation theory.

Therefore, at any iteration n , the function $\tilde{\Delta}_n(\mu)$ gives a qualitative indication of the error at μ , and so choosing a new snapshot parameter μ^{n+1} at the maximum of this function is a greedy function that indirectly seeks to minimize the RBM error. However, it does not provide a certifiable error since we cannot compute the value of ϵ_N . Furthermore, our Lemma shows only that our residual-free error estimate is an upper bound for the true error; a rigorous procedure would also establish that our error estimate is a lower bound. Without this lower bound guarantee, one can contrive situations where our error estimate chooses parameter values that are not indicative of RBM subspace quality. However, we have not observed this in any examples we have tried. Our analysis cannot currently exclude the possibility of such pathological problems, and our ongoing work seeks to establish a lower bound estimate that would rigorously justify our residual-free objective.

4. Numerical results

In this section, we present numerical examples to demonstrate the accuracy and efficiency of the proposed two new approaches. The difference between all these approaches appears only in how parameter values μ^j are selected for use in the RBM algorithm. We have three approaches to compare: (i) The standard RBM strategy that uses $\Delta_N(\mu)$ for the greedy objective as defined in (2.12) with the residual norm computed using (2.14). (ii) The RBM strategy that again uses $\Delta_N(\mu)$ from (2.12) as the greedy objective, but now uses the formula (3.3) to compute the residual norm. (iii) The greedy objective is $\tilde{\Delta}_N(\mu)$, as defined in (3.6).

Since the parameter values selected by each of these three procedures is different, we use the subscript $i = 1, 2, 3$, to differentiate quantities for these methods. We denote these error bounds are $\epsilon_i(\mu)$ for $i = 1, 2, 3$. More specifically,

$$\epsilon_1(\mu) = \Delta_N(\mu) \text{ defined in (2.12)} \quad (4.1)$$

$$\epsilon_2(\mu) = \frac{\|r_N(\cdot; \mu)\|_{(X^N)^*}}{\alpha_{LB}^N(\mu)}, \text{ where } \|r_N(\cdot; \mu)\|_{(X^N)^*} \text{ is computed in (3.3)} \quad (4.2)$$

$$\epsilon_3(\mu) = \tilde{\Delta}_N(\mu) \quad (4.3)$$

Note that the first two approaches have rigorous error certification values. We use $u_{N, \epsilon_i}^N(\mu)$ to denote the RBM approximation using greedy strategy i , for $i = 1, 2, 3$. Since the focus of this paper is on the computation of $\|r_N(\cdot; \mu)\|_{(X^N)^*}$ for ϵ_1 and ϵ_2 and the elimination thereof for ϵ_3 , we compute $\alpha_{LB}^N(\mu)$ through direct resolution of the eigenvalue problems.

We test the three implementations on three problems, and present the results respectively in each subsection below.

4.1. Two cases with 1-dimensional parameter

We first test the three RB methods on the following equations with one parameter.

$$(1 + \mu x)u_{xx} + u_{yy} = e^{4xy} \quad \text{on } \Omega. \quad (4.4a)$$

$$(1 + \ell(\mu)x)u_{xx} + u_{yy} = e^{4xy} \quad \text{on } \Omega. \quad (4.4b)$$

The first equation has continuous dependence on the parameter while the second has discontinuous dependence by having

$$\ell(\mu) = \sin\left((\mu - \text{sign}(\mu))\frac{\pi}{2}\right), \quad \mu \in \mathcal{D}.$$

