AN "hp" CERTIFIED REDUCED BASIS METHOD FOR PARAMETRIZED ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS[*]

JENS L. EFTANG[†], ANTHONY T. PATERA[‡], AND EINAR M. RØNQUIST[§]

Abstract. We present a new "*hp*" parameter multi-domain certified reduced basis method for rapid and reliable online evaluation of functional outputs associated with parametrized elliptic partial differential equations. We propose, and provide theoretical justification for, a new procedure for adaptive partition ("*h*"-refinement) of the parameter domain into smaller parameter subdomains: we pursue a hierarchical splitting of the parameter (sub)domains based on proximity to judiciously chosen parameter *anchor points* within each subdomain. Subsequently, we construct individual standard RB approximation spaces ("*p*"-refinement) over each subdomain. Greedy parameter sampling procedures and *a posteriori* error estimation play important roles in both the "*h*"-type and "*p*"-type stages of the new algorithm. We present illustrative numerical results for a convection-diffusion problem: the new "*hp*"-approach is considerably faster (respectively, more costly) than the standard "*p*"-type reduced basis method in the online (respectively, offline) stage.

Key words. reduced basis; *a posteriori* error estimation; Greedy sampling; *h*-type; *p*-type; *hp* convergence; parameter domain decomposition

AMS subject classifications. 35J25, 65M12, 65N15, 65N15, 65N30

1. Introduction.

The certified reduced basis (RB) method provides a computational framework for rapid and reliable evaluation of functional outputs associated with parametrized partial differential equations. Given any *input* parameter vector—such as geometric factors or material property coefficients—the RB field approximation is constructed as a Galerkin linear combination of pre-computed "truth" finite element (FE) "snapshots" at optimally chosen parameter values; the RB output approximation is then evaluated as a functional of the RB field approximation. The methodology is originally introduced in [1, 21] and then further analyzed in [22, 23]; for a review of both earlier and more recent contributions, see [24].

For problems in which the field variable varies smoothly with the parameters, good RB approximations can be obtained with very few snapshots: the RB approximation converges exponentially fast [5, 8]. Furthermore, rigorous *a posteriori* upper bounds for the error in the RB approximation (with respect to the truth discretization) can be readily developed [24]. Finally, under an assumption on "affine" parameter dependence (perhaps only approximate [4, 13]), both the RB output approximation *and* the associated RB output error bound can be computed very efficiently by offline-online computational procedures [24]. The RB method is especially attractive in engineering contexts in which low marginal (online) computational cost is advantageous: "real-time"—such as parameter estimation [19] and optimal control—and "many-query"—such as multiscale [6] or stochastic simulation [7].

The RB approximation space is specifically constructed to provide accurate approximations for any parameter value in a predefined parameter domain. Hence, larger parameter domains typically induce larger RB spaces and greater computational cost.

[*] This work has been supported by the Norwegian University of Science and Technology, AFOSR Grant number FA 9550-07-1-0425, and OSD/AFOSR Grant number FA 9550-09-1-0613.

[†] Department of Mathematical Sciences, NTNU (*eftang@math.ntnu.no*)

[‡] Department of Mechanical Engineering, MIT (*patera@mit.edu*)

[§] Department of Mathematical Sciences, NTNU (*ronquist@math.ntnu.no*)

Accepted for publication in SIAM Journal on Scientific Computing, July 2010.

In this paper we propose, and provide theoretical justification for, a new procedure for adaptive partition ("h"-refinement) of the parameter domain into smaller parameter subdomains: we pursue a hierarchical splitting of the parameter (sub)domains based on proximity to judiciously chosen parameter *anchor points* within each subdomain. Subsequently, we construct individual standard RB approximation spaces ("p"-refinement) over each subdomain. Greedy sampling procedures and rigorous *a posteriori* error estimation play important roles in both the "h"-type and "p"-type stages of the algorithm.

In this new approach, the RB approximation associated with any new parameter value is, as always, constructed as a linear Galerkin combination of snapshots from the parameter (sub)domain in which this new parameter value resides. However, we expect the online computational cost of the new approach to be greatly reduced relative to the online cost of the standard RB approach due to the smaller parameter (sub)domains and hence lower-dimensional local RB approximation spaces associated with the "hp" approximation. The method should be particularly effective for problems in which the solution has very different structure in different regions of the parameter domain—problems for which a snapshot from one parameter region may be of limited value for the RB approximation in another parameter region.

The notion of parameter domain refinement within the model order reduction framework is considered in several earlier works. In [2, 3], a reduced-order parameter multi-element "interpolation" procedure is introduced for aeroelasticity problems; this interpolation procedure and our approach here share a similar error-adaptive domain-decomposition foundation. However, the two approaches are quite different in conception: interpolation on a manifold rather than Galerkin projection (here); parameter domain partition based on a Voronoi diagram rather than a hierarchical tree structure decomposition (here); heuristic error indicators rather than rigorous error bounds (here); and less strict rather than strict offline-online segregation (here). However, our own approach cannot yet treat problems of the complexity considered in [2, 3].

In other related work [14, 15, 25], adaptive *train sample* refinement is considered to render the Greedy parameter sampling procedure more efficient: richer samples are consided only as needed in the Greedy iterations [25] and only where needed in the parameter domain [14, 15]. Our approach invokes a similar technique: we include new points in the train sample within each subdomain at each new level of "h"-refinement; we thus effectively adapt the train sample to the more "difficult" parameter regions which require many subdomains. Reference [14] also proposes a multiple-bases ("hp") approach which shares certain features with our approach here, but also differs in several important ways in particular related to the "h"-refinement partition strategy.

In §2 we give the general problem statement along with various definitions required throughout the paper. In §3 we review the standard ("p"-type) RB method; in §4 we present the new "h"-type RB method and provide an *a priori* convergence theory for a zeroth-order approximation in the one-parameter case; in §5 we present the new "hp"-type RB method as a combination of the "p"- and "h"-type methods. In §6 we present numerical results for a convection-diffusion model problem and in particular we compare the computational cost of the new "hp"-approach to the standard "p"-type method. We conclude in §7 with some final remarks.

2. Problem Statement.

We shall consider linear, elliptic, coercive, second-order partial differential equations. We denote the physical domain by $\Omega \subset \mathbb{R}^2$, and we introduce the spaces $L^2(\Omega) = \{v : \int_\Omega v^2 \,\mathrm{d}\Omega < \infty\}$, $H^1(\Omega) = \{v \in L^2(\Omega) : |\nabla v| \in$

$L^2(\Omega)\}$, and $H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$. We further define the space associated with the exact solution (hence ^{e}) $X^e \equiv X^e(\Omega)$ such that $H_0^1(\Omega) \subseteq X^e(\Omega) \subset H^1(\Omega)$. We introduce a compact parameter domain $\mathcal{D} \subset \mathbb{R}^P$; a point in $\mathcal{D}$ shall be denoted $\boldsymbol{\mu} = (\mu_1, \dots, \mu_P)$.

For each $\boldsymbol{\mu} \in \mathcal{D}$, $a(\cdot,\cdot;\boldsymbol{\mu})$ is an X^e–coercive and X^e–continuous bilinear form and $f(\cdot;\boldsymbol{\mu})$ is an X^e–bounded linear functional. To accomodate an efficient offline–online computational procedure, we assume that a and f admit affine expansions as

$$a(\cdot,\cdot;\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} a^q(\cdot,\cdot)\Theta_a^q(\boldsymbol{\mu}), \qquad f(\cdot;\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} f^q(\cdot)\Theta_f^q(\boldsymbol{\mu}), \tag{2.1}$$

for modest Q_a and Q_f, where the a^q and f^q are $\boldsymbol{\mu}$-independent continuous bilinear forms and linear functionals, respectively, and the Θ_a^q and Θ_f^q are $\boldsymbol{\mu}$-dependent continuous functions. (The assumption (2.1) can be relaxed with the *empirical interpolation* method [4, 13] for the construction of good affine *approximations* to a and f.) For simplicity, we introduce $Q = \max\{Q_a, Q_f\}$.

The exact problem statement reads: Given any $\boldsymbol{\mu} \in \mathcal{D}$, find $u^e(\boldsymbol{\mu}) \in X^e$ such that

$$a(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v;\boldsymbol{\mu}), \quad \forall v \in X^e. \tag{2.2}$$

The output of interest can then be evaluated as a functional of the field variable, say $s(\boldsymbol{\mu}) = l(u^e(\boldsymbol{\mu});\boldsymbol{\mu})$ for some X^e–bounded linear functional $l(\cdot;\boldsymbol{\mu})$. In this paper, however, for simplicity of exposition, we consider no particular output(s) of interest; our “*hp*” procedure does not depend on the output functional(s) chosen.

We next introduce a “truth” finite element (FE) space $X \equiv X^{\mathcal{N}}(\Omega) \subset X^e(\Omega)$ of finite dimension $\mathcal{N}$. The truth discretization of (2.2) reads: For any $\boldsymbol{\mu} \in \mathcal{D}$, find $u(\boldsymbol{\mu}) \in X$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v;\boldsymbol{\mu}), \quad \forall v \in X. \tag{2.3}$$

We assume that X is rich enough that the error between the truth and exact solutions is in practice negligible. The RB approximation will be built upon truth snapshots $u(\boldsymbol{\mu}_n) \approx u^e(\boldsymbol{\mu}_n)$, $1 \le n \le N$, for judiciously chosen $\boldsymbol{\mu}_1 \in \mathcal{D}, \dots, \boldsymbol{\mu}_N \in \mathcal{D}$, and the RB error shall be measured with respect to the truth FE approximation.

For any $\boldsymbol{\mu} \in \mathcal{D}$, let $a_s(\cdot,\cdot;\boldsymbol{\mu})$ denote the symmetric part of $a(\cdot,\cdot;\boldsymbol{\mu})$—for all $v, w \in X$, $a_s(w,v;\boldsymbol{\mu}) = \frac{1}{2}(a(w,v;\boldsymbol{\mu}) + a(v,w;\boldsymbol{\mu}))$; further, let $\bar{\boldsymbol{\mu}} \in \mathcal{D}$ denote a fixed *reference parameter*. We then define the parameter-independent X-inner product and corresponding X-norm as

$$(\cdot,\cdot)_X \equiv a_s(\cdot,\cdot;\bar{\boldsymbol{\mu}}), \qquad \|\cdot\|_X = \sqrt{(\cdot,\cdot)_X}, \tag{2.4}$$

respectively. By our assumptions, $\|\cdot\|_X$ is equivalent to the H^1-norm.

Finally, we introduce for all $\boldsymbol{\mu} \in \mathcal{D}$ the coercivity and continuity constants of $a(\cdot,\cdot;\boldsymbol{\mu})$ with respect to the X-norm,

$$\alpha(\boldsymbol{\mu}) \equiv \inf_{w \in X} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_X^2}, \qquad \gamma(\boldsymbol{\mu}) \equiv \sup_{v \in X} \sup_{w \in X} \frac{a(v,w;\boldsymbol{\mu})}{\|v\|_X \|w\|_X}, \tag{2.5}$$

respectively. For any particular $\boldsymbol{\mu} \in \mathcal{D}$, we further require lower and upper bounds,

$$0 < \alpha_{\mathrm{LB}}(\boldsymbol{\mu}) \le \alpha(\boldsymbol{\mu}), \tag{2.6}$$

$$\infty > \gamma_{\mathrm{UB}}(\boldsymbol{\mu}) \ge \gamma(\boldsymbol{\mu}), \tag{2.7}$$

which shall play a role in our computational procedures. We shall also invoke lower and upper bounds over $\mathcal{D}$,

$$\underline{\alpha} = \min_{\boldsymbol{\mu} \in \mathcal{D}} \alpha(\boldsymbol{\mu}), \tag{2.8}$$

$$\overline{\gamma} = \max_{\boldsymbol{\mu} \in \mathcal{D}} \gamma(\boldsymbol{\mu}), \tag{2.9}$$

for the purposes of our theoretical arguments.

We shall later need the following lemma,

LEMMA 2.1. *Let $\Theta_a^q : \mathcal{D} \to \mathbb{R}$, $1 \le q \le Q_a$, $\Theta_f^q : \mathcal{D} \to \mathbb{R}$, $1 \le q \le Q_f$, satisfy Lipschitz conditions*

$$|\Theta_a^q(\boldsymbol{\mu}_1) - \Theta_a^q(\boldsymbol{\mu}_2)| \le C_a |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|, \quad \forall \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathcal{D}, \quad 1 \le q \le Q_a, \tag{2.10}$$

$$|\Theta_f^q(\boldsymbol{\mu}_1) - \Theta_f^q(\boldsymbol{\mu}_2)| \le C_f |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|, \quad \forall \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathcal{D}, \quad 1 \le q \le Q_f. \tag{2.11}$$

Then, given any $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathcal{D}$, there exists a positive constant $\tilde{C}$ such that

$$\|u(\boldsymbol{\mu}_1) - u(\boldsymbol{\mu}_2)\|_X \le \tilde{C} |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|. \tag{2.12}$$

Proof. We have

$$a(u(\boldsymbol{\mu}_1), v; \boldsymbol{\mu}_1) = f(v; \boldsymbol{\mu}_1), \quad \forall v \in X, \tag{2.13}$$

$$a(u(\boldsymbol{\mu}_2), v; \boldsymbol{\mu}_2) = f(v; \boldsymbol{\mu}_2), \quad \forall v \in X. \tag{2.14}$$

By bilinearity of a, we thus have for all $v \in X$,

$$\begin{aligned} a(u(\boldsymbol{\mu}_1) - u(\boldsymbol{\mu}_2), v; \boldsymbol{\mu}_1) = f(v; \boldsymbol{\mu}_1) &- f(v; \boldsymbol{\mu}_2) \\ &+ a(u(\boldsymbol{\mu}_2), v; \boldsymbol{\mu}_2) - a(u(\boldsymbol{\mu}_2), v; \boldsymbol{\mu}_1). \end{aligned} \tag{2.15}$$

We first examine the right-hand side of (2.15).

By the triangle inequality and the affine expansions (2.1) for a and f, we have for all $w, v \in X$ and any $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathcal{D}$,

$$|a(w, v; \boldsymbol{\mu}_1) - a(w, v; \boldsymbol{\mu}_2)| \le \sum_{q=1}^{Q_a} |a^q(w, v)\big(\Theta_a^q(\boldsymbol{\mu}_1) - \Theta_a^q(\boldsymbol{\mu}_2)\big)|, \tag{2.16}$$

and

$$|f(v; \boldsymbol{\mu}_1) - f(v; \boldsymbol{\mu}_2)| \le \sum_{q=1}^{Q_f} |f^q(v)\big(\Theta_f^q(\boldsymbol{\mu}_1) - \Theta_f^q(\boldsymbol{\mu}_2)\big)|, \tag{2.17}$$

respectively. By our hypothesis (2.10) and (2.11) on Θ_a^q, $1 \le q \le Q_a$ and Θ_f^q, $1 \le q \le Q_f$, respectively, and continuity of a^q, $1 \le q \le Q_a$, and f^q, $1 \le q \le Q_f$, there exist constants $\tilde{c}_1$ and $\tilde{c}_2$ (independent of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$) such that

$$|a(w, v; \boldsymbol{\mu}_1) - a(w, v; \boldsymbol{\mu}_2)| \le \tilde{c}_1 \|v\|_X \|w\|_X |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|, \tag{2.18}$$

and

$$|f(v; \boldsymbol{\mu}_1) - f(v; \boldsymbol{\mu}_2)| \le \tilde{c}_2 \|v\|_X |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|. \tag{2.19}$$

Recall that Q_a and Q_f are fixed and finite.

