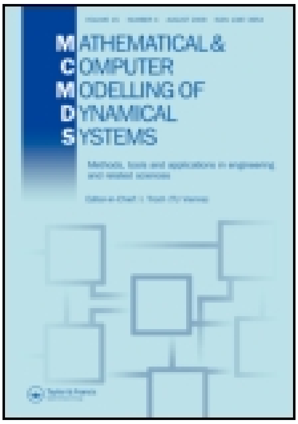


Mathematical and Computer Modelling of Dynamical Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/nmcm20

An *hp* certified reduced basis method for parametrized parabolic partial differential equations

Jens L. Eftang [a], David J. Knezevic [b] & Anthony T. Patera [b]

[a] Department of Mathematical Sciences, Norwegian University of Science and Technology, NO-7491, Trondheim, Norway

[b] Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139, USA

Available online: 28 Jul 2011

To cite this article: Jens L. Eftang, David J. Knezevic & Anthony T. Patera (2011): An *hp* certified reduced basis method for parametrized parabolic partial differential equations, Mathematical and Computer Modelling of Dynamical Systems, 17:4, 395-422

To link to this article: http://dx.doi.org/10.1080/13873954.2011.547670

Mathematical and Computer Modelling of Dynamical Systems
Vol. 17, No. 4, August 2011, 395–422

An *hp* certified reduced basis method for parametrized parabolic partial differential equations

Jens L. Eftang[a]*, David J. Knezevic[b] and Anthony T. Patera[b]

[a]*Department of Mathematical Sciences, Norwegian University of Science and Technology, NO-7491, Trondheim, Norway;* [b]*Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA*

(*Received 5 April 2010; final version received 7 September 2010*)

In this article, we introduce an *hp* certified reduced basis (RB) method for parabolic partial differential equations. We invoke a Proper Orthogonal Decomposition (POD) (in time)/Greedy (in parameter) sampling procedure first in the initial partition of the parameter domain (*h*-refinement) and subsequently in the construction of RB approximation spaces restricted to each parameter subdomain (*p*-refinement). We show that proper balance between additional POD modes and additional parameter values in the initial subdivision process guarantees convergence of the approach. We present numerical results for two model problems: linear convection–diffusion and quadratically non-linear Boussinesq natural convection. The new procedure is significantly faster (more costly) in the RB Online (Offline) stage.

Keywords: parabolic partial differential equations; certified reduced basis; *a posteriori* error estimation; POD/Greedy; *hp* reduced basis; convection–diffusion; Boussinesq natural convection

1. Introduction

The certified reduced basis (RB) method is a model-order reduction framework for rapid evaluation of functional outputs, such as surface temperatures or fluxes, for partial differential equations (PDEs) that depend on an input parameter vector, for example, related to geometric factors or material properties. There are four key ingredients to the certified RB framework:

- Galerkin projection: optimal linear combination of N pre-computed $\mathcal{N}$-degree-of-freedom 'truth' finite element (FE) field snapshots [1,2];
- POD/Greedy sampling: POD (in time)/Greedy (in parameter) [3] optimal selection and combination of FE field snapshots;
- *a posteriori* error estimation: rigorous upper bounds for the error in the RB (output) approximation with respect to the 'truth' FE discretization [4,5]; and
- Offline–Online computational decomposition: $\mathcal{O}(\mathcal{N}^\bullet)$-complexity *preprocessing* followed by $\mathcal{O}(N^\bullet)$-complexity certified *input–output prediction* [5,6].

We shall describe each ingredient further in subsequent sections.

*Corresponding author. Email: eftang@math.ntnu.no

ISSN 1387-3954 print/ISSN 1744-5051 online

DOI: 10.1080/13873954.2011.547670
http://www.informaworld.com

We shall assume that the field variable depends smoothly on the parameters. In that case we can expect, and we can rigorously confirm *a posteriori*, that $N \ll \mathcal{N}$; we can then furthermore anticipate rapid Online evaluation of the RB output approximation and associated RB output error bound. The certified RB method is thus computationally attractive in two important engineering contexts: 'real time', such as parameter estimation and optimal control; 'many query', such as multiscale or stochastic simulation. In both instances, the Offline effort either is *unimportant* or can be *amortized* over many input–output evaluations. In both instances, rigorous error control without direct appeal to the 'truth' is crucial.

For many problems, the field variable may be quite different in different regions of the parameter domain, and hence a snapshot from one region may be of little value to the RB approximation in another region. To exploit this opportunity, we introduce in [7] an *hp*-RB method for linear elliptic equations. In the Offline stage, we first adaptively subdivide the original parameter domain into smaller regions (*h*-refinement); we then construct individual RB approximation spaces spanned by snapshots restricted to parameter values within each of these parameter subdomains (*p*-refinement). In the Online stage, the RB approximation associated with any new parameter value is then constructed as a (Galerkin) linear combination of snapshots from the parameter subdomain that contains the new parameter value. The dimension of the *local* approximation space, and thus the Online cost, shall be very low: every basis function contributes significantly to the RB approximation. We note that an alternative 'multiple bases generation' procedure is introduced in [8]; a different 'interpolation' approach to parametric reduced order modelling with parameter subdomains is described in [9].

In this article, we extend the work in [7] to linear and non-linear parabolic equations through a POD (in time)/Greedy (in parameter) procedure. The POD/Greedy sampling approach [3] is invoked both in the initial partition of the parameter domain (*h*-refinement) and subsequently in the construction of RB approximation spaces restricted to each parameter subdomain (*p*-refinement). Much of the elliptic machinery from [7] extends to the parabolic case because we only subdivide the parameter (and not the temporal) domain. The critical *new* issue for the *hp*-POD/Greedy algorithm for parabolic problems is proper balance between additional POD modes and additional parameter values in the initial subdivision process.

The *hp*-POD/Greedy procedure was first introduced in the conference proceedings paper [10]. We extend [10] here in several important ways. First, we introduce an improvement to the algorithm: an additional Offline splitting step that permits direct control of the Online computational cost. Second, we introduce (for a simple but illustrative case) a new *a priori* convergence theory for the initial subdivision process; we show in particular that the procedure is convergent provided sufficiently many POD modes are included in the RB spaces. Good convergence of the subdivision process is critical to both Offline and Online performances. Third, and finally, we extend our considerations to quadratically non-linear parabolic problems. This class of problems is particularly 'ripe' for the *hp* approach due to the $\mathcal{O}(N^4)$ computational cost associated with RB error bound evaluation [11,12]: even a small reduction in N – the number of RB basis functions – will result in significant Online computational savings.

We begin in Section 2 with the problem statement(s). In Section 3, we introduce the *hp*-RB approximation, the associated RB error bounds and the necessary computational procedures. In Section 4, we present the *hp*-POD/Greedy algorithm and the new *a priori* convergence theory. Finally, in Section 5, we present numerical results for two model problems: a linear time-invariant (LTI) convection–diffusion problem and a quadratically non-linear Boussinesq natural convection problem; we focus our discussion on computational cost and Online economization compared with the standard (p-type) RB method.

2. Problem statement

We directly consider a discrete-time parametrized parabolic PDE defined over a spatial domain $\Omega \subset \mathbb{R}^2$ for discrete time levels $t^k = k\Delta t, 0 \le k \le K$; here $\Delta t = t_f/K$, and t_f is the final time. We further introduce a P-dimensional parameter domain $D \subset \mathbb{R}^P$ and denote by $\mu \in \mathcal{D}$ a particular parameter value. For a given $\mu \in \mathcal{D}$, we shall denote the exact solution to our discrete-time parabolic PDE as $u^k(\mu) \equiv u(t^k, \mu), 0 \le k \le K$.