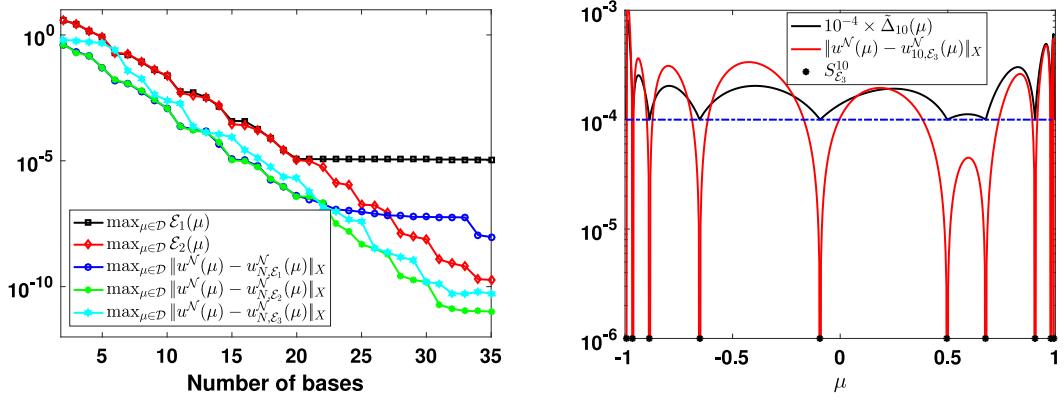


Fig. 2. Results for one-parameter case (4.4a): The comparison of the three approaches (Left) and a demonstration that the residual-free error indicator matches with the true error well (Right).

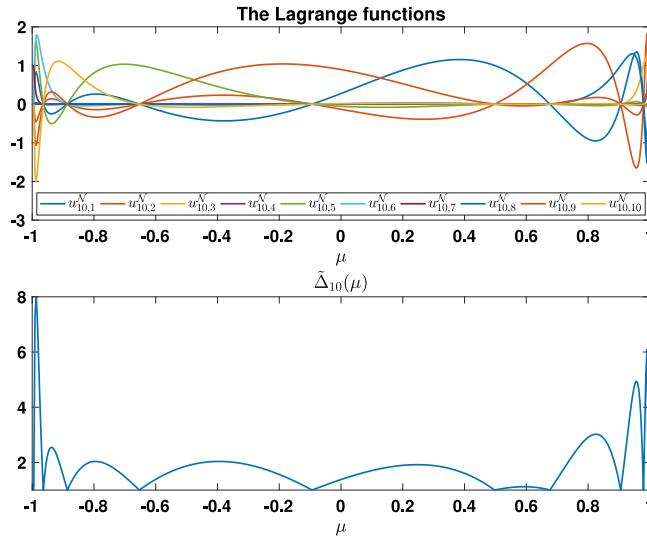


Fig. 3. The Lagrange shape functions.

We take the physical domain as $\Omega = [-1, 1] \times [-1, 1]$ and impose homogeneous Dirichlet boundary conditions on $\partial\Omega$. The truth approximation is a spectral Chebyshev collocation method based on 50 degrees of freedom for each direction. The parameter domain \mathcal{D} for μ is taken to be $[-0.995, 0.995]$, and the training set $\mathcal{E}_{\text{train}}$ a uniform Cartesian grid with 512 equally spaced points.

We show in Fig. 2 (left) the history of convergence for the three approaches. The classical approach stagnates before reaching the square root of machine epsilon as expected. However, both new approaches have worst-case error estimate and corresponding exact error converging further toward machine accuracy. It is worth noting that the residual-free function $\tilde{\Delta}_N(\mu)$ used in the greedy scheme for method $i = 3$ is not a rigorous error bound. However, the RB space built from its maximizers has similar approximation properties, as confirmed by the cyan curve in Fig. 2 (left). The fact that $\tilde{\Delta}_N$ captures the pattern of the true error as the parameter μ varies is shown in Fig. 2 (right). Here the black $\tilde{\Delta}_{10}(\mu)$ curve is multiplied by 10^{-4} to achieve better alignment with the error curve. Finally, we show in Fig. 3 the 10 Lagrange shape functions $u_{Nm}^N(\cdot)$ as implicitly defined in (2.9) and (2.7) when a 10-dimensional RB space is used.