We now let $v = u(\boldsymbol{\mu}_1) - u(\boldsymbol{\mu}_2)$ in (2.15) and deduce from the triangle inequality, (2.18), and (2.19) that

$$\begin{aligned} a(u(\boldsymbol{\mu}_1) - u(\boldsymbol{\mu}_2), u(\boldsymbol{\mu}_1) - u(\boldsymbol{\mu}_2); \boldsymbol{\mu}_1)& \\ \leq \big(\tilde{c}_1 \|u(\boldsymbol{\mu}_2)\|_X + \tilde{c}_2\big)\|u(\boldsymbol{\mu}_1) - u(\boldsymbol{\mu}_2)\|_X |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|.& \qquad (2.20) \end{aligned}$$

By coercivity and the bound (2.8), we get

$$\|u(\boldsymbol{\mu}_1) - u(\boldsymbol{\mu}_2)\|_X \leq \frac{1}{\underline{\alpha}}\big(\tilde{c}_1 \|u(\boldsymbol{\mu}_2)\|_X + \tilde{c}_2\big)|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1|. \qquad (2.21)$$

Finally, by the Lax-Milgram Lemma, (2.8), boundedness of $f(\cdot;\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in \mathcal{D}$, and the fact that $\mathcal{D} \subset \mathbb{R}^P$ is closed,

$$\|u(\boldsymbol{\mu}_2)\|_X \leq \frac{\|f(\cdot;\boldsymbol{\mu}_2)\|_{X'}}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})} \leq \max_{\boldsymbol{\mu}\in\mathcal{D}} \frac{\|f(\cdot;\boldsymbol{\mu})\|_{X'}}{\underline{\alpha}}, \qquad (2.22)$$

(here X' denotes the dual space of X) and we thus obtain the desired result with

$$\tilde{C} = \max_{\boldsymbol{\mu}\in\mathcal{D}} \frac{\tilde{c}_1 \|f(\cdot;\boldsymbol{\mu})\|_{X'} + \tilde{c}_2 \underline{\alpha}}{\underline{\alpha}^2}. \qquad (2.23)$$

(We can develop a constant $\tilde{C}$ that is furthermore independent of $\mathcal{N}$ by replacing the truth entities $\underline{\alpha}$ and $\|f(\cdot;\boldsymbol{\mu})\|_{X'}$ in (2.23) by the corresponding exact entities.) □

3. The “*p*”-type Reduced Basis Method.

In the standard RB approach, a single approximation space is enriched with new basis functions until the space is considered sufficiently rich; we shall refer to this approach as the “*p*”-type RB method. The new “*h*”-type and “*hp*”-type methods will borrow and adapt several of the ingredients from the standard approach: *a posteriori* error estimation; greedy parameter sampling; and offline–online computational decoupling of the RB discretization and the truth FE discretization through a construction–evaluation decomposition. Below, we summarize the standard RB approximation with particular emphasis on these key ingredients.

3.1. Approximation.

The RB approximation space $X_N \equiv X_N(\Omega) \subset X^{\mathcal{N}}(\Omega)$ is defined in terms of a set of parameter vectors $\boldsymbol{\mu}_1 \in \mathcal{D}, \dots, \boldsymbol{\mu}_N \in \mathcal{D}$ as

$$X_N = \mathrm{span}\{u(\boldsymbol{\mu}_1), \dots, u(\boldsymbol{\mu}_N)\}. \qquad (3.1)$$

(Note that in practice, an $(\cdot,\cdot)_X$–orthonormal basis for X_N is constructed by a Gram-Schmidt procedure.) The RB approximation reads: Given any $\boldsymbol{\mu} \in \mathcal{D}$, find $u_N(\boldsymbol{\mu}) \in X_N$ such that

$$a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v;\boldsymbol{\mu}), \quad \forall v \in X_N. \qquad (3.2)$$

Under the assumption that $u(\boldsymbol{\mu})$ depends smoothly on the parameters, we expect that N—the dimension of the RB space—can be chosen much smaller than $\mathcal{N}$—the dimension of the truth space X—for comparable numerical accuracy.

We finally *formally* define the “order” p of the RB approximation as $p \equiv N^{1/P} - 1$. We shall return to this point and elaborate on this interpretation in Remark 5.

3.2. *A Posteriori* **Error Estimation.** We recall here the derivation of an *a posteriori* X-norm bound for the error in the RB field approximation relative to the corresponding truth approximation [24].

Given any $\boldsymbol{\mu} \in \mathcal{D}$, we obtain the RB approximation, $u_N(\boldsymbol{\mu})$, from (3.2); we then define for all $v \in X$ the RB residual as

$$r_N(v;\boldsymbol{\mu}) \equiv f(v;\boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}), v;\boldsymbol{\mu}). \tag{3.3}$$

The *Riesz representation* of the residual, $\mathcal{R}_N(\boldsymbol{\mu}) \in X$, satisfies

$$(\mathcal{R}_N(\boldsymbol{\mu}), v)_X = r_N(v;\boldsymbol{\mu}), \quad \forall v \in X. \tag{3.4}$$

We can now state

LEMMA 3.1 (*A Posteriori* X-norm Error Bound). *For any $\boldsymbol{\mu} \in \mathcal{D}$, the RB error bound*

$$\Delta_N(\boldsymbol{\mu}) \equiv \frac{\|\mathcal{R}_N(\boldsymbol{\mu})\|_X}{\alpha_{\rm LB}(\boldsymbol{\mu})}, \tag{3.5}$$

satisfies

$$\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X \leq \Delta_N(\boldsymbol{\mu}), \tag{3.6}$$

$$\frac{\Delta_N(\boldsymbol{\mu})}{\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X} \leq \frac{\gamma_{\rm UB}(\boldsymbol{\mu})}{\alpha_{\rm LB}(\boldsymbol{\mu})}, \tag{3.7}$$

for $\alpha_{\rm LB}(\boldsymbol{\mu})$ and $\gamma_{\rm UB}(\boldsymbol{\mu})$ given by (2.6) *and* (2.7)*, respectively.*

Proof. The RB error, $e_N(\boldsymbol{\mu}) = u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$, satisfies the error-residual equation

$$a(e_N(\boldsymbol{\mu}), v;\boldsymbol{\mu}) = r_N(v;\boldsymbol{\mu}), \quad \forall v \in X. \tag{3.8}$$

To obtain (3.6), we choose $e_N(\boldsymbol{\mu})$ for v in (3.8) and invoke (3.4) and the Cauchy-Schwarz inequality to get

$$a(e_N(\boldsymbol{\mu}), e_N(\boldsymbol{\mu});\boldsymbol{\mu}) = (\mathcal{R}_N(\boldsymbol{\mu}), e_N(\boldsymbol{\mu}))_X \leq \|\mathcal{R}_N(\boldsymbol{\mu})\|_X \|e_N(\boldsymbol{\mu})\|_X; \tag{3.9}$$

we then invoke coercivity and (2.6) to arrive at

$$\alpha_{\rm LB}(\boldsymbol{\mu}) \|e_N(\boldsymbol{\mu})\|_X^2 \leq \|\mathcal{R}_N(\boldsymbol{\mu})\|_X \|e_N(\boldsymbol{\mu})\|_X. \tag{3.10}$$

The result (3.6) now directly follows from the definition (3.5).

To obtain (3.7), we choose $\mathcal{R}_N(\boldsymbol{\mu})$ for v in (3.8) and invoke (3.4), continuity, and (2.7) to get

$$\|\mathcal{R}_N(\boldsymbol{\mu})\|_X^2 = a(e_N(\boldsymbol{\mu}), \mathcal{R}_N(\boldsymbol{\mu});\boldsymbol{\mu}) \leq \gamma_{\rm UB}(\boldsymbol{\mu}) \|e_N(\boldsymbol{\mu})\|_X \|\mathcal{R}_N(\boldsymbol{\mu})\|_X; \tag{3.11}$$

hence $\|\mathcal{R}_N(\boldsymbol{\mu})\|_X / \|e_N(\boldsymbol{\mu})\|_X \leq \gamma_{\rm UB}(\boldsymbol{\mu})$ and the result (3.7) follows from the definition (3.5). □

3.3. Construction–Evaluation Decomposition. Thanks to the assumption (2.1) on affine parameter dependence, the computational procedures for the RB solution and error bound admit *construction-evaluation* decompositions (see also [18, 20]): the construction stage is computationally expensive—the operation count depends on $\mathcal{N}$—however in the subsequent evaluation stage we can then rapidly—independently of $\mathcal{N}$—evaluate the RB approximation and RB error bound for any $\boldsymbol{\mu} \in \mathcal{D}$. (In actual practice we would of course also evaluate the RB output and RB output error bound—at negligible additional cost.) The construction-evaluation decomposition in turn permits the full offline–online computational decoupling described in the Introduction; we further discuss this decoupling below.

We first describe the construction-evaluation decomposition for the RB approximation: Let $\{\zeta_1 \in X_N, \dots, \zeta_N \in X_N\}$ denote an X-orthonormal basis for X_N. In the construction stage, we assemble the matrices $A_N^q \in \mathbb{R}^{N\times N}$, $1 \le q \le Q_a$, and the vectors $F_N^q \in \mathbb{R}^N$, $1 \le q \le Q_f$, the elements of which are defined by

$$A^q_{N,mn} \equiv a^q(\zeta_n, \zeta_m), \qquad F^q_{N,m} \equiv f^q(\zeta_m), \qquad 1 \le m, n \le N, \tag{3.12}$$

respectively. In the evaluation stage—given any $\boldsymbol{\mu} \in \mathcal{D}$—we evaluate the functions $\Theta_a^q(\boldsymbol{\mu})$, $1 \le q \le Q_a$, and $\Theta_f^q(\boldsymbol{\mu})$, $1 \le q \le Q_f$, in $\mathcal{O}(Q)$ operations; we then construct the RB stiffness matrix and load vector as

$$A_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) A_N^q, \qquad F_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) F_N^q, \tag{3.13}$$

respectively, in $\mathcal{O}(Q_a N^2 + Q_f N) = \mathcal{O}(QN^2)$ operations; finally, we solve the associated system of equations

$$A_N(\boldsymbol{\mu}) \underline{u}_N(\boldsymbol{\mu}) = F_N(\boldsymbol{\mu}) \tag{3.14}$$

for the RB basis coefficients $\underline{u}_N(\boldsymbol{\mu}) \equiv [u_{N,1}(\boldsymbol{\mu}), \dots, u_{N,N}(\boldsymbol{\mu})]^{\mathrm{T}}$ in $\mathcal{O}(N^3)$ operations (we must anticipate that $A_N(\boldsymbol{\mu})$ is dense).

We next describe the construction-evaluation decomposition for the dual norm of the residual. By linearity, we can write (3.4) as

$$\begin{aligned}(\mathcal{R}_N(\boldsymbol{\mu}), v)_X &= \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) f^q(v) - \sum_{q=1}^{Q_a} \sum_{n=1}^{N} \Theta_a^q(\boldsymbol{\mu}) u_{N,n}(\boldsymbol{\mu}) a^q(\zeta_n, v) &(3.15)\\ &\equiv \sum_{n=1}^{\hat{N}} \Gamma_n(\boldsymbol{\mu}) \mathcal{L}_n(v), &(3.16)\end{aligned}$$

where $\hat{N} = Q_f + N Q_a$. By linear superposition, we can thus write

$$\mathcal{R}_N(\boldsymbol{\mu}) = \sum_{n=1}^{\hat{N}} \Gamma_n(\boldsymbol{\mu}) \mathcal{G}_n, \tag{3.17}$$

where, for $1 \le n \le \hat{N}$,

$$(\mathcal{G}_n, v)_X = \mathcal{L}_n(v), \quad \forall v \in X. \tag{3.18}$$

We thus have

$$\begin{aligned}\|\mathcal{R}_N(\boldsymbol{\mu})\|_X^2 &= (\mathcal{R}_N(\boldsymbol{\mu}), \mathcal{R}_N(\boldsymbol{\mu}))_X &(3.19)\\ &= \sum_{m=1}^{\hat{N}} \sum_{n=1}^{\hat{N}} \Gamma_m(\boldsymbol{\mu}) \Gamma_n(\boldsymbol{\mu}) G_{mn}, &(3.20)\end{aligned}$$

where the G_{mn} are defined as

$$G_{mn} \equiv (\mathcal{G}_m, \mathcal{G}_n)_X, \quad 1 \le m, n \le \hat{N}. \tag{3.21}$$

In the construction stage we first perform the truth Poisson solves (3.18) for $\mathcal{G}_n$, $1 \le n \le \hat{N}$; we then compute and store the inner products G_{mn}, $1 \le m, n \le \hat{N}$. In the evaluation stage, we evaluate the functions $\Gamma_n(\boldsymbol{\mu})$, $1 \le n \le \hat{N}$, in $\mathcal{O}(NQ_a + Q_f) = \mathcal{O}(NQ)$ operations and then perform the summation (3.19) in $\mathcal{O}((NQ_a + Q_f)^2) = \mathcal{O}(N^2Q^2)$ operations.

In general, the coercivity lower bound $\alpha_{\rm LB}(\boldsymbol{\mu})$ will not be known analytically and must be computed. An efficient construction-evaluation decomposition for the coervivity lower bound—the *successive constraint method*—can be found in [17, 24]; the evaluation complexity is independent of $\mathcal{N}$. We do not discuss this component further here in particular because for our numerical example of §6 an analytical lower bound $\alpha_{\rm LB}(\boldsymbol{\mu})$ is in fact available.

Algorithm 1: Greedy^1(Ξ, $\boldsymbol{\mu}_1$, $\epsilon_{\rm tol}$, $\tilde{N}_{\max}$)

```
initialize: $N \leftarrow 0$, $\epsilon_0 \leftarrow \infty$, $X_0 \leftarrow \{0\}$
while $\epsilon_N > \epsilon_{\rm tol}$ and $N < \tilde{N}_{\max}$ do
    $N \leftarrow N + 1$
    $X_N \leftarrow X_{N-1} \oplus \mathrm{span}\{u(\boldsymbol{\mu}_N)\}$
    $\epsilon_N \leftarrow \max_{\boldsymbol{\mu} \in \Xi} \Delta_N(\boldsymbol{\mu})$
    $\boldsymbol{\mu}_{N+1} \leftarrow \arg\max_{\boldsymbol{\mu} \in \Xi} \Delta_N(\boldsymbol{\mu})$
end
$N_{\max} \leftarrow N$
```

3.4. Greedy Parameter Sampling. We now discuss the construction of the hierarchical RB approximation spaces $X_N = \mathrm{span}\{u(\boldsymbol{\mu}_n)\}_{n=1}^N \subset X$, $1 \le N \le N_{\max}$ (see also [24, 26]). We first introduce a finite *train sample* $\Xi \subset \mathcal{D}$; a (random, say) initial parameter vector $\boldsymbol{\mu}_1 \in \mathcal{D}$; an error tolerance $\epsilon_{\rm tol}$; and a maximum RB dimension $\tilde{N}_{\max}$. We then perform Algorithm 1. The output of the algorithm is the RB space $X_{N_{\max}}$, where $N_{\max} \le \tilde{N}_{\max}$. Note that the construction-evaluation decomposition allows us to use a dense train sample: each evaluation of the error bound in the max is very inexpensive; the truth is invoked only for the "winning" candidates, $\boldsymbol{\mu}_N$, $1 \le N \le N_{\max}$.

3.5. Offline-Online Computational Decoupling. We now describe the full offline-online decoupling procedure for the "p"-type RB approximation: the offline stage—performed only once as pre-processing—may be very expensive ($\mathcal{N}$-dependent), however the subsequent ($\mathcal{N}$-independent) online stage—performed many times for the computation of the RB solution (and output) and RB error bound (and output error bound)—is very fast.