We consider Backward Euler ($\theta = 1$) and Crank–Nicolson ($\theta = 0.5$) temporal discretization schemes (more generally, we may consider $0.5 \le \theta \le 1$); we define $u^{k+\theta}(\mu) \equiv \theta u^{k+1}(\mu) + (1-\theta)u^k(\mu)$. The exact formulation reads as follows: for any $\mu \in \mathcal{D}$, find $u^k(\mu) \in X$, $1 \le k \le K$, such that

$$\frac{1}{\Delta t} m(u^{k+1}(\mu) - u^k(\mu), v; \mu) + a(u^{k+\theta}(\mu), v; \mu) + b(u^{k+\theta}(\mu), u^{k+\theta}(\mu), v; \mu) = f(v; \mu), \quad \forall v \in X, \tag{1}$$

subject to initial condition $u^0(\mu)$. In the sequel, we shall always assume zero initial conditions. We then evaluate our output of interest as $s^k(\mu) = \ell(u^k(\mu); \mu)$ for $0 \le k \le K$. Here, X denotes a Sobolev space over $\Omega \subset \mathbb{R}^2$; typically $(H_0^1(\Omega))^d \subseteq X \subseteq (H^1(\Omega))^d$, where $H^1(\Omega) = \{v : |\nabla v| \in L^2(\Omega)\}$, $H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$, where $\partial\Omega$ is the boundary of Ω, $L^2(\Omega)$ is the space of square integrable functions over Ω and d is the dimension of the field. (In our exposition $d = 1$; later, for the Boussinesq problem, $d = 3$.)

We suppose that X is equipped with an inner product $(\cdot,\cdot)_X$ and induced norm $\|\cdot\|_X = (\cdot,\cdot)_X^{1/2}$; we further denote by $(\cdot,\cdot)$ the standard $L^2(\Omega)$ inner product and by $\|\cdot\|_{L^2} = (\cdot,\cdot)^{1/2}$ the standard $L^2(\Omega)$ norm. For any $\mu \in \mathcal{D}$, $\underline{m}(\cdot,\cdot;\mu)$ is a coercive and continuous bilinear form over $L^2(\Omega)$, $a(\cdot,\cdot;\mu)$ is a coercive and continuous bilinear form over X, $b(\cdot,\cdot,\cdot;\mu)$ is a continuous trilinear form over X, $f(\cdot;\mu)$ is an X-bounded linear functional and $\ell(\cdot;\mu)$ is an $L^2(\Omega)$-bounded linear 'output' functional. We introduce coercivity constants

$$\alpha(\mu) \equiv \inf_{v\in X} \frac{a(v,v;\mu)}{\| v \|_X^2}, \qquad \sigma(\mu) \equiv \inf_{v\in X} \frac{m(v,v;\mu)}{\| v \|_{L^2}^2}; \tag{2}$$

under our assumptions $\alpha(\mu) > 0$ and $\sigma(\mu) > 0$, respectively, for any $\mu \in \mathcal{D}$. Note for $b = 0$, our problem is linear and coercive.

To develop efficient Offline–Online computational procedures for the RB field approximation, RB output approximation and RB error bound, we shall suppose that all our forms admit 'affine' expansions in functions of μ. Specifically, for any $\mu \in \mathcal{D}$

$$a(\cdot,\cdot;\mu) = \sum_{q=1}^{Q_a} a^q(\cdot,\cdot)\Theta_a^q(\mu), \tag{3}$$

where $Q_a < Q$ and Q is finite and preferably modest. We suppose that m, b and f admit similar expansions in at most Q terms. Many problems (including the examples of this article) admit an affine expansion; for other problems, approximate affine representations can be developed [13,14].

We now introduce the 'truth' spatial discretization of the PDE. We suppose a regular triangulation $\mathcal{T}^{\mathcal{N}}(\Omega)$ of Ω and introduce a corresponding high-resolution FE space $X^{\mathcal{N}} \subset X$ of dimension $\mathcal{N}$. The truth discretization of Equation (1) reads as follows: for any $\mu \in \mathcal{D}$, find $u^{\mathcal{N}\,k}(\mu) \in X^{\mathcal{N}}$, $1 \leq k \leq K$, such that

$$\begin{aligned}\frac{1}{\Delta t} m(u^{\mathcal{N}\,k+1}(\mu) - u^{\mathcal{N}\,k}(\mu), v; \mu) + a(u^{\mathcal{N}\,k+\theta}(\mu), v; \mu) \\ + b(u^{\mathcal{N}\,k+\theta}(\mu), u^{\mathcal{N}\,k+\theta}(\mu), v; \mu) = f(v; \mu), \quad \forall v \in X^{\mathcal{N}},\end{aligned} \tag{4}$$

subject to initial condition $u^{\mathcal{N}\,0} = 0$; then evaluate the truth output approximation as $s^{\mathcal{N}\,k}(\mu) = \ell(u^{\mathcal{N}\,k}(\mu); \mu)$ for $0 \leq k \leq K$. It is this truth FE approximation that we wish to accelerate by RB treatment. We shall assume that $X^{\mathcal{N}}$ is rich enough that the exact and truth solutions are indistinguishable at the desired level of numerical accuracy. As we shall observe below, the RB Online computational cost is independent of $\mathcal{N}$, and the RB approximation is stable as $\mathcal{N} \to \infty$. We can thus choose $\mathcal{N}$ conservatively.

3. *hp*-RB approximation

For a parameter domain $\mathcal{D} \subset \mathbb{R}^P$, the *hp*-RB method serves to construct a hierarchical partition of $\mathcal{D}$ into M distinct parameter subdomains $\mathcal{V}_{B^m} \subset \mathcal{D}$, $1 \leq m \leq M$. Each of these subdomains $\mathcal{V}_{B^m}$ has associated nested RB approximation spaces $X_{1,B^m} \subset, \ldots, \subset X_{N_{\max,B^m}, B^m}$, where $\dim(X_{N,B^m}) = N$, $1 \leq N \leq N_{\max,B^m}$. We define $N_{\max} \equiv \max_{1 \leq m \leq M} N_{\max,B^m}$. The procedure for the construction of the parameter domain partition and associated RB spaces, as well as the form of the 'identifiers' B^m, shall be made explicit in Section 4. In this section, we discuss the RB approximation, the RB *a posteriori* error estimators and the associated computational procedures *given* the parameter domain partition and associated RB spaces.

3.1. RB approximation

For any new $\mu \in \mathcal{D}$, we first determine $m^* \in [1, M]$ such that $\mu \in \mathcal{V}_{B^{m^*}}$ $(\subset \mathcal{D})$. Given any N, we define $\hat{N} \equiv \min\{N, N_{\max,B^{m^*}}\}$. The RB approximation of Equation (4) reads as follows: for any $\mu \in \mathcal{D}$, find $u_N^k(\mu) \in X_N \equiv X_{\hat{N},B^{m^*}}$, $1 \leq k \leq K$, such that

$$\begin{aligned}\frac{1}{\Delta t} m(u_N^{k+1}(\mu) - u_N^k(\mu), v; \mu) + a(u_N^{k+\theta}(\mu), v; \mu) \\ + b(u_N^{k+\theta}(\mu), u_N^{k+\theta}(\mu), v; \mu) = f(v; \mu), \quad \forall v \in X_N,\end{aligned} \tag{5}$$

subject to initial condition $u_N^0 = 0$; then evaluate the RB output approximation as $s_N^k(\mu) = \ell(u_N^k(\mu); \mu)$ for $0 \leq k \leq K$.

3.2. A posteriori error estimation

A rigorous *a posteriori* upper bound for the RB error is crucial for the Offline *hp*-POD/Greedy sampling procedure as well as for the Online certification of the RB approximation and the RB output. The key computational ingredients of the RB error bound are the RB residual dual norm and lower bounds for the stability constants.

Given an RB approximation, $u_N^k(\mu)$, $0 \le k \le K$, for $\mu \in \mathcal{D}$, we write the RB residual $r_N^k(\upsilon;\mu)$, $1 \le k \le K$, as

$$r_N^{k+1}(v;\mu) = f(v;\mu) - \frac{1}{\Delta t} m(u_N^{k+1}(\mu) - u_N^k(\mu), v;\mu) - a(u_N^{k+\theta}(\mu), v;\mu) \\ - b(u_N^{k+\theta}(\mu), u_N^{k+\theta}(\mu), v;\mu), \quad \forall v \in X^{\mathcal{N}}. \tag{6}$$

The Riesz representation of the residual $\hat{e}_N^k(\mu) \in X^{\mathcal{N}}$, $1 \le k \le K$, satisfies

$$(\hat{e}_N^k(\mu), v)_X = r_N^k(v;\mu), \quad \forall v \in X^{\mathcal{N}}. \tag{7}$$

We denote by $\epsilon_N^k(\mu) = \| \hat{e}_N^k(\mu) \|_X = \sup_{v \in X^{\mathcal{N}}} \frac{r_N^k(v;\mu)}{\|v\|_X}$ the residual dual norm.