The Lagrange shape functions used in the residual-free method inherit the structure dictated by the PDE. For example, they are discontinuous if the PDE enforces this. The second example (4.4b) has discontinuous parameter dependence, and we show the associated Lagrange shape functions for this case in Fig. 4. The results show both that convergence is not directly affected by the discontinuity since the Lagrange functions now inherit discontinuous dependence from the PDE.

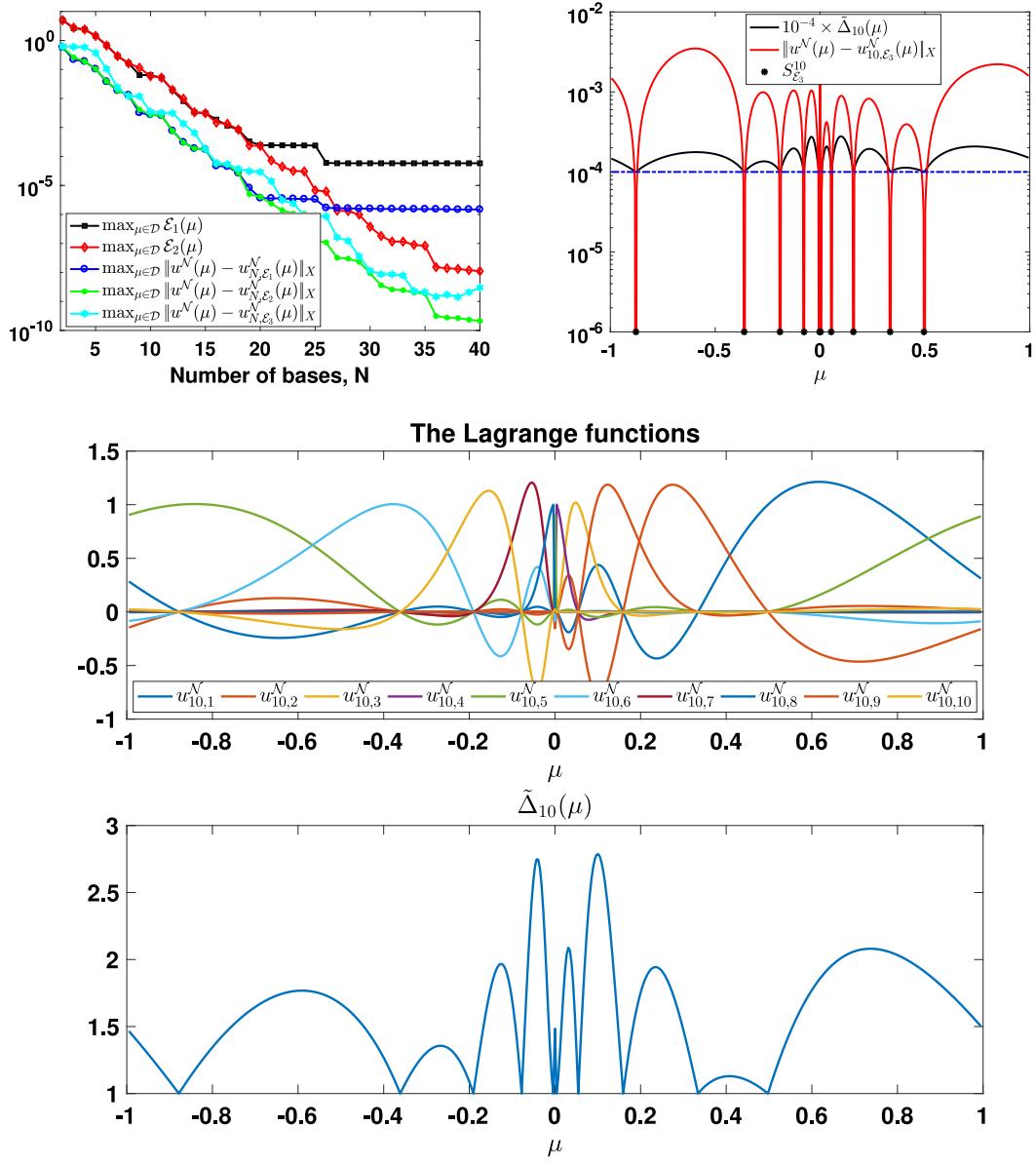


Fig. 4. Results for the one-dimensional case with discontinuous dependence on parameter (4.4b). Top left demonstrates us the comparison of histories of convergence for the three approaches. Top right demonstrates that the residual-free error indicator roughly matches with the true error well even when the parametric dependence is discontinuous. Note that while error minima locations are successfully predicted, the maximum location(s) may be different. The bottom figure shows discontinuous Lagrange functions that are necessary for the RB solution to approximate the truth solution well.

4.2. The first 2-dimensional test case

As a first test case with 2-dimensional parameter, we consider the following equation whose solution space ends up being well approximated by a 40-dimensional RBM surrogate.

$$-u_{xx} - \mu_1 u_{yy} - \mu_2 u = -10 \sin(8x(y-1)) \quad \text{on } \Omega. \quad (4.5)$$

The physical domain is $\Omega = [-1, 1] \times [-1, 1]$ and we impose homogeneous Dirichlet boundary conditions on $\partial\Omega$. The truth approximation is a spectral Chebyshev collocation method with $\mathcal{N}_x = 50$ degrees of freedom in each direction. This means the truth approximation has dimension $\mathcal{N} = \mathcal{N}_x^2$. The parameter domain \mathcal{D} for (μ_1, μ_2) is taken to be $[0.1, 4] \times [0, 2]$. For the training set $\mathcal{E}_{\text{train}}$ we discretize \mathcal{D} using a tensorial 129×65 uniform Cartesian grid.