The offline stage is essentially the Greedy^1 algorithm (Algorithm 1). The parameter-independent entities $A^q_{N_{\max}} \in \mathbb{R}^{N_{\max} \times N_{\max}}$, $1 \le q \le Q_a$, $F^q_{N_{\max}} \in \mathbb{R}^{N_{\max}}$, $1 \le q \le Q_f$ and $(\mathcal{G}_m, \mathcal{G}_n)_X$, $1 \le m, n \le Q_a N_{\max} + Q_f$ are retained from the construction stage of the last iteration. The permanent *online* storage requirement is thus $\mathcal{O}(Q_a N^2_{\max} + Q_f N_{\max}) = \mathcal{O}(QN^2_{\max})$ for the $A^q_{N_{\max}}$ and $F^q_{N_{\max}}$, and $\mathcal{O}((Q_a N_{\max} + Q_f)^2) = \mathcal{O}(Q^2 N^2_{\max})$ for the $(\mathcal{G}_m, \mathcal{G}_n)_X$. We note that since the RB spaces are nested,

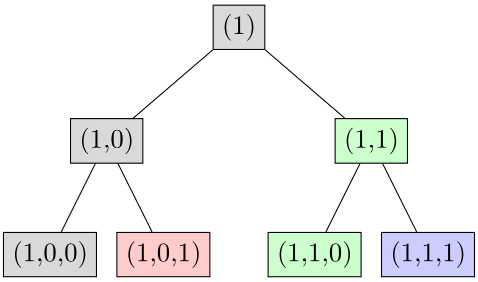

FIGURE 4.1. *A perfect binary tree and associated Boolean vectors corresponding to $L = 3$.*

we can extract subarrays from the stored entities in order to construct RB approximations of any order $1 \le N \le N_{\max}$ (hence providing for online adaptivity).

The online stage is, for the "p"-type method, equivalent to the evaluation stage: given any $\boldsymbol{\mu} \in \mathcal{D}$, we assemble the RB system in $\mathcal{O}(Q_a N^2 + Q_f N) = \mathcal{O}(QN^2)$ operations, compute the RB solution in $\mathcal{O}(N^3)$ operations, and finally evaluate the RB error bound in $\mathcal{O}((NQ_a + Q_f)^2) = \mathcal{O}(N^2Q^2)$ operations.

4. The "h"-type Reduced Basis Method.

In this section we formulate the "h"-type reduced basis method. We first provide preliminaries required throughout the paper; we next present the "h"-type approximation algorithm; we then consider *a posteriori* error estimation; we subsequently describe the offline-online computational decomposition; finally, we develop a new *a priori* convergence theory for the "zeroth-order" approximation in the case of one parameter.

4.1. Preliminaries: Tree-Subdomain Structure.

We first define the set of Boolean vectors of length l,

$$\mathcal{B}_l \equiv \{1\} \times \{0,1\}^{l-1}; \tag{4.1}$$

we shall denote a particular member of $\mathcal{B}_l$ as

$$B_l = (1, i_2 \ldots, i_l) \in \mathcal{B}_l. \tag{4.2}$$

We next introduce a perfect binary tree with L levels and $K = 2^{L-1}$ leaf nodes as shown in Figure 4.1 (for the particular case $L = 3$); we then associate each $B_l \in \mathcal{B}_l$, $1 \le l \le L$, to a node in the tree. We note that appending a '0' to a vector B_l corresponds to a left bend in the tree and appending a '1' to a vector B_l corresponds to a right bend in the tree; we define these "bends" by the concatenation

$$(B_l, i) \equiv (1, i_2 \ldots, i_l, i), \qquad i \in \{0,1\} \tag{4.3}$$

and we say that B_l is the parent of the two children (B_l, i), $i \in \{0,1\}$.

Given an initial parameter domain $\mathcal{D}$, we shall perform the "h"-refinement by recursive splitting of $\mathcal{D}$ into smaller parameter subdomains. The subdomains will be defined hierarchically and thus can be associated to a tree; we assume for the moment that we can organize the $2^L - 1$ subdomains in a *perfect* binary tree. We denote the subdomains as

$$\mathcal{V}_{B_l} \subset \mathcal{D}, \quad B_l \in \mathcal{B}_l, \quad 1 \le l \le L, \tag{4.4}$$

and we require the parent-child hierarchy

$$\mathcal{V}_{(B_l,0)} \subset \mathcal{V}_{B_l}, \tag{4.5}$$

$$\mathcal{V}_{(B_l,1)} \subset \mathcal{V}_{B_l}. \tag{4.6}$$

We associate to each subdomain $\mathcal{V}_{B_l}$ a set of $\bar{N}$ parameter values denoted by

$$\mathcal{M}_{\bar{N},B_l} = \{\boldsymbol{\mu}_{1,B_l}, \dots, \boldsymbol{\mu}_{\bar{N},B_l}\}, \quad B_l \in \mathcal{B}_l, \quad 1 \le l \le L, \tag{4.7}$$

in which $\boldsymbol{\mu}_{1,B_l} \in \mathcal{V}_{B_l}, \dots, \boldsymbol{\mu}_{\bar{N},B_l} \in \mathcal{V}_{B_l}$; we may then define the RB approximation spaces (of dimension $\bar{N}$) associated with the "models" $\mathcal{M}_{\bar{N},B_l}$ and subdomains $\mathcal{V}_{B_l}$, $B_l \in \mathcal{B}_l$, $1 \le l \le L$, as

$$X_{\bar{N},B_l} = \operatorname{span}\{u(\boldsymbol{\mu}_{1,B_l}), \dots, u(\boldsymbol{\mu}_{\bar{N},B_l})\}, \quad B_l \in \mathcal{B}_l, \quad 1 \le l \le L. \tag{4.8}$$

(The actual bases are, as always, orthonormalized.)

To each model $\mathcal{M}_{\bar{N},B_l}$ and corresponding subdomain $\mathcal{V}_{B_l}$ we associate a parameter *anchor point*, $\hat{\boldsymbol{\mu}}_{B_l}$, defined as

$$\hat{\boldsymbol{\mu}}_{B_l} \equiv \boldsymbol{\mu}_{1,B_l}. \tag{4.9}$$

We shall further impose (by construction) that, for $1 \le l \le L-1$,

$$\hat{\boldsymbol{\mu}}_{(B_l,0)} = \hat{\boldsymbol{\mu}}_{B_l}, \tag{4.10}$$

$$\hat{\boldsymbol{\mu}}_{(B_l,1)} \ne \hat{\boldsymbol{\mu}}_{B_l}; \tag{4.11}$$

the anchor point is thus inherited only by the "left" child. The partition of $\mathcal{D}$ into subdomains is inferred from proximity to the anchor points.

To this end, we introduce for any Boolean vector $B_l \in \mathcal{B}_l$, $1 \le l \le L$, a "proximity function" $d_{B_l} : \mathcal{D} \to \mathbb{R}^+$. For example, we can choose the Euclidean distance between two points,

$$d_{B_l}(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{B_l}\|_2. \tag{4.12}$$

We then successively evaluate the proximity function to determine, for any given $1 \le l \le L$, which subdomain $\mathcal{V}_{(1,i_2^*,\dots,i_l^*)} \subset \mathcal{D}$ contains a given $\boldsymbol{\mu} \in \mathcal{D}$,

$$\begin{aligned} i_2^* &= \arg \min_{i \in \{0,1\}} d_{(1,i)}(\boldsymbol{\mu}), \\ i_3^* &= \arg \min_{i \in \{0,1\}} d_{(1,i_2^*,i)}(\boldsymbol{\mu}), \\ &\vdots \\ i_l^* &= \arg \min_{i \in \{0,1\}} d_{(1,i_2^*,\dots,i_{l-1}^*,i)}(\boldsymbol{\mu}). \end{aligned} \tag{4.13}$$

We discuss the computational complexity shortly.

In general, the partition will not have the same number of refinement levels along every branch of the associated binary tree: in practice, the tree is not necessarily perfect. In this case, L shall denote the maximum number of levels in the tree—the *tree depth.* Branches are terminated with "empty leaf models" $\mathcal{M}_{\bar{N},(B_l,0)} = \mathcal{M}_{\bar{N},(B_l,1)} = \emptyset$ for some $B_l \in \mathcal{B}_l$, $1 \le l \le L$. (For any such B_l associated with empty child models we adopt the convention $d_{(B_l,0)} = d_{(B_l,1)} \equiv \infty$; we then terminate the search (4.13) whenever $d_{(1,i_2^*,\dots,i)}(\boldsymbol{\mu}) = \infty$ (for $i = 0, 1$).) We now define K as the number of leaf nodes in the tree (*exclusive* of the terminator empty models). The uniformity of the tree associated with the partition of $\mathcal{D}$ can be measured by a *relative tree depth*

$$\eta_{\text{depth}} = \frac{\text{tree depth } (= L)}{\log_2 K + 1}; \tag{4.14}$$

note that $\eta_{\text{depth}} \geq 1$, and that $\eta_{\text{depth}} = 1$ corresponds to a perfect binary tree.

Algorithm 2: Greedy^2(Ξ_{B_l}, $\boldsymbol{\mu}_{1,B_l}$, ϵ_{tol}, $\tilde{N}_{\max,B_l}$)

```
initialize: $N \leftarrow 0$, $\epsilon_{0,B_l} \leftarrow \infty$, $X_{0,B_l} \leftarrow \{0\}$, $\mathcal{M}_{0,B_l} \leftarrow \emptyset$
while $\epsilon_N > \epsilon_{\text{tol}}$ and $N < \tilde{N}_{\max,B_l}$ do
    $N \leftarrow N + 1$
    $X_{N,B_l} \leftarrow X_{N-1,B_l} \oplus \text{span}\{u(\boldsymbol{\mu}_{N,B_l})\}$
    $\mathcal{M}_{N,B_l} \leftarrow \mathcal{M}_{N-1,B_l} \cup \{\boldsymbol{\mu}_{N,B_l}\}$
    $\epsilon_{N,B_l} \leftarrow \max_{\boldsymbol{\mu} \in \Xi_{B_l}} \Delta_N^{h\text{RB}}(\boldsymbol{\mu})$
    $\boldsymbol{\mu}_{N+1,B_l} \leftarrow \arg\max_{\boldsymbol{\mu} \in \Xi_{B_l}} \Delta_N^{h\text{RB}}(\boldsymbol{\mu})$
end
$N_{\max,B_l} \leftarrow N$
```

In what follows we shall need Algorithm 2, which is largely a restatement of Greedy^1 (Algorithm 1) restricted to a particular subdomain $\mathcal{V}_{B_l}$ for given $B_l \in \mathcal{B}_l$. The evaluation of the *a posteriori* error bound $\Delta_N^{h\text{RB}}$ (defined shortly) is now performed over $\Xi_{B_l} \subset \mathcal{V}_{B_l}$. The output of the algorithm is an RB space $X_{N_{\max,B_l}}$ and an associated model $\mathcal{M}_{N_{\max,B_l},B_l}$. Note that even for $N_{\max,B_l} = 1$ we perform one pass of the whole loop and hence identify (and retain) $\boldsymbol{\mu}_{2,B_l}$; however, in general, we only compute at most $\tilde{N}_{\max,B_l}$ snapshots. For the "h"-type RB approximation of this section we shall require $N_{\max,B_l} \equiv \tilde{N}_{\max,B_l} \equiv \bar{N}$ for all B_l.

4.2. Approximation. We now introduce the equi-order "h"-type RB approximation algorithm. We start from the original parameter domain $\mathcal{V}_{(1)} = \mathcal{D}$ ($l = 1$, $B_l = (1)$); we introduce a finite train sample $\Xi_{(1)} \subset \mathcal{V}_{(1)}$; we choose an initial parameter anchor point $\hat{\boldsymbol{\mu}}_{(1)} \in \mathcal{D}$; we choose the error tolerance ϵ_{tol}^1; we set the desired maximum RB space dimension $\bar{N} \geq 1$. The partition is then determined as follows.

1. Construct a model with $\bar{N}$ parameter values for the current subdomain $\mathcal{V}_{B_l}$ with the standard Greedy^2 algorithm (Algorithm 2). The RB space $X_{\bar{N},B_l}$ and the model $\mathcal{M}_{\bar{N},B_l}$ are outputs from Greedy^2(Ξ_{B_l}, $\hat{\boldsymbol{\mu}}_{B_l}$, $0, \bar{N}$). Note that we set the argument $\epsilon_{\text{tol}} = 0$ to enforce a RB space of dimension $\bar{N}$.[1]

2. Compute the maximum *a posteriori* error bound over the train sample associated with the current subdomain

$$\epsilon_{\bar{N},B_l} = \max_{\boldsymbol{\mu} \in \Xi_{B_l}} \Delta_{\bar{N}}^{h\text{RB}}(\boldsymbol{\mu}). \tag{4.15}$$

[1] We assume here that the parametric manifold $\mathcal{M} = \{u(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{D}\}$ can not be approximated exactly by $\bar{N} \ll \mathcal{N}$ snapshots. We also assume here that the train sample over each subdomain is sufficiently rich.

3. If $\epsilon_{\bar{N},B_l} < \epsilon^1_{\rm tol}$: The refinement is sufficiently good; set

$$\mathcal{M}_{\bar{N},(1,i_2,\ldots,i_l,0)} = \emptyset, \tag{4.16}$$

$$\mathcal{M}_{\bar{N},(1,i_2,\ldots,i_l,1)} = \emptyset; \tag{4.17}$$

we thus terminate the branch of the associated binary tree.

4. If $\epsilon_{\bar{N},B_l} \geq \epsilon^1_{\rm tol}$:

(i) Define anchor points for two new models $\mathcal{M}_{\bar{N},(B_l,0)}$ and $\mathcal{M}_{\bar{N},(B_l,1)}$ as $\hat{\boldsymbol{\mu}}_{(B_l,0)} = \hat{\boldsymbol{\mu}}_{B_l}$ and $\hat{\boldsymbol{\mu}}_{(B_l,1)} = \boldsymbol{\mu}_{2,B_l}$, respectively. The model $\mathcal{M}_{\bar{N},(B_l,0)}$ inherits the anchor point from its "parent," while the model $\mathcal{M}_{\bar{N},(B_l,1)}$ takes as anchor point the first parameter value chosen by the Greedy^2 algorithm; in the sense of the *a posteriori* error estimator, these two points are *maximally different* and hence good places to "anchor" the new models. When $\bar{N} \geq 2$ the remaining $\bar{N} - 2$ parameter values of $\mathcal{M}_{\bar{N},B_l}$, as well as the associated snapshots and approximation spaces, are discarded.

(ii) Define a new and denser train sample $\tilde{\Xi}_{B_l} \subset \mathcal{V}_{B_l}$ of size $|\tilde{\Xi}_{B_l}| = 2|\Xi_{(1)}|$. (The temporary sample $\tilde{\Xi}_{B_l}$ is thus twice as large as the initial train sample.)

(iii) Construct $\Xi_{(B_l,0)} \subset \mathcal{V}_{(B_l,0)}$ and $\Xi_{(B_l,1)} \subset \mathcal{V}_{(B_l,1)}$ from $\tilde{\Xi}_{B_l}$ based on proximity to $\hat{\boldsymbol{\mu}}_{(B_l,0)}$ and $\hat{\boldsymbol{\mu}}_{(B_l,1)}$, respectively: a point $\boldsymbol{\mu} \in \tilde{\Xi}_{B_l}$ belongs to $\Xi_{(B_l,0)}$ if and only if $d_{(B_l,0)}(\boldsymbol{\mu}) \leq d_{(B_l,1)}(\boldsymbol{\mu})$; otherwise $\boldsymbol{\mu}$ belongs to $\Xi_{(B_l,1)}$.

5. Split the current branch into two new branches: set $B^{\rm left}_{l+1} = (B_l, 0)$ and $B^{\rm right}_{l+1} = (B_l, 1)$; update $l \leftarrow l + 1$ and proceed to step 1 first for $B_l = B^{\rm left}_l$ and then for $B_l = B^{\rm right}_l$.