We next introduce positive lower bounds for the coercivity constants of m and a, σ_{LB} and α_{LB}, respectively, such that for all $\mu \in \mathcal{D}$

$$0 < \sigma_{\mathrm{LB}}(\mu) \le \sigma(\mu), \qquad 0 < \alpha_{\mathrm{LB}}(\mu) \le \alpha(\mu). \tag{8}$$

We also introduce a lower bound for the (possibly negative) stability constant

$$\rho_N(t^{k+1};\mu) \equiv \inf_{v \in X^{\mathcal{N}}} \frac{2b(u_N^{k+\theta}(\mu), v, v;\mu) + a(v,v;\mu)}{\| v \|_{L_2}^2}, \quad 0 \le k \le K-1, \tag{9}$$

which we shall denote $\rho_N^{\mathrm{LB}}(t^k;\mu)$: $\rho_N^{\mathrm{LB}}(t^k;\mu) \le \rho_N(t^k;\mu)$ for $1 \le k \le K$ and all $\mu \in \mathcal{D}$. We further define $\tau_N^{\mathrm{LB}}(t^k;\mu) = \min(\rho_N^{\mathrm{LB}}(t^k;\mu), 0)$.

We can then develop the $L^2(\Omega)$ error bound

$$\Delta_N^k(\mu) = \sqrt{\frac{\Delta t \sum_{k'=1}^{k} \left(\frac{\epsilon_N(t^{k'};\mu)^2}{1-(1-\theta)\Delta t\, \tau_N^{\mathrm{LB}}(t^{k'};\mu)} \prod_{j=1}^{k'-1} \frac{1+\theta \Delta t\, \tau_N^{\mathrm{LB}}(t^j;\mu)}{1-(1-\theta)\Delta t\, \tau_N^{\mathrm{LB}}(t^j;\mu)} \right)}{\alpha_{\mathrm{LB}}(\mu)\sigma_{\mathrm{LB}}(\mu) \prod_{k'=1}^{k} \frac{1+\theta \Delta t\, \tau_N^{\mathrm{LB}}(t^{k'};\mu)}{1-(1-\theta)\Delta t\, \tau_N^{\mathrm{LB}}(t^{k'};\mu)}}}, \tag{10}$$

for which it can be demonstrated [4,12,11] that $\| u^{\mathcal{N}\,k}(\mu) - u_N^k(\mu) \|_{L^2} \le \Delta_N^k(\mu)$, $1 \le k \le K$, $\forall \mu \in \mathcal{D}$.[1] We can furthermore develop an RB output error bound

$$\Delta_{N,s}^k(\mu) \equiv \left(\sup_{v \in X^{\mathcal{N}}} \frac{\ell(v;\mu)}{\| v \|_{L_2}} \right) \Delta_N^k(\mu), \tag{11}$$

for which it can be demonstrated that $|s^{\mathcal{N}\,k}(\mu) - s_N^k(\mu)| \le \Delta_{N,s}^k(\mu)$, $1 \le k \le K$, $\forall \mu \in \mathcal{D}$.

3.3. Computational procedures

3.3.1. Construction–evaluation

Thanks to the 'affine' assumption (3), we can develop Construction–Evaluation procedures for the RB field, RB output and RB error bound. We first consider the RB field and the RB output. In the *Construction* stage, given the RB basis functions, we form and store all the necessary parameter-independent entities at cost $\mathcal{O}(\mathcal{N}^\bullet)$. In the *Evaluation* stage, we first determine the subdomain to which the given new parameter μ belongs: an $\mathcal{O}(\log_2 M)$

binary search suffices thanks to the hierarchical subdomain construction, which we will make explicit in the next section [7]. We next assemble the RB system (5) at cost $\mathcal{O}(QN^2)$ ($N \leq N_{\max}$) in the LTI case [6] and at cost $\mathcal{O}(n_{\text{Newton}}QN^3K)$ in the quadratically non-linear case [11,12]; we then solve this system at cost $\mathcal{O}(N^3 + KN^2)$ in the LTI case and at cost $\mathcal{O}(n_{\text{Newton}}KN^3)$ in the quadratically non-linear case. (Here n_{Newton} is the number of Newton iterations required to solve the non-linear equations at each timestep.) Given the RB field, the RB output can be evaluated at cost $\mathcal{O}(KN)$.

We next consider the RB error bound (10). We invoke the Riesz representation of the residual and linear superposition to develop Construction–Evaluation procedures for the residual dual norm.[2] In the *Construction* stage, we again compute and store all the necessary parameter-independent entities at cost $\mathcal{O}(\mathcal{N}^{\bullet})$. In the *Evaluation* stage, we can evaluate the residual dual norm at cost $\mathcal{O}(KN^2 + Q^2N^2)$ for LTI problems [6] and at cost $\mathcal{O}(KQ^2N^4)$ for quadratically non-linear problems [11,12]. (In the sequel, we shall assume $Q = \mathcal{O}(1)$, as is the case in our numerical examples.) We note that the $\mathcal{O}(N^4)$ cost for quadratically non-linear problems compromises rapid evaluation for larger N and in practice limits $N_{\max}$ – motivation for an *hp* approach.

3.3.2. *Offline–Online decomposition*

The Construction–Evaluation procedures enable efficient Offline–Online decomposition for the computation of the RB field approximation, RB output approximation and RB output error bound. The Offline stage, which is performed only once as preprocessing, can be very expensive – $\mathcal{N}$-dependent complexity; the Online stage, which is typically performed many times, is comparably inexpensive – $\mathcal{N}$-independent complexity. We note that our RB formulation (5) inherits the temporal discretization of the truth (4); we may thus *not* choose Δt arbitrarily small without compromise to RB Online cost.

In the *hp*-RB *Offline* stage, we perform the *hp*-POD/Greedy sampling procedure, which we discuss in the next section and which is the focus of this article: we invoke Construction–Evaluation procedures to identify good RB spaces and to compute and store the Construction quantities required in the Online stage. The link between the Offline and Online stages is the permanent storage of the *Online Dataset*; the storage requirement for the *hp*-RB method is $\mathcal{O}(MN_{\max}^2)$ in the linear case and $\mathcal{O}(MN_{\max}^4)$ in the quadratically non-linear case. We recall that M is the number of subdomains identified by the *hp*-POD/Greedy. In the *hp*-RB *Online* stage, we perform Evaluation based on the Online Dataset: we calculate the RB field approximation, the RB output approximation and the RB error bound at the given new parameter in $\mathcal{O}(N^{\bullet})$ complexity.

4. *hp*-POD/Greedy sampling

In this section, we discuss the *hp*-POD/Greedy procedure for the construction of the parameter subdomain partition and the associated RB approximation spaces. We employ a hierarchical parameter domain splitting procedure and hence we may organize the subdomains in a binary tree. Let L denote the number of levels in the tree. For $1 \leq l \leq L$, we introduce Boolean vectors

$$B_l = (1, i_1, i_2, \ldots, i_l) \in \{1\} \times \{0, 1\}^l. \tag{12}$$

For any B_l, $1 \leq l \leq L - 1$ we define the concatenation $(B_l, i) \equiv (1, i_1, \ldots, i_l, i)$, $i \in \{0, 1\}$. The M subdomains of $\mathcal{D}$ are associated to the M leaf nodes of the binary tree; we denote by B^m, $1 \le m \le M$, the Boolean vectors that correspond to the leaf nodes; we can thus label the parameter subdomains as $\mathcal{V}_{B^m} \subset \mathcal{D}$, $1 \le m \le M$. Similarly, we denote by $X_{1,B^m} \subset \cdots \subset X_{N_{\max,B^m},B^m}$ $(\subset X^{\mathcal{N}})$ the set of nested RB approximation spaces associated to $\mathcal{V}_{B^m}$, $1 \le m \le M$.

4.1. Procedure

The *hp*-POD/Greedy algorithm introduced here applies to both the linear and non-linear cases. However, we adopt the notation of the linear ($b = 0$) and scalar ($d = 1$) problems for simplicity.

Algorithm 4.1: $[\{\chi^i \in X, 1 \le i \le \Delta N\}] = \mathrm{POD}(\{w^k \in X^{\mathcal{N}}, 1 \le k \le K\}, \Delta N)$

1: $C_{ij} \leftarrow (w^i, w^j)_X/K$, $1 \le i,j \le K$;
2: Solve $C\psi^i = \lambda^i \psi^i$, $(\psi^i)^T C \psi^i = \frac{1}{K}$, for $(\psi^i \in \mathbf{R}^K, \lambda^i \in \mathbf{R})$ associated with the ΔN largest eigenvalues of C;
3: Compute $\chi^i = \sum_{k=1}^{K} \psi_k^i w^k$ for $1 \le i \le \Delta N$.