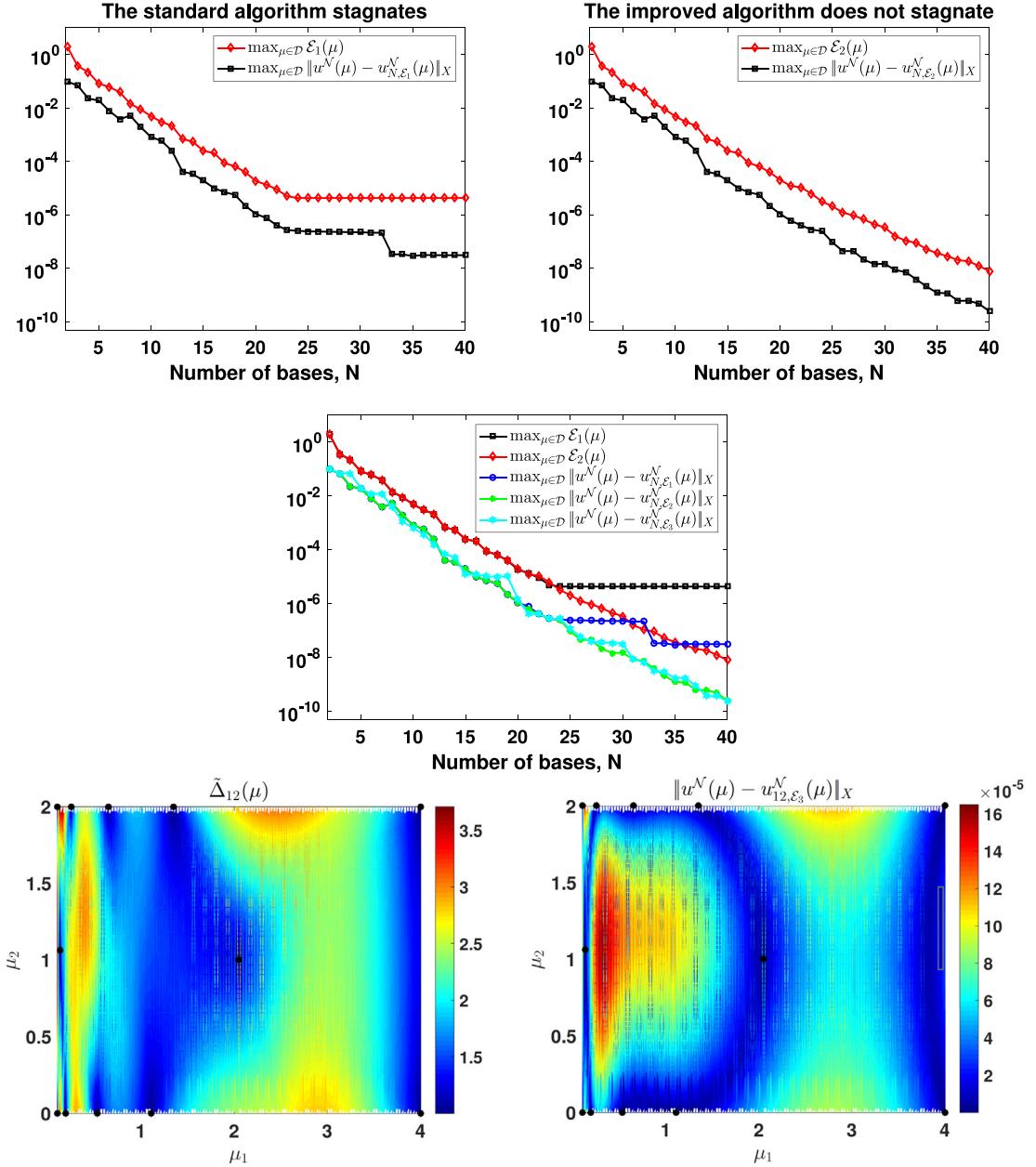


Fig. 5. History of convergence for the classical and new approaches for Eq. (4.5). The pictures on the bottom row indicate that residual-free error indicator roughly matches with the true error well. Note that while error minima locations are successfully predicted, the maximum location(s) may be different.

We present the results in Fig. 5, and observe the same behavior as in the one-dimensional case. Namely, the error estimate and the true error stagnates when the traditional error estimate is adopted. This stagnation is eliminated by the two newly proposed approaches. What is more, the error indicator for the residual-free approach tracks the true error across the parameter domain as well as the one-dimensional case. It thus comes as no surprise that the third approach, albeit without a certificate, produces RB solution as accurate as the second method.

We show in Fig. 6 the sets of snapshot locations S^N for all three methods. To differentiate them, we adopt the notation $S_{\mathcal{E}_i}^N$ for the sets produced by method i . We overlay the two sets $S_{\mathcal{E}_1}^{40}$ and $S_{\mathcal{E}_2}^{40}$ in the first picture. The larger the marker, the earlier it is picked by the greedy algorithm. Clearly, the two methods start by selecting the same points before deviating. The first method starts to clutter points in an unphysical manner and keeps doing so in the same neighborhood of the parameter domain. This leads to stagnation and potential singularity in the reduced solver. The enhanced approach ($S_{\mathcal{E}_2}^{40}$) avoids clustering and thus achieves better accuracy. The third, residual-free, approach demonstrates similar behavior.