Algorithm 3: $h{\rm RB}(\Xi_{B_l}, \hat{\boldsymbol{\mu}}_{B_l}, \bar{N}, \epsilon^1_{\rm tol})$

```
Find $X_{\bar{N},B_l}$ and $\mathcal{M}_{\bar{N},B_l}$ from Greedy$^2(\Xi_{B_l}, \hat{\boldsymbol{\mu}}_{B_l}, \infty, \bar{N})$
$\epsilon_{\bar{N},B_l} \leftarrow \max_{\boldsymbol{\mu}\in\Xi_{B_l}} \Delta^{h{\rm RB}}_{\bar{N}}(\boldsymbol{\mu})$;
if $\epsilon_{\bar{N},B_l} < \epsilon^1_{\rm tol}$ then
    Terminate branch: $\mathcal{M}_{\bar{N},(B_l,0)} = \emptyset$ and $\mathcal{M}_{\bar{N},(B_l,1)} = \emptyset$.
else
    Define $\hat{\boldsymbol{\mu}}_{(B_l,0)} = \boldsymbol{\mu}_{1,B_l}$ and $\hat{\boldsymbol{\mu}}_{(B_l,1)} = \boldsymbol{\mu}_{2,B_l}$
    Construct $\Xi_{(B_l,0)} \subset \mathcal{V}_{(B_l,0)}$ and $\Xi_{(B_l,1)} \subset \mathcal{V}_{(B_l,1)}$
    $h{\rm RB}(\Xi_{(B_l,0)}, \hat{\boldsymbol{\mu}}_{(B_l,0)}, \bar{N}, \epsilon^1_{\rm tol})$
    $h{\rm RB}(\Xi_{(B_l,1)}, \hat{\boldsymbol{\mu}}_{(B_l,1)}, \bar{N}, \epsilon^1_{\rm tol})$
end
```

The procedure may be more precisely defined by $h{\rm RB}(\Xi_{(1)}, \hat{\boldsymbol{\mu}}_{(1)}, \bar{N}, \epsilon^1_{\rm tol})$, where hRB is the recursive function defined in Algorithm 3. The output from Algorithm 3 is K subdomains associated with the K leaf nodes of the binary tree (exclusive of the terminator empty models). Each subdomain is associated to an $\bar{N}$-parameter model and a $\bar{N}$-dimensional approximation space. We emphasize that the *intermediate* models and approximation spaces—associated with *non-leaf* nodes at earlier levels in the tree—are discarded and do not "survive" with respect to the online stage. Finally, we note that the depth of the tree, L, is simply the number of nodes in the longest branch (exclusive of the terminator empty models).

Remark 1 (Train Sample Refinement). In step 4(ii) in the algorithm above, additional points are added to the train sample such that the number of points in the two *new* train samples will be roughly the same as in the old train sample, and in

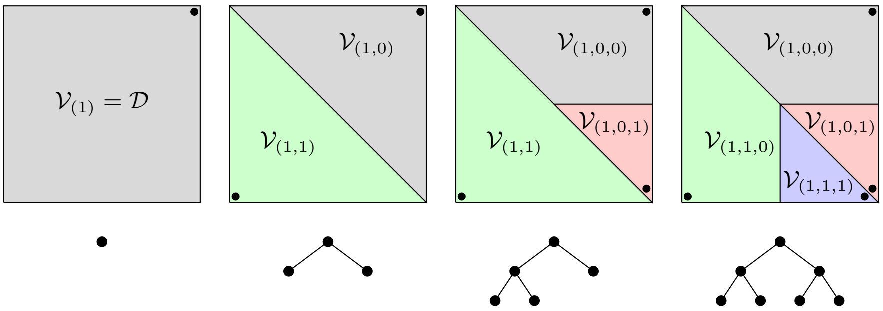

FIGURE 4.2. *Two levels of "h"-refinement and associated binary tree; here $L = 3$.*

particular always much larger than $\bar{N}$. As a result, the "global" train sample over $\mathcal{D}$—the union of all the points in the train samples over all parameter subdomains—is adaptively refined as the "h"-type RB approximation becomes more accurate: the train sample is denser in regions of $\mathcal{D}$ with smaller subdomains. We thus effectively include more train points where the solution varies more rapidly with the parameters.

The train sample refinement is performed by a simple accept-reject Monte Carlo procedure: we draw from the uniform distribution over $\mathcal{D}$; we then use the search (4.13) to determine whether a point belongs to a subdomain and thus can be included as a new point in the associated train sample. In the case that the proximity function is Euclidean distance (as in (4.12)), we need not sample from the entire parameter domain $\mathcal{D}$: we first compute the bounding box of the old train sample; we then sample the new points from a larger box that contains the bounding box with some safety margin—the assumption is that this larger box contains the entire subdomain. In the case in which the proximity function is the error bound (as we describe shortly), we sample from the entire domain $\mathcal{D}$ since we have no *a priori* knowledge of the shape or connectedness of the subdomains.

An alternative and clearer approach to the train sample refinement 4(ii)–4(iii) might be to first split the current train sample into train samples associated with each subdomain, and then enrich each of these samples to achieve size $|\Xi_{(1)}|$. However, for the numerical results in this paper we pursue the "first enrich then split" procedure described in step 4. above. We note that as long as the train samples are sufficiently rich the particular refinement procedure will not affect the numerical results significantly. $\Diamond$

Remark 2 ("Redundant" Truth Solves). The Greedy algorithm—in particular in the case of a low-order (small $\bar{N}$) approximation—is likely to choose parameter values close to the boundaries of the parameter subdomains. As a result, two or more models may comprise some identical (or nearly identical) parameter values, and thus some of the offline truth solves are in some sense redundant. One way to reduce this snapshot redundancy is to share basis functions between approximation spaces if the associated greedily selected parameter values are sufficiently close. The development of an efficient algorithm for automatic sharing of basis functions is the subject of future work. $\Diamond$

In Figure 4.2 we illustrate the first two levels of "h"-refinement together with the associated binary tree for an "h"-type approximation with $\bar{N} = 1$. The initial domain is $\mathcal{V}_{(1)} = \mathcal{D}$ and the initial model is $\mathcal{M}_{1,(1)} = \{\hat{\boldsymbol{\mu}}_{(1)}\}$ where the anchor point $\hat{\boldsymbol{\mu}}_{(1)} = \boldsymbol{\mu}_{1,(1)}$ is chosen as the upper right corner of the parameter domain. The method then greedily chooses the point $\boldsymbol{\mu}_{2,(1)}$ near the lower left corner of $\mathcal{V}_{(1)}$; the initial anchor point is then re-labelled as $\hat{\boldsymbol{\mu}}_{(1,0)} = \hat{\boldsymbol{\mu}}_{(1)}$ and the greedily chosen point is re-labelled as $\hat{\boldsymbol{\mu}}_{(1,1)} = \boldsymbol{\mu}_{2,(1)}$. We now define two new models $\mathcal{M}_{1,(1,0)} = \{\hat{\boldsymbol{\mu}}_{(1,0)}\}$ and $\mathcal{M}_{1,(1,1)} = \{\hat{\boldsymbol{\mu}}_{(1,1)}\}$, whose associated subdomains $\mathcal{V}_{(1,0)}$ and $\mathcal{V}_{(1,1)}$ are determined from proximity—here Euclidean distance—to the two anchor points. Next, $\mathcal{V}_{(1,1)}$ and $\mathcal{V}_{(1,0)}$ are partitioned in the same fashion (we assume here that the tolerance is satisfied within $\mathcal{V}_{(1,0,0)}$ and $\mathcal{V}_{(1,0,1)}$).

Finally, we may now define the "h"-type RB approximation. Given any $\boldsymbol{\mu} \in \mathcal{D}$ we first determine the subdomain $\mathcal{V}_{B_l^*}$ containing $\boldsymbol{\mu}$ from the search (4.13); for a perfect binary tree, $l = L$, however more generally $l \leq L$. We then find $u_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu}) \in X_{\bar{N},B_l^*}$ such that

$$a(u_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in X_{\bar{N},B_l^*}. \tag{4.18}$$

(Note that B_l^* depends on $\boldsymbol{\mu}$.) We discuss computational complexity shortly. Finally, we *formally* define the "order" of the "h"-type approximation as $p \equiv \bar{N}^{1/P} - 1$. We elaborate on this interpretation in Remark 5.

4.3. *A Posteriori* **Error Estimation.** We can apply the same *a posteriori* bound developed for the "p"-type RB approximation in §3.2 to the "h"-type (and below, "hp"-type) RB approximations. However, we shall require some new notation for the "h"-type error bound.

Given any $\boldsymbol{\mu} \in \mathcal{D}$ and a partition of $\mathcal{D}$ into subdomains, we determine B_l^* from the binary search (4.13) and compute the RB solution $u_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu})$ from (4.18). The RB residual is

$$r_{\bar{N}}^{h\mathrm{RB}}(v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) - a(u_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}), \quad \forall v \in X; \tag{4.19}$$

the Riesz representation of the residual is denoted by $\mathcal{R}_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu})$. Our upper bound for the X-norm error $\|u(\boldsymbol{\mu}) - u_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu})\|_X$ is then given by

$$\Delta_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu}) \equiv \frac{\|\mathcal{R}_{\bar{N}}^{h\mathrm{RB}}(\boldsymbol{\mu})\|_X}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})}. \tag{4.20}$$

Lemma 3.1 now directly applies with an appropriate change of notation.

Remark 3 (The Error Bound as Proximity Function). For any $B_l \in \mathcal{B}_l$ (associated with a non-empty model), $1 \leq l \leq L$, and any $\boldsymbol{\mu} \in \mathcal{D}$, we can derive the RB error bound associated with the RB approximation to $u(\boldsymbol{\mu})$ in the space $X_{\bar{N},B_l}$; we denote this error bound by $\Delta_{\bar{N},B_l}(\boldsymbol{\mu})$. As an alternative to the proximity function introduced in (4.12), we can use

$$d_{B_l}(\boldsymbol{\mu}) = \Delta_{\bar{N}=1,B_l}(\boldsymbol{\mu}), \tag{4.21}$$

to measure the "distance" between the points $\hat{\boldsymbol{\mu}}_{B_l}$ and $\boldsymbol{\mu}$. Note that we use the error bound assocated with the RB approximation for $\bar{N} = 1$ (which is simply a multiple of the snapshot associated with the anchor point) even when $\bar{N} > 1$; hence evaluation of (4.21) does not depend on $\bar{N}$. In §6, we provide results with the proximity function defined both as in (4.12) and as in (4.21). $\Diamond$

Remark 4 (Multiple Inner Products). The "h"-type RB approximation offers a natural way of introducing multiple X-inner products (2.4) in the computation of the dual norm of the residual for the *a posteriori* error bounds—we may choose a different X-inner product for each subdomain. For example, we could choose the anchor point in any subdomain to be the reference parameter, and thus define an “optimized” inner product, associated with that subdomain. With this approach, we would expect sharper error bounds and thus a better parameter domain partition (as well as, ultimately, greater online efficiency).

To compute the dual norm of the residual we must (in the construction stage) solve a number of problems on the form (3.4) with different right-hand sides. If we solve the discrete system directly, we must invert one operator for each inner product; hence there is a computational advantage associated with only a single inner product. If we solve the discrete system iteratively, however, we can introduce individual inner products within each subdomain at very little computational penalty. In this paper, however, we have not pursued a multiple inner product approach for our numerical examples. $\Diamond$

4.4. Offline-Online Decomposition. In the *offline* stage, we determine the partition of the parameter domain and construct the corresponding RB models and spaces: we perform $h\mathrm{RB}(\Xi_{(1)}, \hat{\boldsymbol{\mu}}_{(1)}, \bar{N}, \epsilon^1_{\mathrm{tol}})$. For the purposes of this subsection, we assume a perfect binary tree; note that a perfect binary tree with K leaf nodes has $2K-1$ nodes in total. We also assume that the cardinality of the train sample over each of the subdomains is equal to n_{train}.

The offline stage computational cost derives from several components:

1. Snapshot Truth Solves. In the case $\bar{N} \geq 2$ each node in the tree except the root node inherits one snapshot from its parent. We must thus compute $\bar{N}$ snapshots for the model associated with the root node, and $\bar{N}-1$ additional snapshots for each of the $2K-2$ models associated with all nodes except the root node. Note that we only retain the basis functions associated with the K leaf nodes; we thus discard $\bar{N}-2$ snapshots for each intermediate (non-leaf) model. In the case $\bar{N}=1$ we compute $\bar{N}K$ snapshots in total since the snapshot associated with an intermediate model is inherited by one of the children.

2. Reduced Basis Preprocessing. In the case $\bar{N} \geq 2$ we must compute $(Q_a\bar{N}^2 + Q_f\bar{N})$ truth inner products to form the parameter-independent “stiffness” matrices and loads (e.g., as in (3.12)) for the model associated with the root node, and $(2K-2)(Q_a(\bar{N}^2-1) + Q_f(\bar{N}-1))$ additional truth inner products in total to form the parameter-independent “stiffness” matrices and loads for the remaining $2K-2$ models. In the case $\bar{N}=1$ we must compute $(2K-1)(Q_a\bar{N}^2 + Q_f\bar{N})$ truth inner products in total to form the parameter-independent “stiffness” matrices and loads.

3. Error Bound Preprocessing. In the case $\bar{N} \geq 2$ we must compute $\bar{N}Q_a + Q_f$ truth Poisson solves of the form (3.18) for the model associated with the root node, and $(2K-2)(\bar{N}-1)Q_a$ additional truth Poisson solves in total for the remaining $(2K-2)$ models. We must also compute $(\bar{N}Q_a + Q_f)^2$ truth inner products of the form (3.21) related to the dual norm of the residual for the model associated with the root node, and $(2K-2)((\bar{N}Q_a + Q_f)^2 - (Q_a + Q_f)^2)$ additional truth inner products in order to evaluate the dual norm of the residual for the $(2K-2)$ remaining models. In the case $\bar{N}=1$ we must compute $(2K-1)\bar{N}Q_a + Q_f$ truth Poisson solves in total and $(2K-1)(\bar{N}Q_a + Q_f)^2$ truth inner-products in total.

4. Error Bound Evaluations. For $\bar{N} \geq 1$ we must solve $n_{\mathrm{train}}\bar{N}(2K-1)$ RB systems to obtain the residual coefficients and evaluate $n_{\mathrm{train}}\bar{N}(2K-1)$ RB error bounds during the Greedy sampling including both the intermediate and final models. This results, to leading order, in $\mathcal{O}(n_{\mathrm{train}}\bar{N}(2K-1)(\bar{N}^3 + \bar{N}^2Q^2))$ operations in total.

In the case $\bar{N} \geq 2$ the combined (1-4) offline cost is thus approximately $2\bar{N}K$

truth snapshot computations, $2Q_a\bar{N}K+Q_f$ truth Poisson solves, $2K(Q_a\bar{N}^2+Q_f\bar{N})+2K(\bar{N}Q_a+Q_f)^2$ truth inner products, and $\mathcal{O}(n_{\text{train}}2\bar{N}K(\bar{N}^3+\bar{N}^2Q^2))$ operations to evaluate the error bounds. Note that the additional cost associated with the (ultimately discarded) $K-1$ intermediate models required for the construction of the parameter domain partition is not onerous—a factor of two. In the case $\bar{N}=1$ we retain all computed entities associated with the indermediate models during the partition procedure for the final models, and there is thus only minor additional cost (anchor point identification in 4.) associated with the partition procedure.

The link between the offline and online stages is the parameter-independent data *constructed* in the offline stage and stored (permanently) for *evaluation* in the online stage. Since we retain only the data associated with the final models, the online storage for the "h"-type RB approximation is Q_aK matrices of size $\bar{N}\times\bar{N}$ and Q_fK vectors of size $\bar{N}$; the online storage associated with the RB error bounds is $K(\bar{N}Q_a+Q_f)^2/2$. (If we were to retain intermediate models for purposes of online adaptivity clearly the online storage would increase; we do not consider this case further since in actual practice online adaptivity is typically pursued through the "*hp*"-approach.)