We introduce as Algorithm 4.1 the POD algorithm (the Method of Snapshots [16]). For specified ΔN and $\{w^k \in X^{\mathcal{N}}, 1 \le k \le K\}$, Algorithm 4.1 returns $\Delta N \le K$ X-orthonormal functions[3] $\{\chi^i \in X, 1 \le i \le \Delta N\}$ such that $\mathcal{P}_{\Delta N} = \mathrm{span}\{\chi^i, 1 \le i \le \Delta N\}$ satisfies the optimality property

$$\mathcal{P}_{\Delta N} = \arg \inf_{\substack{Y \subset \mathrm{span}\{w^k, 1 \le k \le K\} \\ \dim Y \le \Delta N}} \left(\frac{1}{K} \sum_{k=1}^{K} \inf_{w \in Y} \| w^k - w \|_X^2 \right)^{1/2}. \tag{13}$$

The set $\{\chi^i, 1 \le i \le \Delta N\}$ contains the ΔN first *POD modes* of $\mathrm{span}\{w^1, \ldots, w^K\}$.

We next introduce as Algorithm 4.2 the POD/Greedy sampling procedure of [3] (see also [17]). Let $\mathcal{V} \subseteq \mathcal{D}$. For specified ΔN, an RB space dimension upper bound $\overline{N}$, an initial parameter value $\mu^* \in \mathcal{V}$, a finite train sample $\Xi_{\text{train}} \subset \mathcal{V}$ and an error bound tolerance ϵ, Algorithm 4.2 returns $\tilde{N}_{\max} \le \overline{N}$ nested RB spaces $X_1 \subset \ldots \subset X_{\tilde{N}_{\max}}$ (note that as the spaces are nested by construction, we only specify $X_{\tilde{N}_{\max}}$ as the return argument) and $\epsilon_{\max} = \max_{\mu \in \Xi_{\text{train}}} \Delta^K_{\tilde{N}_{\max}}(\mu)$ such that either $\epsilon_{\max} \le \epsilon$ or $\tilde{N}_{\max} = \overline{N}$. (Note that in the POD/Greedy we may take the $L^2([0, t_f]; X)$ RB error bound $\Delta^K_{N,X}$ rather than the $L^2(\Omega)$ RB error bound Δ^K_N [17]; for the linear coercive case, $\Delta^K_{N,X}(\mu) = \sigma_{\text{LB}}^{1/2}(\mu)\Delta^K_N(\mu)$.)

We initialize the POD/Greedy by setting $N = 0$, $X_N = \{0\}$ and $\epsilon_{\max} = \infty$. Then, while the dimension of the RB space is less than $\overline{N}$ *and* the tolerance ϵ is not satisfied over Ξ_{train}, we enrich the RB space: we first compute the projection error $e^k_{N,\text{proj}}(\mu^*) = u^{\mathcal{N}\,k}(\mu^*) - \mathrm{proj}_{X_N}(u^{\mathcal{N}\,k}(\mu^*))$, $1 \le k \le K$, where $\mathrm{proj}_{X_N}(w)$ denotes the X-orthogonal projection of $w \in X^{\mathcal{N}}$ onto X_N; we next increase the dimension of the RB space by adding the ΔN first POD modes of the projection error to the current RB space; we then greedily determine the next parameter value over Ξ_{train} based on the *a posteriori* error estimator at the final time. We invoke Construction–Evaluation procedures for the computation of the maximum RB error bound over Ξ_{train} (line 7 of Algorithm 4.2); as the RB error bound calculation is very fast ($\mathcal{N}$-independent in the limit of many evaluations), we may choose Ξ_{train} very dense.

Algorithm 4.2: $[X_{\tilde{N}_{\max}}, \epsilon_{\max}] = \mathrm{POD/Greedy}(\Delta N, \overline{N}, \epsilon, \mu^*, \Xi_{\mathrm{train}})$

1: Set $X_N = \{0\}$, $N = 0$, $\epsilon_{\max} = \infty$;
2: **while** $\epsilon_{\max} > \epsilon$ and $N < \overline{N}$ **do**
3: $\quad e^k_{N,\mathrm{proj}}(\mu^*) \leftarrow u^{\mathcal{N}\,k}(\mu^*) - \mathrm{proj}_{X_N}(u^{\mathcal{N}\,k}(\mu^*))$, $1 \le k \le K$;
4: $\quad$ **for** $i = 1, \ldots, \min\{\Delta N, \overline{N} - N\}$ **do**
5: $\quad\quad X_{N+i} \leftarrow X_N \oplus \mathrm{span}\{POD(\{e^k_{N,\mathrm{proj}}(\mu^*), 1 \le k \le K\}, i)\}$;
6: $\quad$ **end for**
7: $\quad \mu^* \leftarrow \arg\max_{\mu \in \Xi} \Delta^K_N(\mu)$;
8: $\quad \epsilon_{\max} \leftarrow \Delta^K_N(\mu^*)$;
9: $\quad N \leftarrow N + \Delta N$;
10: **end while**
11: $\tilde{N}_{\max} \leftarrow N$;

We finally introduce as Algorithm 4.3 the *hp*-POD/Greedy algorithm. For specified ΔN, an RB space dimension upper bound $\overline{N}$, error bound tolerances $\epsilon^1_{\mathrm{tol}}$ and $\epsilon^2_{\mathrm{tol}}$, an initial parameter *anchor point* $\hat{\mu}^0_{(1)}$ and an initial train sample $\Xi_{\mathrm{train},(1)} \subset \mathcal{D}$ of cardinality n_{train}, Algorithm 4.3 constructs a hierarchical splitting of $\mathcal{D}$ into $M = M(\epsilon^1_{\mathrm{tol}}, \overline{N})$ subdomains $\mathcal{V}_{B^m}$, $1 \le m \le M$, and associates to each parameter subdomain an RB space $X_{N_{\max,B^m},B^m}$ of dimension $N_{\max,B^m} \le N_{\max} \le \overline{N}$ such that for each subdomain $\mathcal{V}_{B^m}$, the tolerance $\epsilon^1_{\mathrm{tol}} > 0$ is satisfied over $\Xi_{\mathrm{train},B^m} \subset V_{B^m}$ by $\tilde{\Delta}^K_{R,B^m}$ *and* the tolerance $\epsilon^2_{\mathrm{tol}}$ is satisfied over Ξ_{train,B^m} by $\Delta^K_{N_{\max,B^m}}$. We introduce here $\tilde{\Delta}^K_{R,B_l}$ as the RB error bound associated with the *temporary* space $\tilde{X}_{R,B_l}$, and we recall that $\Delta^K_{N_{\max,B^m}}$ is the RB error bound associated with the *returned* space $X_{N_{\max,B^m},B^m}$. (In the *hp*-RB Online stage, we may readily extract spaces $X_{N,B^m} \subset X_{N_{\max,B^m}}$ of any dimension N, $1 \le N \le N_{\max,B^m}$.)

We now comment on the constant $\eta > 1$, which in turn determines the dimension R of the temporary spaces $\tilde{X}_{R,B_l}$ (lines 3–6): we successively increment R and evaluate $\tilde{\Delta}^K_{R,B_l}(\hat{\mu}^0_{B_l})$ until $\tilde{\Delta}^K_{R,B_l}(\hat{\mu}^0_{B_l}) < \epsilon^1_{\mathrm{tol}}/\eta$. For $\eta > 1$, the tolerance $\epsilon^1_{\mathrm{tol}}$ is then satisfied by $\tilde{\Delta}^K_{R,B_l}$ in a neighbourhood of the anchor point $\hat{\mu}^0_{B_l}$, and we thus avoid arbitrarily small subdomains. We note that $\eta = \infty$ corresponds to $R = K$; however, typically $R \ll K$ is sufficient and we may thus choose η close to (but larger than) unity.