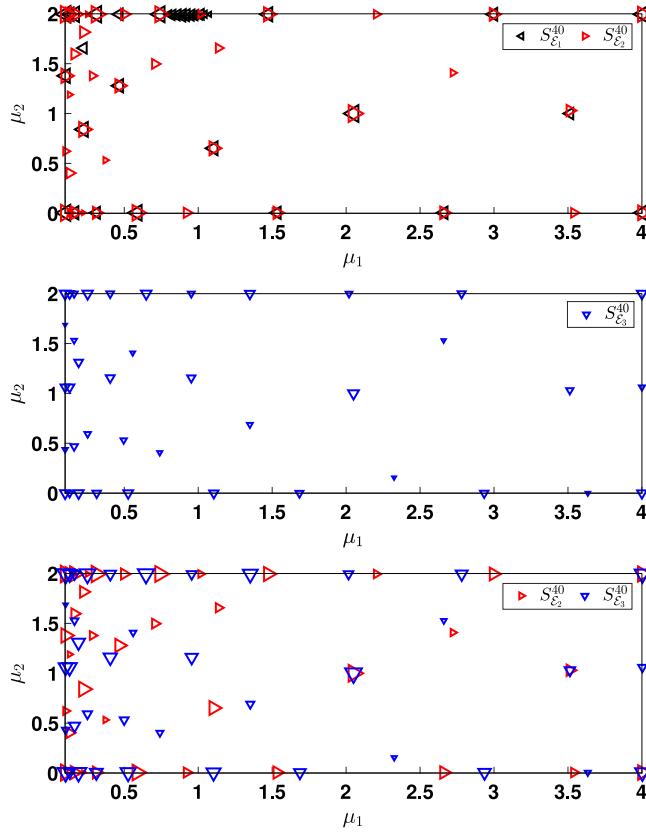


Fig. 6. Location of the chose snapshots for the classical approach and the two novel methods for the second 2-dimensional test case.

4.3. The second 2-dimensional test case

The second two-dimensional example has an equation that induces a solution manifold that requires many more snapshots to achieve small error:

$$(1 + \mu_1 x)u_{xx} + (1 + \mu_2 y)u_{yy} = e^{4xy} \quad \text{on } \Omega. \quad (4.6)$$

The parameter domain \mathcal{D} here is taken to be $[-0.99, 0.99]^2$. The physical domain Ω , boundary condition, truth solver and its resolution are all the same as the first two-dimensional case. We discretize \mathcal{D} using a tensorial 160×160 Cartesian grid with 160 equally-spaced points in each dimension.

The difficulty of this problem stems from the fact that the equation is close to degenerate at the four corners of the parameter domain. Thus the stability constant approaches zero toward the four corners, making accurate calculation of the residual norm even more critical. For example, the ratio (2.12) blows up if the denominator (the stability constant) converges to zero while the numerator stays at the root machine accuracy level. The results are shown in Fig. 7 confirming, again, all previously stated properties for the two novel approaches. The important role of an accurate residual norm calculation is apparent, as for example the chosen parameter values are unnecessarily more clustered toward the corners using the classical approach \mathcal{E}_1 , see top row of Fig. 7.

5. Concluding remarks

We have proposed two novel strategies for computing objective functions in the offline greedy algorithm in the reduced basis method. Our first strategy is residual-based, and follows standard practice in RBM by defining the objective to be an *a posteriori* upper bound for the error committed by a finite element method. This bound requires computation of a residual norm. In the standard RBM setting, this residual norm is computed in a way that can succumb to loss of significance when the magnitude of the norm reaches root machine precision. Our residual-based reformulation circumvents this premature stagnation without any loss in efficiency.

Our second strategy is residual-free, and uses only the RBM coefficients in the greedy objective. The particular function is the Lebesgue function from interpolation theory. We can provide a theoretical result guaranteeing that the parametric