In the *online* stage, given any $\boldsymbol{\mu}\in\mathcal{D}$, we first determine the subdomain which contains $\boldsymbol{\mu}$ via the binary search (4.13) in $\mathcal{O}(\log_2 K)$ operations. Thanks to the construction-evaluation decomposition, we can then assemble and solve the corresponding system of algebraic equations in $\mathcal{O}(Q\bar{N}^2)$ and $\mathcal{O}(\bar{N}^3)$ operations, respectively, and compute the associated *a posteriori* error bound in $\mathcal{O}(\bar{N}^2Q^2)$ operations. Note that the search (4.13) is an $\mathcal{O}(\log_2 K)$ operation only under the hypothesis that the depth of the tree associated with the partition of $\mathcal{D}$, L, is proportional to $\log_2 K$; we provide numerical results to support this hypothesis in §6. We also emphasize that the efficient $\mathcal{O}(\log_2 K)$ search is a particular property of our hierarchical partition construction; if we were to partition the parameter domain based on (say) a Voronoi diagram, determination of the subdomain which contains $\boldsymbol{\mu}\in\mathcal{D}$ would be less efficient.

4.5. *A Priori* **Theory: $\bar{N}=1$, $P=1$.** In this section we develop an *a priori* convergence theory for a "h"–type RB approximation of "zeroth-order" ($\bar{N}=1$) in the one-parameter case ($P=1$) when the Euclidean distance is used as the proximity function. We focus on $\bar{N}=1$ since in fact $\bar{N}=1$ is crucial to the "*hp*"-approach of §5: the theory developed here demonstrates that an $\bar{N}=1$ greedy approach can indeed generate a reasonably efficient partition; convergence is crucial for offline and also ultimately online performance. We consider $P=1$ for simplicity; at the conclusion of this section we provide a remark addressing (non-rigorously) the $\bar{N}>1$ (higher "order") and $P>1$ cases.

For our purposes here, we do not need the Boolean indexing of the anchor points and subdomains: we shall consider Algorithm 3 after generation of K subdomains; we re-label the K anchor points as $\hat{\boldsymbol{\mu}}'_1,\hat{\boldsymbol{\mu}}'_2,\dots,\hat{\boldsymbol{\mu}}'_K$ (numbered in the order in which they are chosen by Algorithm 3). When Algorithm 3 identifies a new anchor point (and thus subdomain), the parameter domain partition changes; we introduce mappings $I_{\tilde{K}}:\mathcal{D}\to\{1,\dots,\tilde{K}\}$, $1\le\tilde{K}\le K$, such that with $\tilde{K}$ anchor points, for any $\boldsymbol{\mu}\in\mathcal{D}$,

$$\hat{\boldsymbol{\mu}}'_{I_{\tilde{K}}(\boldsymbol{\mu})}=\hat{\boldsymbol{\mu}}_{B^*(\boldsymbol{\mu};\tilde{K})}, \tag{4.22}$$

where $B^*(\boldsymbol{\mu};\tilde{K})$ is the Boolean index of the particular subdomain (amongst the $\tilde{K}$ subdomains) containing $\boldsymbol{\mu}$. Below, we omit the $'$ for brevity.

For the purpose of this section, given $\tilde{K}$ anchor points and corresponding subdomains, we denote by $u_{\tilde{K}}(\boldsymbol{\mu})$ the "zeroth order" ($\bar{N}=1$) "h"-type RB approximation for any $\boldsymbol{\mu} \in \mathcal{D}$. With the implicit mapping above, we have

$$u_{\tilde{K}}(\boldsymbol{\mu}) = \omega_{\tilde{K}}(\boldsymbol{\mu}) u(\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\boldsymbol{\mu})}), \tag{4.23}$$

where the coefficient $\omega_{\tilde{K}}(\boldsymbol{\mu})$ is given by the Galerkin projection as

$$\omega_{\tilde{K}}(\boldsymbol{\mu}) = \frac{f\big(u(\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\boldsymbol{\mu})}); \boldsymbol{\mu}\big)}{a\big(u(\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\boldsymbol{\mu})}), u(\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\boldsymbol{\mu})}); \boldsymbol{\mu}\big)}. \tag{4.24}$$

Note that (4.23) holds since we consider $\bar{N} = 1$: a *single* RB basis function associated with each subdomain. We denote by $r_{\tilde{K}}(v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) - a(u_{\tilde{K}}(\boldsymbol{\mu}), v; \boldsymbol{\mu})$ the RB residual, and let $\mathcal{R}_{\tilde{K}}(\boldsymbol{\mu}) \in X$ satisfy $(\mathcal{R}_{\tilde{K}}(\boldsymbol{\mu}), v)_X = r_{\tilde{K}}(v; \boldsymbol{\mu})$ for all $v \in X$. Our X-norm error upper bound is then written in this subsection as

$$\Delta_{\tilde{K}}(\boldsymbol{\mu}) = \frac{\|\mathcal{R}_{\tilde{K}}(\boldsymbol{\mu})\|_X}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})}, \tag{4.25}$$

which is simply a specialization of (4.20).

We need two further preliminary results. First, it is clear from Cea’s Lemma (with respect to the X-norm), (2.9), and (2.8) that for any $\tilde{K}$, $1 \le \tilde{K} \le K$, and any $\boldsymbol{\mu} \in \mathcal{D}$,

$$\|u(\boldsymbol{\mu}) - u_{\tilde{K}}(\boldsymbol{\mu})\|_X \le \frac{\overline{\gamma}}{\underline{\alpha}} \|u(\boldsymbol{\mu}) - u(\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\boldsymbol{\mu})})\|_X, \tag{4.26}$$

since $u(\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\boldsymbol{\mu})})$ is a particular member of the (one-dimensional) reduced basis space. Second, from (3.7) of Lemma 3.1, we obtain for any $\tilde{K}$, $1 \le \tilde{K} \le K$, and any $\boldsymbol{\mu} \in \mathcal{D}$,

$$\Delta_{\tilde{K}}(\boldsymbol{\mu}) \le \frac{\overline{\gamma}}{\underline{\alpha}} \|u(\boldsymbol{\mu}) - u_{\tilde{K}}(\boldsymbol{\mu})\|_X, \quad 1 \le \tilde{K} \le K. \tag{4.27}$$

We can now state

Proposition 4.1 (Convergence in the case $\bar{N} = 1$, $P = 1$). *The “h”-type RB approximation terminates for finite $K(\epsilon^1_{\mathrm{tol}})$ subdomains. Further, the convergence is first order in the sense that*

$$K(\epsilon^1_{\mathrm{tol}}) \le \max\left\{1, \frac{C}{\epsilon^1_{\mathrm{tol}}}\right\} \tag{4.28}$$

for a constant C given by

$$C = \frac{2\bar{\gamma}^2 \tilde{C} |\mathcal{D}|}{\underline{\alpha}^2}, \tag{4.29}$$

where $\tilde{C} = (\tilde{c}_1 \max_{\boldsymbol{\mu} \in \mathcal{D}} \|f(\cdot; \boldsymbol{\mu})\|_{X'} + \underline{\alpha}\tilde{c}_2)/\underline{\alpha}^2$ is the constant developed in Lemma 2.1 and $|\mathcal{D}|$ is the length of $\mathcal{D} \subset \mathbb{R}$.

Proof. Algorithm 3 provides a sequence of anchor points $\hat{\boldsymbol{\mu}}_1, \dots, \hat{\boldsymbol{\mu}}_K$ for $K \ge 1$. We have by construction of our algorithm *either* $K = 1$ *or* $K > 1$ and

$$\epsilon^1_{\mathrm{tol}} < \Delta_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1}), \quad 1 \le \tilde{K} \le K - 1. \tag{4.30}$$

In the former case the proof is complete; we henceforth consider the latter case.

We deduce from (4.27), (4.26), and Lemma 2.1, respectively, that

$$\begin{aligned}\Delta_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1}) &\le \frac{\overline{\gamma}}{\underline{\alpha}}\|u(\hat{\boldsymbol{\mu}}_{\tilde{K}+1}) - u_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})\|_X && (4.31)\\ &\le \frac{\overline{\gamma}^2}{\underline{\alpha}^2}\|u(\hat{\boldsymbol{\mu}}_{\tilde{K}+1}) - u(\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})})\|_X && (4.32)\\ &\le \frac{\overline{\gamma}^2}{\underline{\alpha}^2}\tilde{C}|\hat{\boldsymbol{\mu}}_{\tilde{K}+1} - \hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})}|, && (4.33)\end{aligned}$$

for $1 \le \tilde{K} \le K-1$; hence from (4.30)

$$|\hat{\boldsymbol{\mu}}_{\tilde{K}+1} - \hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})}| > \frac{\underline{\alpha}^2\epsilon^1_{\mathrm{tol}}}{\overline{\gamma}^2\tilde{C}}, \tag{4.34}$$

for $1 \le \tilde{K} \le K-1$.

For $\tilde{K}$, $1 \le \tilde{K} \le K-1$, the algorithm selects the next anchor point $\hat{\boldsymbol{\mu}}_{\tilde{K}+1}$ and the intermediate subdomain associated with anchor point number $I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})$ is divided into two new subdomains. It is clear that the length of each of the two new subdomains is at least as large as half the distance between the new anchor point $\hat{\boldsymbol{\mu}}_{\tilde{K}+1}$ and anchor point $\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})}$, namely $|\hat{\boldsymbol{\mu}}_{\tilde{K}+1} - \hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})}|/2$ (recall that the distance function is the Euclidean distance and that we consider $P = 1$ parameter). Let δ^K_k denote the length of the subdomain associated with anchor point $\hat{\boldsymbol{\mu}}_k$, $1 \le k \le K$. We note that each of the K subdomains generated by Algorithm 3 results from the splitting of the intermediate subdomain associated with anchor point $\hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})}$ for *some* $\tilde{K} \in \{1,\dots,K-1\}$; hence for $1 \le k \le K$, there exists a $\tilde{K} \in \{1,\dots,K-1\}$ such that

$$\delta^K_k \ge |\hat{\boldsymbol{\mu}}_{\tilde{K}+1} - \hat{\boldsymbol{\mu}}_{I_{\tilde{K}}(\hat{\boldsymbol{\mu}}_{\tilde{K}+1})}|/2, \tag{4.35}$$

and thus by (4.34)

$$\underline{\delta}^K \equiv \min_{1\le k\le K} \delta^K_k > \frac{\underline{\alpha}^2\epsilon^1_{\mathrm{tol}}}{2\overline{\gamma}^2\tilde{C}}. \tag{4.36}$$

Note that $\underline{\delta}^K$ is not the smallest distance between two anchor points: rather, it is the smallest length of any of the K subdomains.

Let $|\mathcal{D}|$ denote the length of $\mathcal{D}$. With K subdomains, it is clear that $K\underline{\delta}^K \le |\mathcal{D}|$. We now assume $K > C/\epsilon^1_{\mathrm{tol}}$. From (4.36) it then follows that

$$K\underline{\delta}^K > \frac{C}{\epsilon^1_{\mathrm{tol}}}\underline{\delta}^K \ge \left(\frac{2\overline{\gamma}^2\tilde{C}|\mathcal{D}|}{\underline{\alpha}^2\epsilon^1_{\mathrm{tol}}}\right)\left(\frac{\underline{\alpha}^2\epsilon^1_{\mathrm{tol}}}{2\overline{\gamma}^2\tilde{C}}\right) = |\mathcal{D}|, \tag{4.37}$$

which is clearly false. We have thus reached a contradiction: the "h"-type RB approximation can not generate a sequence of anchor points $\hat{\boldsymbol{\mu}}_1,\dots,\hat{\boldsymbol{\mu}}_K$ for $K > C/\epsilon^1_{\mathrm{tol}}$; thus the algorithm must terminate for $1 \le K \le C/\epsilon^1_{\mathrm{tol}}$ subdomains. □

Remark 5 (Convergence in the case $\bar{N} \ge 1$, $P \ge 1$). We first recall a polynomial approximation result. Consider approximation of a sufficiently smooth function on a bounded domain in $\mathbb{R}^P$ by piecewise polynomial interpolation of order p over K subdomains: we expect the error to decrease as $(1/K)^{(p+1)/P}$, or as $(1/K)^{(\bar{N}^{1/P})/P}$

if we associate $\bar{N} = (p+1)^P$ degrees of freedom to each subdomain (consistent with our earlier definition of “order”).

In the “zeroth-order” multi-parameter case ($\bar{N} = 1$, $P > 1$) we expect (but do not prove) that our method converges for

$$K < \max\left\{1, \frac{C}{(\epsilon^1_{\rm tol})^P}\right\} \tag{4.38}$$

subdomains for some positive constant C. This poor convergence for $P \gg 1$ suggests the advantage and indeed necessity of “*p*”–convergence [25] or “*hp*”–convergence rather than solely “*h*”-convergence. Next, in the higher order, one-parameter case ($\bar{N} > 1$, $P = 1$), we might expect convergence

$$K < \max\left\{1, \frac{C}{(\epsilon^1_{\rm tol})^{\frac{1}{\bar{N}}}}\right\} \tag{4.39}$$

for some positive constant C. Finally, in the *general* case $\bar{N} \geq 1$, $P \geq 1$, we might expect convergence

$$K < \max\left\{1, \frac{C}{(\epsilon^1_{\rm tol})^{\frac{P}{\bar{N}^{1/P}}}}\right\}. \tag{4.40}$$

We shall consider these heuristic arguments again in the context of numerical results.

Note that our bound (4.28) and estimates (4.39) and (4.40) should capture the correct order but of course the constant will be very pessimistic: by design, the Greedy should adapt the sample to best accomodate local variations. ♢

Remark 6 (A Sharper Convergence Result). We show in Appendix A that the constant C in Proposition 4.1 can be improved to $C_{\rm im} = 2\tilde{C}|\mathcal{D}|(1 + \overline{\gamma}/\underline{\alpha})$ by an approach that, instead of (4.31)–(4.33), considers directly the equation for the Riesz representation of the residual. However, the approach above is more general and applicable in other (e.g. interpolation [9]) contexts. ♢

5. The “*hp*”-type Reduced Basis Method. With the “*hp*”-type RB method, we combine the “*h*”- and “*p*”-type methods: we first construct a partition of the parameter domain with “*h*”-refinement; we then compute independent approximation spaces restricted to each parameter subdomain with “*p*”-refinement—in general, the approximation spaces will have different dimensions.

5.1. Approximation. The parameter domain partition is first constructed by an $\bar{N} = 1$ “*h*”-type approximation for a prescribed error bound tolerance $\epsilon^1_{\rm tol}$. We first construct the initial train sample $\Xi_{(1)} \subset \mathcal{D}$, choose an initial parameter anchor $\hat{\boldsymbol{\mu}}_{(1)} \in \mathcal{D}$, and specify $\epsilon^1_{\rm tol}$; we then execute Algorithm 3, $h\mathrm{RB}(\Xi_{(1)}, \hat{\boldsymbol{\mu}}_{(1)}, 1, \epsilon^1_{\rm tol})$ for $1 \leq k \leq K$. The output from $h\mathrm{RB}(\Xi_{(1)}, \hat{\boldsymbol{\mu}}_{(1)}, 1, \epsilon^1_{\rm tol})$ is K subdomains associated with the K leaf nodes of the binary tree. Each subdomain has an associated one-parameter model and a one-dimensional approximation space; we denote by $B^1, \ldots, B^K$ the K associated Boolean indices. We also store the train sample over each of the final subdomains.