We next consider the splitting of any particular subdomain $\mathcal{V}_{B_l} \subset \mathcal{D}$ into two new subdomains $\mathcal{V}_{(B_l,0)} \subset \mathcal{V}_{B_l}$ and $\mathcal{V}_{(B_l,1)} \subset \mathcal{V}_{B_l}$. We suppose that $\mathcal{V}_{B_l}$ is equipped with a train sample $\Xi_{\mathrm{train},B_l} \subset \mathcal{V}_{B_l}$. Given a parameter anchor point $\hat{\mu}^0_{B_l} \in \mathcal{V}_{B_l}$, we first compute the truth field $u^{\mathcal{N}\,k}(\hat{\mu}^0_{B_l})$, $1 \le k \le K$, and define the temporary RB space $\tilde{X}_{R,B_l}$ associated with the subdomain $\mathcal{V}_{B_l}$ as discussed above. The next step is to evaluate $\tilde{\Delta}^K_{R,B_l}(\mu)$ for all $\mu \in \Xi_{\mathrm{train},B_l}$ in order to identify a second anchor point (line 7) $\hat{\mu}^1_{B_l} = \arg\max_{\mu \in \Xi_{\mathrm{train},B_l}} \tilde{\Delta}^K_{R,B_l}(\mu)$. We note that the two anchor points $\hat{\mu}^0_{B_l}$ and $\hat{\mu}^1_{B_l}$ are maximally different in the sense of the RB error bound, and thus provide good initial parameter values for two new RB spaces.

We now introduce a distance function, $\delta : D \times \mathcal{D} \to \mathbb{R}$; for example, we may choose Euclidean distance. We can then implicitly define two new subdomains $\mathcal{V}_{(B_l,0)} \subset \mathcal{V}_{B_l}$ and $\mathcal{V}_{(B_l,1)} \subset \mathcal{V}_{B_l}$ based on the distance to the two anchor points: $\mathcal{V}_{(B_l,0)} = \{\mu \in \mathcal{V}_{B_l} : \delta(\hat{\mu}^0_{B_l}, \mu) < \delta(\hat{\mu}^1_{B_l}, \mu)\}$ and $\mathcal{V}_{(B_l,1)} = \{\mu \in \mathcal{V}_{B_l} : \delta(\hat{\mu}^0_{B_l}, \mu) \ge \delta(\hat{\mu}^1_{B_l}, \mu)\}$. Note that by this definition, parameter values that are equidistant from the two anchor points $\hat{\mu}^0_{B_l}$ and $\hat{\mu}^1_{B_l}$

Algorithm 4.3: *hp*-POD/Greedy($\Xi_{\text{train},B_l}, \hat{\mu}^0_{B_l}, \epsilon^1_{\text{tol}}, \epsilon^2_{\text{tol}}, \overline{N}, \Delta N$)

1: Set $R \leftarrow 0, \tilde{X}_{R,B_l} \leftarrow \{0\}$;
2: Compute $u^{\mathcal{N}\,k}(\hat{\mu}^0_{B_l}), 1 \le k \le K$;
3: **while** $\tilde{\Delta}^K_{R,B_l}(\hat{\mu}^0_{B_l}) > \epsilon_{\text{tol}}/\eta$ **do**
4: $\quad R \leftarrow R + 1$;
5: $\quad \tilde{X}_{R,B_l} \leftarrow \text{span}\{\text{POD}(\{u^{\mathcal{N}\,k}(\hat{\mu}^0_{B_l}), 1 \le k \le K\}, R)\}$;
6: **end while**
7: $\hat{\mu}^1_{B_l} \leftarrow \arg\max_{\mu \in \Xi_{\text{train},B_l}} \tilde{\Delta}^K_{R,B_l}(\mu)$ and set $\hat{\mu}^0_{(B_l,0)} \leftarrow \hat{\mu}^0_{B_l}, \hat{\mu}^0_{(B_l,1)} \leftarrow \hat{\mu}^1_{B_l}$;
8: **if** $\max_{\mu \in \Xi_{\text{train},B_l}} \tilde{\Delta}^K_{R,B_l}(\mu) > \epsilon^1_{\text{tol}}$ **then**
9: $\quad$ Determine $\Xi_{\text{train},(B_l,0)}, \Xi_{\text{train},(B_l,1)}$;
10: $\quad X_{N_{\max,(B_l,0)},(B_l,0)} \leftarrow$ *hp*-POD/Greedy($\Xi_{\text{train},(B_l,0)}, \hat{\mu}^0_{(B_l,0)}, \epsilon^1_{\text{tol}}, \epsilon^2_{\text{tol}}, \overline{N}, \Delta N$);
11: $\quad X_{N_{\max,(B_l,1)},(B_l,1)} \leftarrow$ *hp*-POD/Greedy($\Xi_{\text{train},(B_l,1)}, \hat{\mu}^0_{(B_l,1)}, \epsilon^1_{\text{tol}}, \epsilon^2_{\text{tol}}, \overline{N}, \Delta N$);
12: **else**
13: $\quad [X_{N_{\max,B_l},B_l}, \epsilon_{\max}] = \text{POD/Greedy}(\Delta N, \overline{N}, \epsilon^2_{\text{tol}}, \hat{\mu}^0_{B_l}, \Xi_{\text{train},B_l})$;
14: $\quad$ **if** $\epsilon_{\max} > \epsilon^2_{\text{tol}}$ **then**
15: $\quad\quad$ Discard $X_{N_{\max,B_l},B_l}$;
16: $\quad\quad$ Determine $\Xi_{\text{train},(B_l,0)}, \Xi_{\text{train},(B_l,1)}$;
17: $\quad\quad X_{N_{\max,(B_l,0)},(B_l,0)} \leftarrow$ *hp*-POD/Greedy($\Xi_{\text{train},(B_l,0)}, \hat{\mu}^0_{(B_l,0)}, \epsilon^1_{\text{tol}}, \epsilon^2_{\text{tol}}, \overline{N}, \Delta N$);
18: $\quad\quad X_{N_{\max,(B_l,1)},(B_l,1)} \leftarrow$ *hp*-POD/Greedy($\Xi_{\text{train},(B_l,1)}, \hat{\mu}^0_{(B_l,1)}, \epsilon^1_{\text{tol}}, \epsilon^2_{\text{tol}}, \overline{N}, \Delta N$);
19: $\quad$ **else**
20: $\quad\quad$ Let $m =$ (number of spaces returned so far $+ 1$) and set $B^m \equiv B_l$;
21: $\quad\quad$ **return** $X_{N_{\max,B^m},B^m} \equiv X_{N_{\max,B_l},B_l}$;
22: $\quad$ **end if**
23: **end if**

---

belong to $\mathcal{V}_{(B_l,1)}$. The final step of splitting is to construct a new train sample associated with each of the two new subdomains (line 9). We first enrich (by adding random points, say) the current train sample $\tilde{\Xi}_{\text{train},B_l} \supset \Xi_{\text{train},B_l}$ such that $\tilde{\Xi}_{\text{train},B_l} \subset \mathcal{V}_{B_l}$ has cardinality $2n_{\text{train}}$; we then define

$$\Xi_{\text{train},(B_l,i)} \equiv \tilde{\Xi}_{\text{train},B_l} \cap \mathcal{V}_{(B_l,i)}, \quad i = 0, 1. \tag{14}$$

We note that we may choose the initial train sample for the *hp*-POD/Greedy to be rather sparse compared with the train sample for the standard POD/Greedy, because we effectively construct an adaptively refined train sample (over $\mathcal{D}$) during the parameter domain partition process. The adaptively generated *hp*-POD/Greedy train sample associated with a given subdomain is typically much smaller than the (global) train sample associated with the standard POD/Greedy.

We apply this splitting scheme recursively to partition $\mathcal{D}$ into the final M subdomains; we can thus organize the subdomains in a binary tree. In Figure 1, we illustrate the procedure, as well as the associated binary tree, for two levels of recursive splitting.

The final step is p-refinement: we identify the nested RB spaces to be associated with the subdomain (line 13). If the POD/Greedy returns with $\epsilon_{\max} > \epsilon^2_{\text{tol}}$, we discard the generated basis and successively perform additional subdomain splitting and POD/Greedy steps until the tolerance is satisfied with at most $\overline{N}$ basis functions (lines 15–18). This additional splitting step permits simultaneous control over ϵ^2_{tol} and $N_{\max}$. We note that

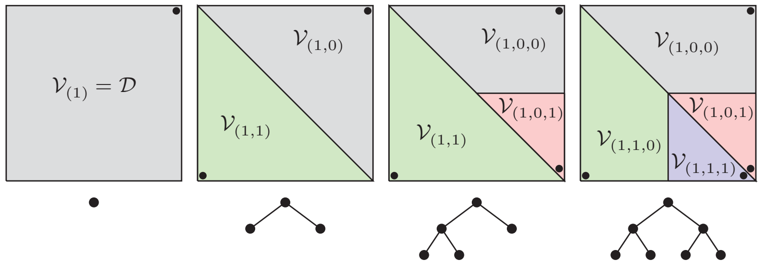


Figure 1. Two levels of h-refinement and associated binary tree; here $L = 3$.