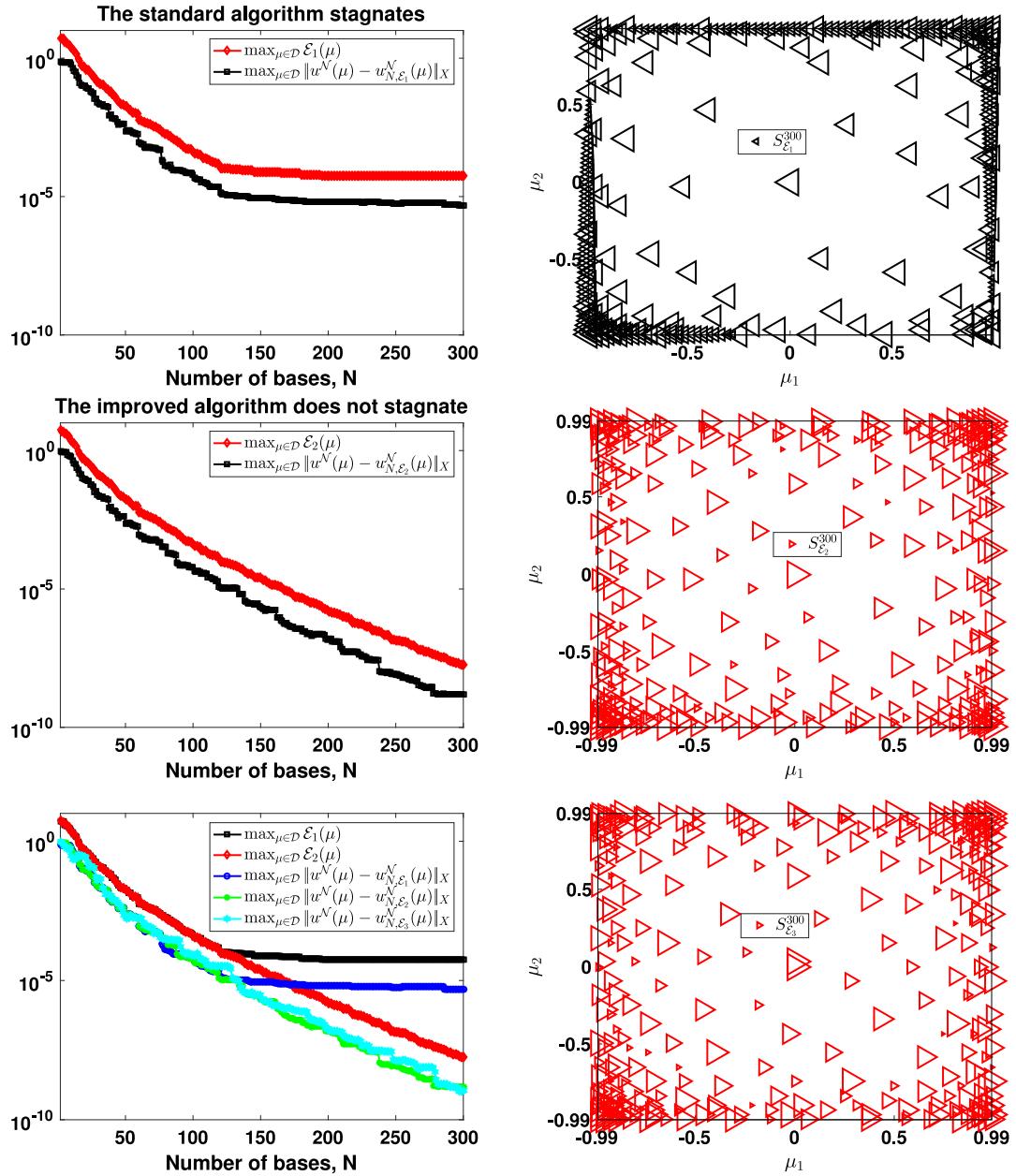


Fig. 7. The results for the second test problem.

behavior of this function provides an upper bound for the parametric variation of the error, and thus is a good objective function for a greedy search. However, the precise connection between the Lebesgue function and the error involves a parameter-independent multiplicative constant that is an uncomputable best approximation error. Therefore, the residual-free method can effectively choose parameter values, but it cannot provide error certification without a quantitative understanding of this best approximation error. Furthermore, we currently lack a result establishing that the residual-free objective is a lower bound for the true error; the establishment of such a result is a subject of ongoing work.

Our numerical experiments demonstrate that both of our strategies can effectively allow RBM to compute solutions to an accuracy much closer to machine precision than the classical reduced basis error estimator.

References

- [1] A. Pinkus, *n*-Widths in Approximation Theory, in: *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*, vol. 7, Springer-Verlag, Berlin, 1985.
- [2] T.A. Porsching, Estimation of the error in the reduced basis method solution of nonlinear equations, *Math. Comp.* 45 (172) (1985) 487–496.