We now append additional basis functions to each approximation space by a standard “*p*”-type procedure over each train sample: we specify the maximum RB space dimension $\tilde{N}_{\max,B^k} = \tilde{N}_{\max} > \bar{N}$, $1 \leq k \leq K$; we specify a new error bound tolerance $\epsilon^2_{\rm tol} < \epsilon^1_{\rm tol}$; for $1 \leq k \leq K$, we then execute Algorithm 2, $\mathrm{Greedy}^2(\Xi_{B^k}, \boldsymbol{\mu}_{1,B^k}, \epsilon^2_{\rm tol}, \tilde{N}_{\max,B^k})$.

(Note that we must replace the "h"-type error bound $\Delta_N^{h\mathrm{RB}}$ in Algorithm 2 by the "hp"-type error bound $\Delta_N^{hp\mathrm{RB}}$, which we introduce shortly.)

The final output is thus K RB approximation spaces $X_{N_{\max},B^k}$ and associated models $\mathcal{M}_{N_{\max},B^k}$, $1 \le k \le K$. Note that $N_{\max,B^k}$ is in general different for different k since the error bound tolerance $\epsilon_{\mathrm{tol}}^2$ might be satisfied by the different approximation spaces over the different train samples with different numbers of basis functions; we define in the "hp" case $N_{\max} = \max_{k=1,\dots,K} N_{\max,B^k}$.

Finally, we may now define the "hp"-type RB approximation. First, given any $\boldsymbol{\mu} \in \mathcal{D}$, we determine the subdomain $\mathcal{V}_{B_l^*}$ containing $\boldsymbol{\mu}$ from the search (4.13). Then, given $1 \le N \le N_{\max}$, we find $u_N^{hp\mathrm{RB}}(\boldsymbol{\mu}) \in X_{N^*,B_l^*}$ such that

$$a(u_N^{hp\mathrm{RB}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in X_{N^*,B_l^*}, \tag{5.1}$$

where $N^* \equiv \min\{N, N_{\max,B_l^*}\}$. (Note that B_l^* and thus N^* depend on $\boldsymbol{\mu}$.)

5.2. *A Posteriori* **Error Estimation.** We shall require some new notation for the "hp"-type *a posteriori* error bound.

Given any $\boldsymbol{\mu} \in \mathcal{D}$ and a partition of $\mathcal{D}$ into subdomains, we determine B_l^* from the binary search (4.13) and compute the RB solution $u_N^{hp\mathrm{RB}}(\boldsymbol{\mu})$ from (5.1). The RB residual is

$$r_N^{hp\mathrm{RB}}(v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) - a(u_N^{hp\mathrm{RB}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}), \quad \forall v \in X; \tag{5.2}$$

the Riesz representation of the residual is denoted by $\mathcal{R}_N^{hp\mathrm{RB}}(\boldsymbol{\mu})$. Our upper bound for the X-norm error $\|u(\boldsymbol{\mu}) - u_N^{hp\mathrm{RB}}(\boldsymbol{\mu})\|_X$ is then given by

$$\Delta_N^{hp\mathrm{RB}}(\boldsymbol{\mu}) \equiv \frac{\|\mathcal{R}_N^{hp\mathrm{RB}}(\boldsymbol{\mu})\|_X}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})}. \tag{5.3}$$

Lemma 3.1 now directly applies with an appropriate change of notation.

5.3. Offline-Online Decomposition. In the *offline* stage, we determine the partition of the parameter domain and construct the corresponding RB models and spaces as discussed above. For the purposes of this subsection, we assume a perfect binary tree. We also assume that the cardinality of the train sample over each of the subdomains is equal to n_{train}. It is crucial to note that, since the initial "h"-refinement is performed for $\bar{N} = 1$, we re-use all computed entities in the later "p"-type stage, and there is thus only minor additional cost associated with the partition procedure.

The offline cost derives from several components:

1. Snapshot Truth Solves. We must compute (at most) $N_{\max}K$ snapshots associated with the partition *and* final approximation spaces.

2. Reduced Basis Preprocessing. We must compute (at most) $K(Q_a N_{\max}^2 + Q_f N_{\max})$ truth inner products to form the parameter-independent "stiffness" matrices and loads (e.g., as in (3.12)) for the partition and final models.

3. Error Bound Preprocessing. We must compute (at most) $KN_{\max}Q_a + Q_f$ truth Poisson solves of the form (3.18) for the partition and final models. We must also compute (at most) $K(N_{\max}Q_a + Q_f)^2$ truth inner products of the form (3.21) related to the dual norm of the residual associated with the partition and final models.

4. Error Bound Evaluations. We must solve $n_{\mathrm{train}}(K-1)$ RB systems (of size $\bar{N} = 1$) to obtain the residual coefficients and evaluate $n_{\mathrm{train}}(K-1)$ RB error bounds (for $\bar{N} = 1$) during the Greedy^2 sampling for the "h"-refinement partition process. We

must also solve (at most) $n_{\text{train}}N_{\max}K$ RB systems to obtain the residual coefficients and evaluate $n_{\text{train}}N_{\max}K$ RB error bounds during the Greedy^2 sampling for the final models. This results, to leading order, in $\mathcal{O}(n_{\text{train}}N_{\max}K(N_{\max}^3 + N_{\max}^2Q^2))$ operations in total.

The combined (1-4) offline cost is thus $N_{\max}K$ truth snapshot computations, $N_{\max}KQ_a + Q_f$ truth Poisson solves, $K(Q_aN_{\max}^2 + Q_fN_{\max}) + K(N_{\max}Q_a + Q_f)^2$ truth inner products, and $\mathcal{O}(n_{\text{train}}N_{\max}K(N_{\max}^3 + N_{\max}^2Q^2))$ operations to evaluate the error bounds.

For each model, we must construct and retain the parameter-independent data necessary to accomodate the efficient evaluation stage for the RB approximation and the associated *a posteriori* error bound, as discussed in §3.3 for the standard RB method. The online (permanent) storage requirement associated with the RB approximation is Q_aK matrices of maximum size $N_{\max} \times N_{\max}$ and Q_f vectors of maximum size $N_{\max}$; the online storage associated with the RB error bounds is $K(N_{\max}Q_a + Q_f)^2/2$.

In the *online* stage, given any $\boldsymbol{\mu} \in \mathcal{D}$, we first determine the subdomain containing $\boldsymbol{\mu}$ via the binary search (4.13) in $\mathcal{O}(\log_2 K)$ operations. (Recall that we presume here a perfect binary tree.) Thanks to the construction-evaluation decomposition, we can then, given $1 \le N \le N_{\max}$, assemble and solve the corresponding system of algebraic equations in $\mathcal{O}(QN^2)$ and $\mathcal{O}(N^3)$ operations, respectively, and compute the associated RB error bound in $\mathcal{O}(N^2Q^2)$ operations.

6. A Convection-Diffusion Model Problem.

6.1. Formulation and Truth Discretization.

We now apply the "*p*"-, "*h*"- and "*hp*"-type RB methods to a steady convection-diffusion model problem parametrized by the angle and magnitude of the prescribed velocity field: Let $\boldsymbol{\mu} = (\mu_1, \mu_2)$ and define $\mathbf{V}(\boldsymbol{\mu}) = [\mu_2 \cos\mu_1, \mu_2 \sin\mu_1]^{\mathrm{T}}$. The governing equations for the exact field variable $u^e(\boldsymbol{\mu})$ are

$$-\nabla^2 u^e(\boldsymbol{\mu}) + \mathbf{V}(\boldsymbol{\mu}) \cdot \nabla u^e(\boldsymbol{\mu}) = 10 \quad \text{in } \Omega, \tag{6.1}$$

$$u^e(\boldsymbol{\mu}) = 0 \quad \text{on } \partial\Omega. \tag{6.2}$$

The physical domain is $\Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \le 2\}$ and $\partial\Omega$ is the boundary of Ω.

We next define for all $w, v \in X^e \equiv X^e(\Omega) \equiv H_0^1(\Omega)$ the parametrized bilinear form

$$\begin{aligned} a(w, v; \boldsymbol{\mu}) &\equiv \int_\Omega \nabla w \cdot \nabla v \, \mathrm{d}\Omega + \int_\Omega \big(\mathbf{V}(\boldsymbol{\mu}) \cdot \nabla w\big) v \, \mathrm{d}\Omega \\ &\equiv \int_\Omega \nabla w \cdot \nabla v \, \mathrm{d}\Omega + \mu_2 \cos\mu_1 \int_\Omega \frac{\partial w}{\partial x} v \, \mathrm{d}\Omega + \mu_2 \sin\mu_1 \int_\Omega \frac{\partial w}{\partial y} v \, \mathrm{d}\Omega, \end{aligned} \tag{6.3}$$

and the linear functional

$$f(v) \equiv f(v; \boldsymbol{\mu}) \equiv 10 \int_\Omega v \, \mathrm{d}\Omega; \tag{6.4}$$

thus (2.1) obtains for $Q_a = 3$ and $Q_f = 1$. We can then state the exact problem in the standard variational form: Given any $\boldsymbol{\mu} \in \mathcal{D}$, find $u^e \in X^e$ such that

$$a(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \quad \forall v \in X^e. \tag{6.5}$$

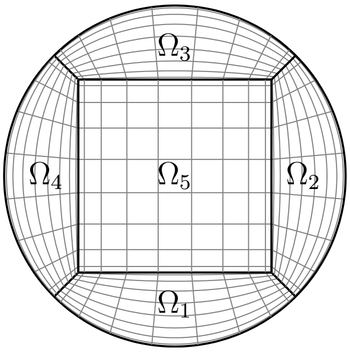

FIGURE 6.1. *The circular physical domain partitioned into five spectral elements.*

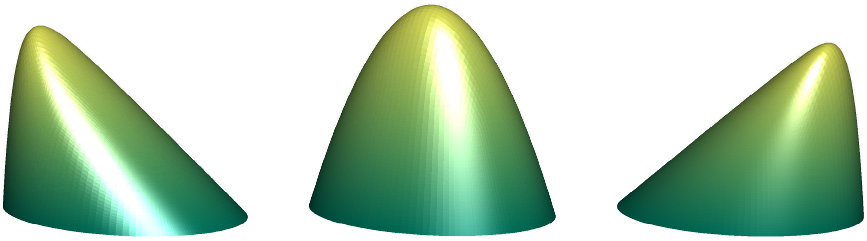

FIGURE 6.2. *Solutions to* (6.7) *for different parameter values* $\boldsymbol{\mu} = (\pi, 10)$ *(left),* $\boldsymbol{\mu} = (0, 0)$ *(middle), and* $\boldsymbol{\mu} = (0, 10)$ *(right).*

Note that for this particular problem, $a_s(w, v; \boldsymbol{\mu}) = \int_\Omega \nabla w \cdot \nabla v \, \mathrm{d}\Omega$ is parameter-independent; thus $a(v, v; \boldsymbol{\mu}) = \|v\|_X^2$ for all $v \in X^e$ and we may choose $\alpha_{\mathrm{LB}} \equiv 1$ as the coercivity lower bound.

Next, we introduce a truth spectral element space $X \equiv X^{\mathcal{N}}(\Omega) \subset X^e(\Omega)$ of dimension $\mathcal{N} = 481$ based on five spectral elements of order ten: we introduce a computational domain $\hat{\Omega} = (-1, 1)^2$ and standard transfinite mappings $\mathcal{F}_i : \hat{\Omega} \to \Omega_i$, $1 \le i \le 5$, [12]; we then define

$$X \equiv X^{\mathcal{N}}(\Omega) = \{v \in H_0^1(\Omega) : v|_{\Omega_i} \circ \mathcal{F}_i \in \mathbb{P}^{10}(\hat{\Omega}), 1 \le i \le 5\}, \tag{6.6}$$

where $\mathbb{P}^{10}(\hat{\Omega})$ denotes the space of polynomials of degree 10 (in each spatial direction) over $\hat{\Omega}$. The truth discretization of (6.5) reads: Given any $\boldsymbol{\mu} \in \mathcal{D}$, find $u(\boldsymbol{\mu}) \in X$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \quad \forall v \in X. \tag{6.7}$$

In Figure 6.2, we plot the solution of (6.7) for three different parameter values. Clearly, the three solutions have a very different structure—this particular problem is thus a good candidate for "*hp*" treatment.

We define three parameter domains,

$$\mathcal{D}_{\mathrm{I}} \equiv \{0\} \times [0, 10], \quad \mathcal{D}_{\mathrm{II}} \equiv [0, \pi] \times \{10\}, \quad \mathcal{D}_{\mathrm{III}} \equiv [0, \pi] \times [0, 10]; \tag{6.8}$$

we shall thus consider $P = 1$ ($\mathcal{D}_{\mathrm{I}}$ or $\mathcal{D}_{\mathrm{II}}$) or $P = 2$ ($\mathcal{D}_{\mathrm{III}}$) parameters.

Admittedly, the computational benefit of the RB approximation for this particular problem is minimal since the truth approximation space is of rather low dimension

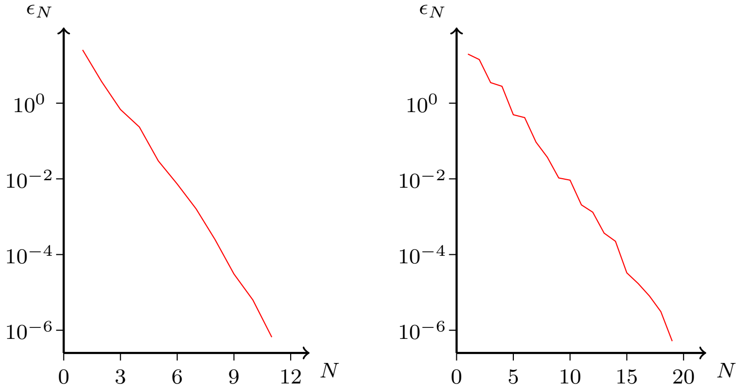

FIGURE 6.3. *Standard RB ("p"-type) convergence results: ϵ_N as a function of N for the one-parameter cases $\mathcal{D} = \mathcal{D}_{\text{I}}$ (left) and $\mathcal{D} = \mathcal{D}_{\text{II}}$ (right).*

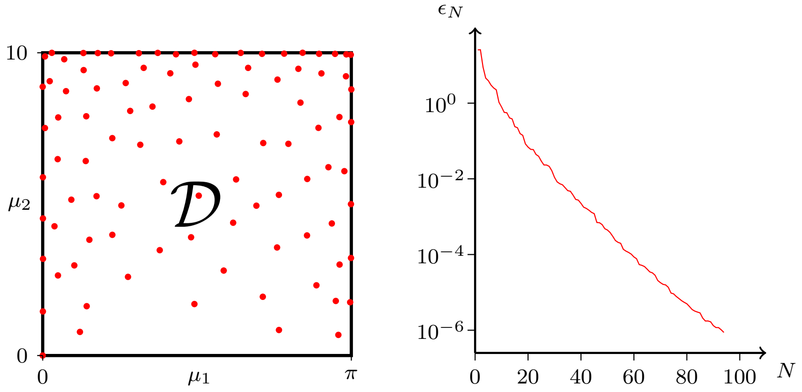

FIGURE 6.4. *Greedy parameter choices (left) and associated standard RB ("p"-type) convergence results (right) for the two-parameter case $\mathcal{D} = \mathcal{D}_{\text{III}}$.*

$\mathcal{N} = 481$. However, our problem here is a good vehicle for exposition of the "p"-type, "h"-type, and "hp"-type RB methods, and is amendable to extensive theoretical and experimental analysis. In [10] we apply the "hp"-type RB method to problems that require and demonstrate the speedup provided by the RB approximation.

6.2. "p"-type RB Approximation Results. In this section, we present the standard ("p"-type) RB convergence results for our model problem.