ΔN – the number of POD modes to include at each Greedy iteration during p-refinement – is typically chosen small: small ΔN leads to more optimal spaces albeit at a higher (Offline) computational cost.

Under the assumption that $\overline{N}$ is chosen such that R is always smaller than $\overline{N}$ (note that we can always 're-specify' $\overline{N}$ if at any point $R > \overline{N}$), the hp-POD/Greedy algorithm provides an Online Dataset such that the RB error bound tolerance ϵ^2_{tol} is satisfied (over the train samples) with at most $N_{\max} \leq \overline{N}$ basis functions. We hope to achieve this goal without the expensive execution of lines 15–18: it is our intent that if ϵ^1_{tol} is satisfied with R basis functions, then $\epsilon^2_{\text{tol}} < \epsilon^1_{\text{tol}}$ will be satisfied with at most $\overline{N} > R$ basis functions; whenever this is true, we discard only R basis functions at each level of splitting.

We regard lines 15–18 as insurance: if ϵ^2_{tol} is not satisfied with at most $\overline{N}$ basis functions – even if ϵ^1_{tol} was satisfied with R basis functions – we discard the computed candidate space, split the subdomain and again execute hp-POD/Greedy in a recursive manner. Ideally ϵ^1_{tol} is chosen such that the insurance is rarely invoked *and* $N_{\max,B^m} \leq \overline{N}$ is close to $\overline{N}$ for most $m \in [1, M]$. If the insurance is invoked too often – ϵ^1_{tol} is too large with respect to the target $\overline{N}$ – the Offline computational cost will be large. If the insurance is rarely or never invoked and $N_{\max,B^m} \ll \overline{N}$ for most $m \in [1, M]$, then ϵ^1_{tol} is too small with respect to the target $\overline{N}$.

Remark 4.1: We note that as the number of subdomains M increases, the hp-POD/Greedy algorithm in general requires a larger (Offline) computational cost and generates a larger Online Dataset than the standard (p-type) POD/Greedy method. However, in the non-linear case, the $\mathcal{O}(N^4)$ cost and storage associated with the RB error bound help to moderate this increase: an increase in M provides a decrease in N such that the product MN^4 grows only modestly. We further note that, thanks to the efficient $\log_2(M)$ subdomain search, M can be very large without compromise to the Online computational cost. In practice, we thus seek M to balance Offline cost and Online storage against Online speed.

Remark 4.2: As discussed in [11,12], we must employ a 'nominal' lower bound ρ^* for the stability factor ρ_N for non-linear parabolic problems during the POD/Greedy: the SCM, which allows for construction of the rigorous lower bound ρ_N^{LB}, can only be performed after generation of the RB space. In this context, ρ^* is a conservatively chosen constant or (say) a linear function of μ. Note that the rigour of our error bounds in the Online stage is not compromised: after completion of the POD/Greedy, we perform the SCM,[4] and subsequently the Online RB error bounds are rigorous.

4.2. A priori convergence analysis

We now introduce an *a priori* convergence theory for Algorithm 4.3. Selection of relatively few and optimal subdomains – small M for specified $\epsilon^1_{\mathrm{tol}}$ – is crucial to reduce both Offline cost and Online cost and storage. We consider here the class of linear scalar problems ($b = 0$, $d = 1$). For simplicity, we consider the case of a single parameter ($P = 1$); we assume a Backward Euler temporal discretization ($\theta = 1$); and we consider the case in which $m(\cdot,\cdot;\mu)$ is parameter independent and in particular equal to the $L^2(\Omega)$ inner product: $m(w,v;\mu) \equiv m(w,v) \equiv \int_\Omega wv$.

We recall that the bilinear form a and the linear functional f admit the affine expansions

$$a(\cdot,\cdot;\mu) = \sum_{q=1}^{Q_a} a^q(\cdot,\cdot)\Theta^q_a(\mu), \qquad f(\cdot;\mu) = \sum_{q=1}^{Q_f} f^q(\cdot)\Theta^q_f(\mu), \tag{15}$$

for all $\mu \in \mathcal{D}$. For our purposes in this section, we shall require that

$$a(\cdot,\cdot;\mu) = a^1(\cdot,\cdot) + \sum_{q=2}^{Q_a} a^q(\cdot,\cdot)\Theta^q_a(\mu) \equiv a^1(\cdot,\cdot) + a_{\mathrm{II}}(\cdot,\cdot;\mu), \tag{16}$$

where a^1 is an X-inner product and a_{II} is L^2-continuous in its second argument. Specifically we require, for any $v \in X$, $w \in X$,

$$a^1(v,w) \le \| v \|_X \| w \|_X, \tag{17}$$

$$a^q(v,w) \le \gamma^q \ \| v \|_X \| w \|_{L^2}, \quad 2 \le q \le Q_a. \tag{18}$$

We also require that the $f^q : X \to \mathbb{R}$ are L^2-bounded:

$$f^q(v) \le \| f^q \|_{L^2} \| v \|_{L^2}, \quad 1 \le q \le Q_f. \tag{19}$$

For simplicity, we suppose that $\|\cdot\|_X = \|\cdot\|_{H^1}$; hence $\| v \|_{L^2} \le \| v \|_X$ for all $v \in X$. We further require that the $\Theta^q_a : \mathcal{D} \to \mathbb{R}$ and $\Theta^q_f : \mathcal{D} \to \mathbb{R}$ are Lipschitz continuous: for any $\mu_1 \in \mathcal{D}$, $\mu_2 \in \mathcal{D}$, there exist constants L^q_a and L^q_f, $1 \le q \le Q_a$, such that

$$|\Theta^q_a(\mu_1) - \Theta^q_a(\mu_2)| \le L^q_a |\mu_1 - \mu_2|, \quad 1 \le q \le Q_a, \tag{20}$$

$$|\Theta^q_f(\mu_1) - \Theta^q_f(\mu_2)| \le L^q_f |\mu_1 - \mu_2|, \quad 1 \le q \le Q_f. \tag{21}$$

We introduce lower and upper bounds over $\mathcal{D}$ for the coercivity and continuity constants of $a(\cdot,\cdot;\mu)$:

$$0 < \underline{\alpha} \equiv \min_{\mu\in\mathcal{D}} \alpha(\mu) = \min_{\mu\in\mathcal{D}} \inf_{v\in X} \frac{a(v,v;\mu)}{\| v \|^2_X}, \quad \infty > \overline{\gamma} \ge \max_{\mu\in\mathcal{D}} \sup_{v\in X} \sup_{w\in X} \frac{a(v,w;\mu)}{\| v \|_X \| w \|_X}, \tag{22}$$

respectively. For simplicity of notation we suppose, for $v, w \in X$ and any $\mu \in \mathcal{D}$, that

$$a_{\mathrm{II}}(w, v; \mu) \leq \overline{\gamma} \parallel w \parallel_X \parallel v \parallel_{L^2} . \tag{23}$$

For our theororetical arguments below, we assume $\underline{\alpha} \leq 1$ and $\overline{\gamma} \geq 1$. The coercivity lower bound $\alpha_{\mathrm{LB}}(\mu)$ shall be given as $\alpha_{\mathrm{LB}}(\mu) = \underline{\alpha}$ for all $\mu \in \mathcal{D}$. *We emphasize that all our assumptions in this section are satisfied by our convection–diffusion numerical example of Section 5.1.*

We consider Algorithm 4.3 with $N_{\max} = \infty$. Hence *p*-refinement – execution of POD/Greedy in line 13 – will converge ($\epsilon_{\max} \leq \epsilon_{\mathrm{tol}}^2$) for any specified $\epsilon_{\mathrm{tol}}^2 > 0$. We thus focus here on *h*-refinement; we show in particular that the *hp*-POD/Greedy algorithm generates a finite number of parameter subdomains.

To this end, we shall require the following continuity result.