- [3] A.K. Noor, Recent advances in reduction methods for nonlinear problems, *Comput. Struct.* 13 (1–3) (1981) 31–44.
- [4] T.A. Porsching, M.-Y.L. Lee, The reduced basis method for initial value problems, *SIAM J. Numer. Anal.* 24 (6) (1987) 1277–1287.
- [5] A. Quarteroni, G. Rozza, A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications, *J. Math. Ind.* 1 (2011) 3, 44.
- [6] P. Chen, A. Quarteroni, G. Rozza, A weighted reduced basis method for elliptic partial differential equations with random input data, *SIAM J. Numer. Anal.* 51 (6) (2013) 3163–3185.
- [7] J.S. Hesthaven, G. Rozza, B. Stamm, Certified reduced basis methods for parametrized partial differential equations, in: *SpringerBriefs in Mathematics*, in: *BCAM Basque Center for Applied Mathematics*, Springer, BCAM SpringerBriefs, Cham, Bilbao, 2016.
- [8] A. Quarteroni, A. Manzoni, F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, Vol. 92, Springer, 2015.
- [9] P. Benner, A. Cohen, M. Ohlberger, K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*, Vol. 15, SIAM, 2017.
- [10] F. Casenave, Accurate a posteriori error evaluation in the reduced basis method, *C. R. Math. Acad. Sci. Paris* 350 (9–10) (2012) 539–542.
- [11] F. Casenave, A. Ern, T. Lelièvre, Accurate and online-efficient evaluation of the a posteriori error bound in the reduced basis method, *ESAIM Math. Model. Numer. Anal.* 48 (1) (2014) 207–229.
- [12] A. Buhr, C. Engwer, M. Ohlberger, S. Rave, 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*, 2014, pp. 4094–4102.
- [13] C. Canuto, T. Tonn, K. Urban, A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear pdes, *SIAM J. Numer. Anal.* 47 (3) (2009) 2001–2022.
- [14] A. Janon, M. Nodet, C. Prieur, Goal-oriented error estimation for the reduced basis method, with application to sensitivity analysis, *J. Sci. Comput.* 68 (1) (2016) 21–41.
- [15] K.C. Hoang, P. Kerfriden, B.C. Khoo, S.P.A. Bordas, An efficient goal-oriented sampling strategy using reduced basis method for parametrized elastodynamic problems, *Numer. Methods Partial Differential Equations* 31 (2) (2015) 575–608.
- [16] J. Jiang, Y. Chen, A. Narayan, A goal-oriented reduced basis methods-accelerated generalized polynomial chaos algorithm, *SIAM/ASA J. Uncertain. Quantification* 4 (1) (2016) 1398–1420.
- [17] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics, *Arch. Comput. Methods Eng.* 15 (3) (2008) 229–275.
- [18] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Math.* 339 (9) (2004) 667–672.
- [19] D. Huynh, G. Rozza, S. Sen, A. Patera, A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants, *C. R. Acad. Sci. Paris, Sér. I* 345 (2007) 473–478.
- [20] D. Huynh, D. Knezevic, Y. Chen, J. Hesthaven, A. Patera, A natural-norm successive constraint method for inf-sup lower bounds, *Comput. Methods Appl. Mech. Engrg.* 199 (2010) 1963–1975.
- [21] Y. Chen, A certified natural-norm successive constraint method for parametric inf-sup lower bounds, *Appl. Numer. Math.* 99 (2016) 98–108.
- [22] M. Yano, A space-time petrov-galerkin certified reduced basis method: Application to the boussinesq equations, *SIAM J. Sci. Comput.* 36 (1) (2014) A232–A266.
- [23] A. Manzoni, F. Negri, Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized pdes, *Adv. Comput. Math.* 41 (5) (2015) 1255–1288.