We introduce uniformly distributed random train samples $\Xi_{\text{I}} \subset \mathcal{D}_{\text{I}}$, $\Xi_{\text{II}} \subset \mathcal{D}_{\text{II}}$, and $\Xi_{\text{III}} \subset \mathcal{D}_{\text{III}}$ of size 10^3, 10^3, and 10^4, respectively. We recall that $\epsilon_N = \max_{\boldsymbol{\mu} \in \Xi} \Delta_N(\boldsymbol{\mu})$ is the maximum X-norm error bound over the train sample associated with the space X_N. In Figure 6.3 we plot ϵ_N as a function of N for the two one-parameter cases $\mathcal{D} = \mathcal{D}_{\text{I}}$ and $\mathcal{D} = \mathcal{D}_{\text{II}}$: we note that N can be quite small even for $\epsilon_N \approx 10^{-6}$. In Figure 6.4 (right) we plot ϵ_N for the two-parameter case $\mathcal{D} = \mathcal{D}_{\text{III}}$. The quite poor convergence of the "p"-type RB is not surprising given the very different solution structures obtained for different parameter values; variations in μ_1 are particularly difficult to resolve—as indicated by the slower convergence for the case $\mathcal{D} = \mathcal{D}_{\text{II}}$ in Figure 6.3 (right)—due to the "movement" of the boundary layer. In Figure 6.4 (left) we present the parameters chosen by the Greedy[1] algorithm: the points are clearly

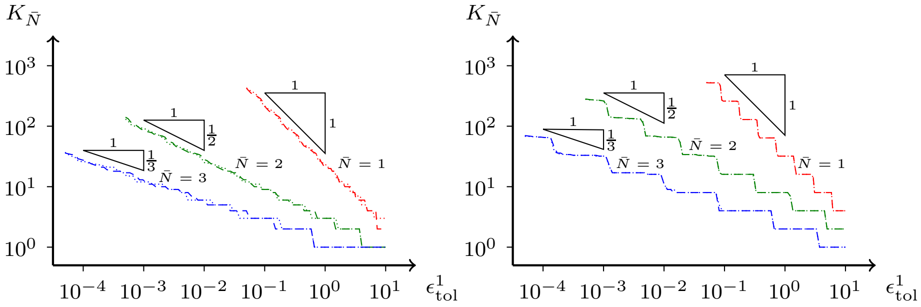

FIGURE 6.5. *"h"-type RB convergence results:* $K_{\bar{N}}(\epsilon^1_{\rm tol})$ *for* $\bar{N} = 1$, $\bar{N} = 2$ *and* $\bar{N} = 3$ *for the one-parameter cases* $\mathcal{D} = \mathcal{D}_{\rm I}$ *(left) and* $\mathcal{D} = \mathcal{D}_{\rm II}$ *(right). Both Euclidean distance (dotted lines) and the* a posteriori *error bound (dashed lines) are considered for the proximity function.*

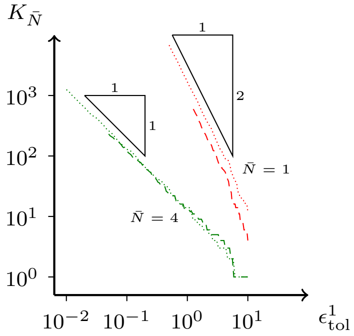

FIGURE 6.6. *"h"-type RB convergence results:* $K_{\bar{N}}(\epsilon^1_{\rm tol})$ *for* $\bar{N} = 1$ *and* $\bar{N} = 4$ *for the two-parameter case* $\mathcal{D} = \mathcal{D}_{\rm III}$. *Both Euclidean distance (dotted lines) and the* a posteriori *error bound (dashed lines) are considered for the proximity function.*

denser for larger velocities—which yield thinner boundary layers.

6.3. "h"-type RB Approximation Results. We now present convergence results for equi-order "h"-type RB approximation; the dimension of the approximation spaces is thus fixed. The convergence results are obtained by first specifying the desired tolerance $\epsilon^1_{\rm tol}$ as well as the RB space dimension $\bar{N}$, the initial train sample $\Xi_{(1)}$, and the initial anchor point $\hat{\boldsymbol{\mu}}_{(1)}$; we then perform hRB$(\Xi_{(1)}, \hat{\boldsymbol{\mu}}_{(1)}, \bar{N}, \epsilon^1_{\rm tol})$. Given $\bar{N}$, we let $K_{\bar{N}}(\epsilon^1_{\rm tol})$ denote the number of subdomains in the partition for specified $\epsilon^1_{\rm tol}$.

We start with the one-parameter cases $\mathcal{D} = \mathcal{D}_{\rm I}$ and $\mathcal{D} = \mathcal{D}_{\rm II}$. In both cases, the initial train samples consist of 100 random points, and the initial anchor point is $\hat{\boldsymbol{\mu}}_{(1)} = (0, 0)$. In Figure 6.5 we present $K_{\bar{N}}(\epsilon^1_{\rm tol})$ for $\bar{N} = 1, 2, 3$ for each of the two cases. The proximity function is either $d_{B_l}(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{B_l}\|_2$ (dotted lines) or $d_{B_l}(\boldsymbol{\mu}) = \Delta_{\bar{N}=1,B_l}(\boldsymbol{\mu})$ (dashed lines): we observe that the choice of the proximity function has little impact on the results. We indicate the slopes for first, second and third order convergence: for the $\bar{N} = 1$ approximation, the convergence rates are in good agreement with the theoretical result (4.28); for the $\bar{N} > 1$ approximations, the convergence is approximately $\bar{N}$th order and hence in agreement with our conjecture (4.38). (We recall that, here, a steeper slope implies slower convergence.)

We next consider the two-parameter case $\mathcal{D} = \mathcal{D}_{\rm III}$. The initial train sample $\Xi_{(1)}$ consist of 10^3 random points, and the initial anchor point is $\hat{\boldsymbol{\mu}}_{(1)} = (0, 0)$. In Figure 6.6 we present $K_{\bar{N}}(\epsilon^1_{\rm tol})$ for $\bar{N} = 1$ and $\bar{N} = 4$. The proximity function is either

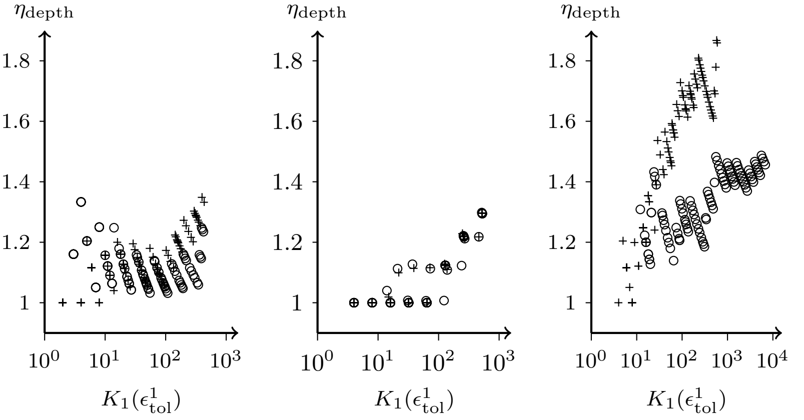

FIGURE 6.7. *Relative tree depths $\eta_{\rm depth}$ as functions of the number of subdomains (leaf nodes) $K_{\bar N}(\epsilon^1_{\rm tol})$ for the $\bar N = 1$ "h"-type approximation for each parametrization $\mathcal{D} = \mathcal{D}_{\rm I}$ (left), $\mathcal{D} = \mathcal{D}_{\rm II}$ (middle), and $\mathcal{D} = \mathcal{D}_{\rm III}$ (right). Both Euclidean distance (○) and the* a posteriori *error bound (+) are considered for the proximity function.*

$d_{B_l}(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{B_l}\|_2$ (dotted lines) or $d_{B_l}(\boldsymbol{\mu}) = \Delta_{\bar N=1,B_l}(\boldsymbol{\mu})$ (dashed lines): now the choice of the proximity function has some, but very slight, impact on the results but only for the $\bar N = 1$ approximation. It is clear from the slopes provided that we achieve roughly $K_1 \sim (\epsilon^1_{\rm tol})^{-2}$ and $K_4 \sim (\epsilon^1_{\rm tol})^{-1}$ as expected from our conjectures (4.39) and (4.40).

Finally, we empirically examine the depth of the associated binary trees. Ideally, we would like the relative tree depth (4.14) to be a constant close to unity; for K subdomains the search (4.13) is in this case an efficient $\log_2 K$ operations binary search. In Figure 6.7 we plot the relative tree depth against the number of subdomains for the $\bar N = 1$ approximation for each of our three parametrizations. (Note the scatter in the plots is induced by the range of $\epsilon^1_{\rm tol}$ considered.) Although from these results it is difficult to reach general conclusions, the relative tree depths are all fairly close to unity and increase with increasing K only very modestly even for $1 \le K \le 10^4$.

6.4. "*hp*"-type Approximation Results. We now present convergence results for an "*hp*"-type RB approximation. For a partition with K subdomains, let $\overline{\Xi}$ denote the union of the associated K train samples; we then define $\epsilon_N^{hp{\rm RB}} \equiv \max_{\boldsymbol{\mu}\in\overline{\Xi}} \Delta_N^{hp{\rm RB}}(\boldsymbol{\mu})$.

We start with the one-parameter cases $\mathcal{D} = \mathcal{D}_{\rm I}$ and $\mathcal{D} = \mathcal{D}_{\rm II}$. The initial train sample consists of 100 random points, and the initial anchor point is $\hat{\boldsymbol{\mu}}_{(1)} = (0,0)$. We use $d_{B_l}(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{B_l}\|_2$ as the proximity function. For the case $\mathcal{D} = \mathcal{D}_{\rm I}$, we specify $\epsilon^1_{\rm tol} = 5$ and $\epsilon^1_{\rm tol} = 0.1$, for which we obtain $K = 4$ and $K = 211$ subdomains, respectively; for the case $\mathcal{D} = \mathcal{D}_{\rm II}$, we specify $\epsilon^1_{\rm tol} = 5$ and $\epsilon^1_{\rm tol} = 0.1$, for which we obtain $K = 8$ and $K = 260$ subdomains, respectively. In Figure 6.8 we plot $\epsilon_N^{hp{\rm RB}}$ as functions of N for each of the two parametrizations. *Given* any error bound tolerance, we note a significant reduction in the required approximation space dimension (in any subdomain) when compared to a standard RB ($K = 1$) approximation. Of course, the *total* number of snapshots NK (for any given tolerance) will *increase* with K: greater suitability of local snapshots does not compensate for lower order in terms of global approximation properties.

We next consider the two-parameter case $\mathcal{D} = \mathcal{D}_{\rm III}$. We use $d_{B_l}(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{B_l}\|_2$

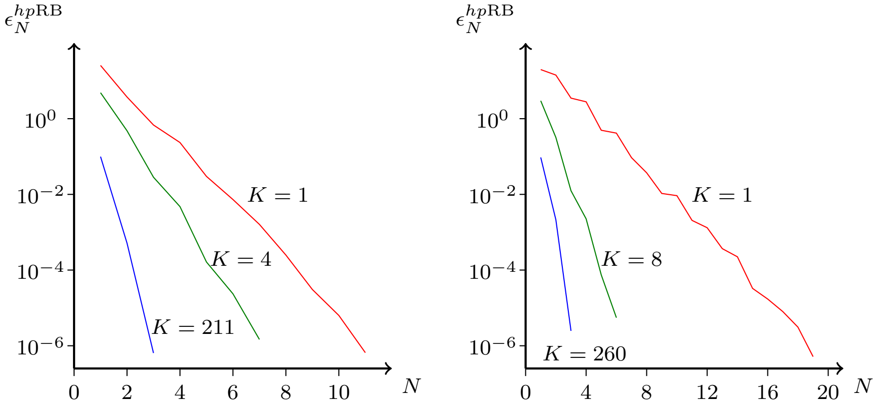

FIGURE 6.8. *"hp"-type RB convergence results: $\epsilon_N^{hp\mathrm{RB}}$ as a function of N for the one-parameter cases $\mathcal{D} = \mathcal{D}_{\mathrm{I}}$ (left) and $\mathcal{D} = \mathcal{D}_{\mathrm{II}}$ (right).*

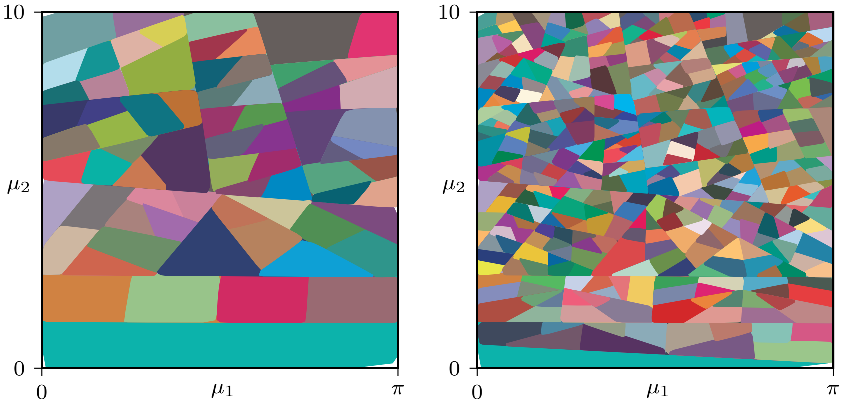

FIGURE 6.9. *Parameter domain partitions for the case $\mathcal{D} = \mathcal{D}_{\mathrm{III}}$. The number of subdomains is $K(\epsilon^1_{\mathrm{tol}}) = 72$ for $\epsilon^1_{\mathrm{tol}} = 5$ (left) and $K(\epsilon^1_{\mathrm{tol}}) = 417$ for $\epsilon^1_{\mathrm{tol}} = 2$ (right).*

as the proximity function. The initial train sample consist of 10^3 random points, and the initial anchor point is $\hat{\boldsymbol{\mu}}_{(1)} = (0, 0)$. In Figure 6.9 we show partitions of the parameter domain for specified $\epsilon^1_{\mathrm{tol}} = 5$ and $\epsilon^1_{\mathrm{tol}} = 2$, for which we obtain $K = 72$ and $K = 417$ subdomains, respectively. We note—similarly to the "p"-type greedy parameter choices in Figure 6.3 (left)—that the subdomains are smaller for larger velocities. In Figure 6.10, we plot for each of the two partitions in Figure 6.9 the maximum error bound $\epsilon_N^{hp\mathrm{RB}}$ as a function of N; we include the results for the standard RB approximation ("p"-type or "hp"-type with $K = 1$) as well. Again, the local order reduction is significant.

In Table 6.1 and Table 6.2 we summarize for $K = 72$ and $K = 417$ subdomains, respectively, the offline and online performance of the "hp" approach *relative* to that of the standard RB method. For given tolerances $\epsilon^2_{\mathrm{tol}}$, we report in the four rows of the tables the relative number of truth solves, the relative number of operations for online evaluation of the RB approximation, the relative number of operations for online evaluation of the RB error bound, and relative online storage, respectively. The reported values are based on the theoretical operation count and storage, which we recall here. For N basis functions and $K \geq 1$ subdomains the number of offline

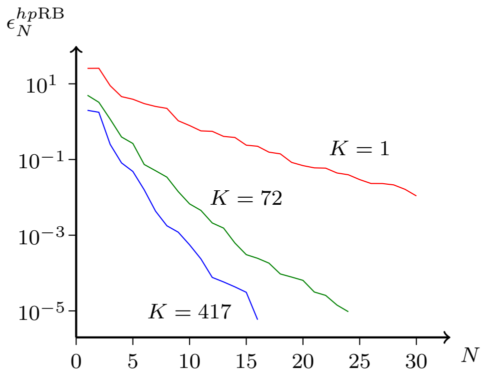

FIGURE 6.10. *Convergence of "hp"-type RB for the two-parameter case $\mathcal{D} = \mathcal{D}_{\mathrm{III}}$.*

TABLE 6.1
Operation count and storage requirement for the "hp"-type RB with $K = 72$ relative to that of the standard RB ($K = 1$) for the two-parameter case $\mathcal{D} = \mathcal{D}_{\mathrm{III}}$.