Lemma 4.1: *For any $\mu_1 \in \mathcal{D}$, $\mu_2 \in \mathcal{D}$, and any $v \in X$, $w \in X$, there exist positive constants c_a and c_f such that*

$$|a(v, w; \mu_1) - a(v, w; \mu_2)| \leq c_a |\mu_1 - \mu_2| \parallel v \parallel_X \parallel w \parallel_{L^2}, \tag{24}$$

$$|f(v; \mu_1) - f(v; \mu_2)| \leq c_f |\mu_2 - \mu_2| \parallel v \parallel_{L^2} . \tag{25}$$

Proof: We refer to Appendix A for the proof. □

We next define for any $\mu \in \mathcal{D}$ and any $v^k \in X$, $1 \leq k \leq K$, the 'energy-norm'

$$|||v^k|||_\mu \equiv \left(m(v^k, v^k) + \Delta t \sum_{k'=1}^{k} a(v^{k'}, v^{k'}; \mu) \right)^{1/2} . \tag{26}$$

To this end, we shall require the following stability result.

Lemma 4.2: *For any $\mu \in \mathcal{D}$, the solution $u^{\mathcal{N}\,k}(\mu) \in X^{\mathcal{N}}$, $1 \leq k \leq K$, of* (4) *for $\theta = 1$ satisfies*

$$|||u^{\mathcal{N}\,k}(\mu)|||_\mu \leq \max_{\mu \in \mathcal{D}} \parallel f(\cdot; \mu) \parallel_{X'} \sqrt{\frac{t^k}{\underline{\alpha}}}, \quad 1 \leq k \leq K. \tag{27}$$

Proof: We refer to Appendix B for the proof. □

For $\mu_1 \in \mathcal{D}, \mu_2 \in \mathcal{D}$ and $1 \leq k \leq K$, we define $\Delta u_N^k \equiv u_N^k(\mu_1) - u_N^k(\mu_2)$. We shall require the following continuity result.

Lemma 4.3: *Assume that $\mu_1 \in \mathcal{D}$ and $\mu_2 \in \mathcal{D}$ belong to the same parameter subdomain (say) $\mathcal{V}_{B_l} \subset \mathcal{D}$, and let X_N denote the RB space associated with $\mathcal{V}_{B_l}$. Let $u_N^k(\mu_1) \in X_N$ and $u_N^k(\mu_2) \in X_N$, $1 \leq k \leq K$, satisfy Equation* (5) *for $\theta = 1$. Then*

$$|||\Delta u_N^k|||_{\mu_2} \leq \tilde{C} |\mu_1 - \mu_2|, \quad 1 \leq k \leq K, \tag{28}$$

where

$$\tilde{C} = \left(\frac{2t^k}{\underline{\alpha}^3} \left(\underline{\alpha}^2 c_f^2 + c_a^2 \max_{\mu \in \mathcal{D}} \parallel f(\cdot;\mu) \parallel_{X'}^2 \right) \right)^{1/2}. \tag{29}$$

Proof: We refer to Appendix C for the proof. □

We shall finally require the following continuity result, which is a discrete counterpart of Proposition 11.1.11 of [18].

Lemma 4.4: *Assume that $\mu_1 \in \mathcal{D}$ and $\mu_2 \in \mathcal{D}$ belong to the same parameter subdomain $\big($say$\big)$ $\mathcal{V}_{B_l} \subset \mathcal{D}$, and let X_N denote the RB space associated with $\mathcal{V}_{B_l}$. Let $u_N^k(\mu_1) \in X_N$ and $u_N^k(\mu_2) \in X_N$, $1 \le k \le K$, satisfiy Equation* (5) *for $\theta = 1$. Then the finite difference $(\Delta u_N^k - \Delta u_N^{k-1})/\Delta t$ is L^2-bounded in time*:

$$\left(\frac{1}{\Delta t} \sum_{k'=1}^{k} \parallel \Delta u_N^{k'} - \Delta u_N^{k'-1} \parallel_{L^2}^2 \right)^{1/2} \le \hat{C} |\mu_1 - \mu_2|, \tag{30}$$

where

$$\hat{C} = \left(\frac{3}{\underline{\alpha}^2} \left(\overline{\gamma}^2 \underline{\alpha} \tilde{C}^2 + t^k \underline{\alpha}^2 c_f^2 + t^k c_a^2 \max_{\mu \in \mathcal{D}} \parallel f(\cdot;\mu) \parallel_{X'} \right) \right)^{1/2}. \tag{31}$$

Proof: We refer to Appendix D for the proof. □

We now claim

Proposition 4.1: *Let $\mathcal{D} \subset \mathbb{R}$ and let $|\mathcal{D}|$ denote the length of $\mathcal{D}$. For specified ϵ_{tol}^1, Algorithm 4.3 terminates for finite $M = M(\epsilon_{\text{tol}}^1)$ subdomains; moreover, the convergence of the* h*-refinement stage is first order in the sense that*

$$M(\epsilon_{\text{tol}}^1) \le \max \left\{ 1, \frac{C}{\epsilon_{\text{tol}}^1} \right\}, \qquad C = C(\eta, |\mathcal{D}|). \tag{32}$$

Proof: The proof has two steps. We first show that the RB error bound is Lipschitz continuous. We then relate this result to our particular procedure to prove convergence of the *hp*-POD/Greedy algorithm.

Step 1: We recall that for $\mu \in \mathcal{D}$, the Riesz representation $\hat{e}_N^k(\mu)$ of the residual $r_N^k(\cdot;\mu)$, $1 \le k \le K$, satisfies

$$(\hat{e}_N^k, v)_X = r_N^k(v;\mu), \quad \forall v \in X^{\mathcal{N}}. \tag{33}$$

Let $\mu_1 \in \mathcal{D}$, $\mu_2 \in \mathcal{D}$. We define $\Delta\hat{e}_N^k \equiv \hat{e}_N^k(\mu_1) - \hat{e}_N^k(\mu_2)$. From Equation (33), we note that by linearity

$$(\Delta\hat{e}_N^k, v)_X = \underbrace{f(v;\mu_1) - f(v;\mu_2)}_{\mathrm{I}} + \underbrace{a(u_N^k(\mu_2), v; \mu_2) - a(u_N^k(\mu_1), v; \mu_1)}_{\mathrm{II}} + \frac{1}{\Delta t}\underbrace{\left(m(u_N^k(\mu_2) - u_N^{k-1}(\mu_2), v) - m(u_N^k(\mu_1) - u_N^{k-1}(\mu_1), v)\right)}_{\mathrm{III}}, \tag{34}$$

for all $v \in X^{\mathcal{N}}$ and for $1 \le k \le K$. For term I, we invoke Lemma 4.1 directly to obtain

$$|f(v;\mu_1) - f(v;\mu_2)| \le c_f|\mu_1 - \mu_2| \parallel v \parallel_X, \quad \forall v \in X. \tag{35}$$

For term II, we first write

$$|a(u_N^k(\mu_2), v; \mu_2) - a(u_N^k(\mu_1), v; \mu_1)| = |a(u_N^k(\mu_1), v; \mu_2) - a(u_N^k(\mu_1), v; \mu_1) - a(\Delta u_N^k, v; \mu_2)|. \tag{36}$$

Then, by the triangle inequality, Lemma 4.1, continuity and Equation (22), we obtain

$$\begin{aligned} |a(u_N^k(\mu_2), v; \mu_2) - a(u_N^k(\mu_1), v; \mu_1)| &\le |a(\Delta u_N^k, v; \mu_2)| + c_a \parallel u_N^k(\mu_1) \parallel_X \parallel v \parallel_X |\mu_1 - \mu_2| \\ &\le \overline{\gamma} \parallel \Delta u_N^k \parallel_X \parallel v \parallel_X + c_a \parallel u_N^k(\mu_1) \parallel_X \parallel v \parallel_X |\mu_1 - \mu_2|. \end{aligned} \tag{37}$$

For term III, we invoke linearity, the Cauchy–Schwarz inequality and the Poincaré inequality[5] to obtain

$$\begin{aligned} |m(u_N^k(\mu_2) - u_N^{k-1}(\mu_2), v) - m(u_N^k(\mu_1) - u_N^{k-1}(\mu_1), v)| &= |m(\Delta u_N^k - \Delta u_N^{k-1}, v)| \\ &\le \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} \parallel v \parallel_{L^2} \le \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} \parallel v \parallel_X . \end{aligned} \tag{38}$$