	$\epsilon_{\mathrm{tol}}^2 = 10^{-2}$	$\epsilon_{\mathrm{tol}}^2 = 10^{-3}$	$\epsilon_{\mathrm{tol}}^2 = 10^{-4}$
Offline Truth Solve Relative Cost	1.66E+1	1.57E+1	1.59E+1
Online RB Solution Relative Cost	3.35E−2	2.82E−2	2.70E−2
Online RB Error Bound Relative Cost	1.09E−1	9.57E−2	9.23E−2
Online Relative Storage	3.96E+0	3.51E+0	3.58E+0

truth solves is $KN(1 + Q_a) + Q_f$.[2] (We assume roughly equal computation times for the convection–diffusion and Poisson solves.) The online operation count is roughly $\mathcal{O}(N^3)$ for the RB solution and $\mathcal{O}((Q_aN + Q_f)^2)$ for the RB error bound; we neglect the $\mathcal{O}(QN^2)$ cost of forming the RB system and the $\mathcal{O}(\log_2 K)$ cost of finding the correct subdomain via the binary search. The online (permanent) storage requirement is dominated by the $\mathcal{O}(KQ^2N^2)$ data required for the RB error bounds.

Admittedly, the "*hp*" approach requires more truth solves—a larger offline cost—than the standard method. However, the online computational savings are significant: in our example with $K = 72$ subdomains the online cost relative to that of the standard RB method is about three percent for the RB solution and about ten percent for the RB error bound; in our example with $K = 417$ subdomains the online relative cost is only about one percent for the RB solution and about five percent for the RB error bound. (Typically the contributions to the total online cost from RB solution and RB error bound computation are comparable.) The online storage requirement is somewhat larger with the "*hp*" approach, though in general the storage requirements are quite modest—the N^2 scaling moderates the growth due to K.

7. Concluding Remarks. The "*hp*"-type RB method has been shown to significantly reduce the online computational cost. On the other hand, the new approach is more expensive than the standard ("*p*"-type) RB method in the offline stage. Hence we must trade offline cost for online performance. The online effort is often our main concern in the real-time or many-query contexts.

[2] We assume that the truth solves (including both snapshot computation and the Poisson solves related to the dual norm of the residual) constitute the most expensive part of the total offline cost. In fact, this assumption *favors* the standard "*p*"-type RB since the error bound sampling is superlinear in N and thus scales more advantageously for the "*hp*"-approach. We thus expect in particular for n_{train} large that the (total) offline relative cost will be lower than reported in the tables.

Table 6.2

Operation count and storage requirement for the "hp"-type RB with $K = 417$ relative to that of the standard RB ($K = 1$) for the two-parameter case $\mathcal{D} = \mathcal{D}_{\mathrm{III}}$.

	$\epsilon^2_{\mathrm{tol}} = 10^{-2}$	$\epsilon^2_{\mathrm{tol}} = 10^{-3}$	$\epsilon^2_{\mathrm{tol}} = 10^{-4}$
Offline Truth Solve Relative Cost	6.58E+1	6.19E+1	6.33E+1
Online RB Solution Relative Cost	1.15E−2	1.03E−2	8.00E−3
Online RB Error Bound Relative Cost	5.48E−2	4.97E−2	4.18E−2
Online Relative Storage	1.08E+1	9.39E+0	9.81E+0

We expect the new approach to be particularly beneficial for problems for which the solution structure is very different in different parts of the parameter domain. While our model problem is specifically constructed to exhibit this property, there are many realistic problems which exhibit similar behavior. As an example, we mention an application of RB to the solution of the Fokker-Planck equation [18]; here, the solution is required for many different parameter values, but the required ("*p*"-type) RB spaces are rather large. Also of interest are problems which exhibit non-smooth parameter dependence—the "*hp*"-approach should automatically refine the parameter domain around singularities and hence perform better than the standard approach;

There are several opportunities for extensions. First, we can generalize our approach to POD-Greedy sampling [16] for parabolic problems [10, 11]: the critical new ingredient is proper balance between additional POD modes and additional Greedy parameter values in the initial subdivision process. Second, we can extend the approach to quadratically nonlinear problems such as the incompressible Navier-Stokes equations [10]—in this case the "*hp*"-approach is particularly advantageous since the (online) computation of the error bound requires $\mathcal{O}(N^4)$ operations for N basis functions and hence the "smaller N for larger K" trade is particularly favorable. Third, we can consider, in the offline stage, a parallel "*hp*" approach—we can subdivide the parameter domain along each branch of the associated binary tree independently and hence concurrently.

Acknowledgements. We acknowledge helpful discussions with D. Knezevic, N. C. Nguyen, D. B. P. Huynh, S. Boyaval, and B. Haasdonk during this work. We also thank D. Knezevic specifically for contributions to the alternative Proposition 4.1 proof given in Appendix A.

Appendix A. An improvement of Proposition 4.1. Let $\boldsymbol{\mu}^* \in \mathcal{D}$ be such that the RB error bound $\Delta_N(\boldsymbol{\mu}^*) = 0$; hence $\boldsymbol{\mu}^*$ corresponds to a parameter value associated with a truth snapshot residing in the RB space. In this case the residual satisfies $r_N(v; \boldsymbol{\mu}^*) = 0$ for all $v \in X^{\mathcal{N}}$. We now consider any $\boldsymbol{\mu} \in \mathcal{D}$. The Riesz representation of the residual at $\boldsymbol{\mu}$, $\mathcal{R}_N(\boldsymbol{\mu}) \in X^{\mathcal{N}}$, satisfies

$$(\mathcal{R}_N(\boldsymbol{\mu}), v)_X = r_N(v; \boldsymbol{\mu}) = r_N(v; \boldsymbol{\mu}) - r_N(v; \boldsymbol{\mu}^*), \quad \forall v \in X^{\mathcal{N}}. \tag{A.1}$$

By the definition of the residual and the triangle inequality we obtain

$$\begin{aligned} |(\mathcal{R}_N(\boldsymbol{\mu}), v)_X| \leq{}& |f(v; \boldsymbol{\mu}) - f(v; \boldsymbol{\mu}^*)| \\ &+ |a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu}^*) - a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu})|, \quad \forall v \in X^{\mathcal{N}}. \end{aligned} \tag{A.2}$$

For the first term on the right of (A.2) we invoke (2.19) to obtain

$$|f(v; \boldsymbol{\mu}) - f(v; \boldsymbol{\mu}^*)| \leq \tilde{c}_2 \|v\|_X \, |\boldsymbol{\mu} - \boldsymbol{\mu}^*|, \quad \forall v \in X^{\mathcal{N}}. \tag{A.3}$$

For the second term on the right hand side of (A.2) we note that

$$
\begin{aligned}
&|a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu}^*) - a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu})| \\
&\quad = \big|a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu}^*) - a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}) + \big(a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu})\big)\big| \\
&\quad \le |a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu}^*) - a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu})| + |a(u_N(\boldsymbol{\mu}^*) - u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu})|\,, \quad \forall v \in X^{\mathcal{N}}.
\end{aligned}
\tag{A.4}
$$

(Note that the term in parenthesis on the second line is equal to zero.)

For the first term on the right-hand side of (A.4) we invoke (2.18) and the Lax-Milgram Lemma to obtain

$$
\begin{aligned}
|a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu}^*) - a(u_N(\boldsymbol{\mu}^*), v; \boldsymbol{\mu})| &\le \tilde{c}_1 \|u_N(\boldsymbol{\mu}^*)\|_X \|v\|_X \,|\boldsymbol{\mu} - \boldsymbol{\mu}^*| \\
&\le \frac{\tilde{c}_1}{\underline{\alpha}} \|f(\cdot; \boldsymbol{\mu}^*)\|_{X'} \|v\|_X \,|\boldsymbol{\mu} - \boldsymbol{\mu}^*| \\
&\le \frac{\tilde{c}_1}{\underline{\alpha}} \max_{\boldsymbol{\mu} \in \mathcal{D}} \|f(\cdot; \boldsymbol{\mu})\|_{X'} \|v\|_X \,|\boldsymbol{\mu} - \boldsymbol{\mu}^*|\,,
\end{aligned}
\tag{A.5}
$$

for all $v \in X^{\mathcal{N}}$. For the second term on the right hand side of (A.4) we invoke continuity of $a(\cdot, \cdot; \boldsymbol{\mu})$ and (2.9), and then Lemma 2.1 to obtain

$$
\begin{aligned}
|a(u_N(\boldsymbol{\mu}^*) - u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu})| &\le \overline{\gamma} \|u_N(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}^*)\|_X \|v\|_X \\
&\le \tilde{C} \overline{\gamma} |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \|v\|_X,
\end{aligned}
\tag{A.6}
$$

for all $v \in X^{\mathcal{N}}$. We now combine (A.2) with (A.3), (A.4), (A.5), and (A.6) to obtain

$$
(\mathcal{R}_N(\boldsymbol{\mu}), v)_X \le |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \, \|v\|_X \left(\tilde{c}_2 + \max_{\boldsymbol{\mu} \in \mathcal{D}} \tilde{c}_1 \frac{\|f(\cdot; \boldsymbol{\mu})\|_{X'}}{\bar{\alpha}} + \tilde{C} \overline{\gamma} \right),
\tag{A.7}
$$

and hence with $v = \mathcal{R}_N(\boldsymbol{\mu})$ and the expression for $\tilde{C}$ from Lemma 2.1,

$$
\|\mathcal{R}_N(\boldsymbol{\mu})\|_X \le |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \, \tilde{C}(\underline{\alpha} + \overline{\gamma}).
\tag{A.8}
$$

We now invoke the result (A.8) above in the context of our partition algorithm.

Assume that we have $\tilde{K}$ subdomains, and let $\boldsymbol{\mu}^*$ correspond to the anchor point associated with the the particular subdomain that also contains $\boldsymbol{\mu}$. For the RB error bound we then get

$$
\Delta_{\tilde{K}}(\boldsymbol{\mu}) \equiv \frac{\|\mathcal{R}_{\tilde{K}}(\boldsymbol{\mu})\|_X}{\alpha_{\mathrm{LB}}(\boldsymbol{\mu})} \le |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \, \tilde{C} \left(1 + \frac{\overline{\gamma}}{\underline{\alpha}} \right).
\tag{A.9}
$$

Finally, we can now replace the arguments (4.31)–(4.33) by (A.9) in order to replace the constant in Proposition 4.1

$$
C = \frac{2 \overline{\gamma}^2 \tilde{C} |\mathcal{D}|}{\underline{\alpha}^2}
\tag{A.10}
$$

by the constant

$$
C_{\mathrm{im}} = 2 \tilde{C} |\mathcal{D}| \left(1 + \frac{\overline{\gamma}}{\underline{\alpha}} \right).
\tag{A.11}
$$

We note that C_{im} is an improvement over C for $\overline{\gamma}/\underline{\alpha}$ sufficiently large.

REFERENCES

[1] B. O. Almroth, P. Stern, and F. A. Brogan, *Automatic choice of global shape functions in structural analysis*, AIAA Journal, 16 (1978), pp. 525–528.

[2] D. Amsallem, J. Cortial, and C. Farhat, *On-Demand CFD-Based Aeroelastic Predictions Using a Database of Reduced-Order Bases and Models*, in 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, January 2009.

[3] D. Amsallem and C. Farhat, *Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity*, AIAA Journal, 46 (2008), pp. 1803–1813.

[4] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672.

[5] P. Binev, A. Cohen, W. Dahmen, R. DeVore, P. Guergana, and P. Wojtaszczyk, *Convergence Rates for Greedy Algorithms in Reduced Basis Methods*. In preparation, 2010.

[6] S. Boyaval, *Reduced-basis approach for homogenization beyond the periodic setting*, Multiscale Model. Simul., 7 (2008), pp. 466–494.

[7] S. Boyaval, C. Le Bris, Y. Maday, N. C. Nguyen, and A. T Patera, *A reduced basis approach for variational problems with stochastic parameters: Application to heat conduction with variable robin coefficient*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 3187 – 3206.

[8] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici, *A Priori Convergence of the Greedy Algorithm for the Parametrized Reduced Basis*. M2AN, submitted 2009.

[9] J.L. Eftang, B. Stamm, and A.T. Patera, *Multi Parameter Element "hp" Empirical Interpolation*. In preparation, 2010.

[10] J. L. Eftang, D. J. Knezevic, and A. T. Patera, *An hp Certified Reduced Basis Method for Parametrized Parabolic Partial Differential Equations*. Submitted to MCMDS, 2010.

[11] J. L. Eftang, A. T. Patera, and E. M. Rønquist, *An "hp" Certified Reduced Basis Method for Parametrized Parabolic Partial Differential Equations*, in ICOSAHOM Proceedings, 2009.

[12] W. J. Gordon and C. A. Hall, *Construction of curvilinear co-ordinate systems and applications to mesh generation*, Internat. J. Numer. Methods Engrg., 7 (1973), pp. 461–477.

[13] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, M2AN Math. Model. Numer. Anal., 41 (2007), pp. 575–605.

[14] B. Haasdonk, M. Dihlmann, and M. Ohlberger, *A Training Set and Multiple Bases Generation Approach for Parametrized Model Reduction Based on Adaptive Grids in Parameter Space*, Tech. Report 28, SRC SimTech, 2010.

[15] B. Haasdonk and M. Ohlberger, *Adaptive Basis Enrichment for the Reduced Basis Method Applied to Finite Volume Schemes*, in Proceedings of the 5th International Symposium on Finite Volumes for Complex Applications, 2008, pp. 471–478.

[16] ———, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, M2AN Math. Model. Numer. Anal., 42 (2008), pp. 277–302.

[17] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera, *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*, C. R. Math. Acad. Sci. Paris, 345 (2007), pp. 473–478.

[18] David J. Knezevic and Anthony T. Patera, *A certified reduced basis method for the fokker–planck equation of dilute polymeric fluids: Fene dumbbells in extensional flow*, SIAM Journal on Scientific Computing, 32 (2010), pp. 793–817.

[19] N. C. Nguyen, G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced Basis Approximation and* a posteriori *Error Estimation for Parametrized Parabolic PDEs; Application to Real-Time Bayesian Parameter Estimation*, in Computational Methods for Large Scale Inverse Problems and Uncertainty Quantifications, Biegler, Biro, Ghattas, Heinkenschloss, Keyes, Mallick, Tenorio, van Bloemen Waanders, and Willcox, eds., John Wiley & and Sons, UK, Submitted 2009.

[20] N. C. Nguyen, G. Rozza, and A. T. Patera, *Reduced Basis Approximation and* a posteriori *Error Estimation for the Time-Dependent Viscous Burgers' Equation*, Calcolo, 46 (2009), pp. 157–185. Calcolo, 2008 (submitted).

[21] A. K. Noor and J. M. Peters, *Reduced basis technique for nonlinear analysis of structures*, AIAA Journal, 18 (1980), pp. 455–462.

[22] T. A. Porsching, *Estimation of the error in the reduced basis method solution of nonlinear equations*, Math. Comp., 45 (1985), pp. 487–496.

[23] W. C. Rheinboldt, *On the theory and error estimation of the reduced basis method for multi-parameter problems*, Nonlinear Anal. Theory Methods Appl., 21 (1993), pp. 849–858.

[24] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced Basis Approximation and* a posteriori *Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations*, Archives of Computational Methods in Engineering, 15 (2008), pp. 229–275.

[25] S. Sen, *Reduced-Basis Approximation and* a posteriori *Error Estimation for Many-Parameter Heat Conduction Problems*, Numerical Heat Transfer, Part B: Fundamentals, 54 (2008), pp. 369–389.

[26] K. Veroy, C. Prud'homme, D. V. Rovas, and A. T. Patera, *A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations*, in Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, 2003.