We now insert the expressions for terms I, II and III into Equation (34); for $v = \Delta\hat{e}_N^k$ we then obtain

$$\begin{aligned} (\Delta\hat{e}_N^k, \Delta\hat{e}_N^k)_X \le\ & c_f|\mu_1 - \mu_2| \parallel \Delta\hat{e}_N^k \parallel_X + \overline{\gamma} \parallel \Delta u_N^k \parallel_X \parallel \Delta\hat{e}_N^k \parallel_X \\ & + c_a \parallel u_N^k(\mu_1) \parallel_X \parallel \Delta\hat{e}_N^k \parallel_X |\mu_1 - \mu_2| + \frac{1}{\Delta t} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} \parallel \Delta\hat{e}_N^k \parallel_X . \end{aligned} \tag{39}$$

We divide through in Equation (39) by $\parallel \Delta\hat{e}^k \parallel_X$, square both sides and invoke the inequality $(A + B + C + D)^2 \le 4(A^2 + B^2 + C^2 + D^2)$ for $A, B, C, D \in \mathbb{R}$ to obtain

$$\parallel \Delta\hat{e}^k \parallel_X^2 \le 4|\mu_1 - \mu_2|^2 (c_f^2 + c_a^2 \parallel u_N^k(\mu_1) \parallel_X^2) + \frac{4}{\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2}^2 + 4\overline{\gamma}^2 \parallel \Delta u_N^k \parallel_X^2 . \tag{40}$$

We multiply through in Equation (40) by Δt, substitute k for k' and sum over k' to obtain

$$\begin{aligned} \Delta t \sum_{k'=1}^{k} \parallel \Delta\hat{e}^{k'} \parallel_X^2 \le\ & 4|\mu_1 - \mu_2|^2 \left(c_f^2 t^k + c_a^2 \Delta t \sum_{k'=1}^{k} \parallel u_N^{k'}(\mu_1) \parallel_X^2 \right) \\ & + 4\overline{\gamma}^2 \left(\frac{1}{\Delta t} \sum_{k'=1}^{k} \parallel \Delta u_N^{k'} - \Delta u_N^{k'-1} \parallel_{L^2}^2 + \Delta t \sum_{k'=1}^{k} \parallel \Delta u_N^{k'} \parallel_X^2 \right). \end{aligned} \tag{41}$$

Next, from coercivity and Lemma 4.2 we note that

$$\Delta t \sum_{k'=1}^{k} \| u_N^{k'}(\mu_1) \|_X^2 \leq \frac{|||u_N^k(\mu_1)|||_{\mu_1}^2}{\underline{\alpha}} \leq \frac{t^k}{\underline{\alpha}^2} \max_{\mu \in \mathcal{D}} \| f(\cdot;\mu) \|_{X'}^2 . \tag{42}$$

Furthermore, from coercivity and Equation (22), and Lemmas 4.3 and 4.4, we note that

$$\begin{aligned} &4\overline{\gamma}^2 \left(\frac{1}{\Delta t} \sum_{k'=1}^{k} \| \Delta u_N^{k'} - \Delta u_N^{k'-1} \|_{L^2}^2 + \Delta t \sum_{k'=1}^{k} \| \Delta u_N^{k'} \|_X^2 \right) \\ &\leq 4\overline{\gamma}^2 \left(\frac{1}{\Delta t} \sum_{k'=1}^{k} \| \Delta u_N^{k'} - \Delta u_N^{k'-1} \|_{L^2}^2 \right. \\ &\qquad \left. + \Delta t \sum_{k'=1}^{k} \frac{a(\Delta u_N^{k'}, \Delta u_N^{k'}; \mu_2)}{\underline{\alpha}} \right) \leq 4\overline{\gamma}^2 |\mu_1 - \mu_2|^2 \left(\hat{C}^2 + \frac{\tilde{C}^2}{\underline{\alpha}} \right). \end{aligned} \tag{43}$$

From Equation (41) with Equations (42) and (43), we thus obtain

$$\Delta t \sum_{k'=1}^{k} \| \Delta \hat{e}^k \|_X^2 \leq c^2 |\mu_1 - \mu_2|^2, \tag{44}$$

where

$$c \equiv 2 \left(\frac{t^k}{\underline{\alpha}^2} \left(\underline{\alpha}^2 c_f^2 + c_a^2 \max_{\mu \in \mathcal{D}} \| f(\cdot;\mu) \|_{X'}^2 \right) + \overline{\gamma}^2 \left(\tilde{C}^2 + \frac{\hat{C}^2}{\underline{\alpha}} \right) \right)^{1/2} . \tag{45}$$

By the definition of the RB error bound (recall that we use $\alpha_{\mathrm{LB}}(\mu) = \underline{\alpha}$) and the reverse triangle inequality, we finally obtain

$$\begin{aligned} |\Delta_N^k(\mu_1) - \Delta_N^k(\mu_2)| &\leq \left| \left(\frac{\Delta t}{\underline{\alpha}} \sum_{k'=1}^{k} \| \hat{e}_N^k(\mu_1) \|_X^2 \right)^{1/2} - \left(\frac{\Delta t}{\underline{\alpha}} \sum_{k'=1}^{k} \| \hat{e}_N^k(\mu_2) \|_X^2 \right)^{1/2} \right| \\ &\leq \left(\frac{\Delta t}{\underline{\alpha}} \sum_{k'=1}^{k} \| \Delta \hat{e}_N^k \|_X^2 \right)^{1/2} \leq \frac{c}{\sqrt{\underline{\alpha}}} |\mu_1 - \mu_2|. \end{aligned} \tag{46}$$

Step 2: The next step is to relate Equation (46) to the convergence of Algorithm 4.3. The algorithm generates a partition of $\mathcal{D}$ into M subdomains. Either $M = 1$, in which case the proof is complete, or $M > 1$. We now examine the case $M > 1$. We consider the splitting of any particular subdomain $\mathcal{V}_{B_l} \subset \mathcal{D}$ into two new subdomains $\mathcal{V}_{(B_l,0)} \subset \mathcal{V}_{B_l}$ and $\mathcal{V}_{(B_l,1)} \subset \mathcal{V}_{B_l}$. We denote here by $\hat{\mu}_0 = \hat{\mu}_{B_l}^0 = \hat{\mu}_{(B_l,0)}^0$ the anchor point associated with $\mathcal{V}_{B_l}$ and $\mathcal{V}_{(B_l,0)}$, and by $\hat{\mu}_1 = \hat{\mu}_{B_l}^1 = \hat{\mu}_{(B_l,1)}^0$ the anchor point associated with $\mathcal{V}_{(B_l,1)}$. We assume that the error tolerance at the final time is not satisfied over (a train sample over) $\mathcal{V}_{B_l}$; hence $\epsilon_{\mathrm{tol}}^1 \leq \tilde{\Delta}_{R,B_l}^K(\hat{\mu}_1)$. We recall that by construction of our procedure $\tilde{\Delta}_{R,B_l}^K(\hat{\mu}_0) \leq \epsilon_{\mathrm{tol}}^1/\eta$ for specified $\eta \geq 1$. We can thus invoke Equation (46) for $\mu_1 = \hat{\mu}_1$, $\mu_2 = \hat{\mu}_0$ and Δ_N^k replaced by $\tilde{\Delta}_{R,B_l}^K$ to conclude that

$$\epsilon^1_{\text{tol}} - \frac{\epsilon^1_{\text{tol}}}{\eta} < |\tilde{\Delta}^K_{R,B_l}(\hat{\mu}_1) - \tilde{\Delta}^K_{R,B_l}(\hat{\mu}_0)| \leq \frac{c}{\sqrt{\underline{\alpha}}}|\hat{\mu}_1 - \hat{\mu}_0|, \tag{47}$$

and hence

$$|\hat{\mu}_1 - \hat{\mu}_0| \geq \frac{\epsilon^1_{\text{tol}}\sqrt{\underline{\alpha}}(\eta-1)}{c\eta}. \tag{48}$$

We now split $\mathcal{V}_{B_l}$ into $\mathcal{V}_{(B_l,0)}$ and $\mathcal{V}_{(B_l,1)}$ based on Euclidean distance to the two anchor points. It is clear that

$$|\mathcal{V}_{(B_l,i)}| \geq \frac{1}{2}|\hat{\mu}_1 - \hat{\mu}_0| > \frac{\epsilon^1_{\text{tol}}\sqrt{\underline{\alpha}}(\eta-1)}{2c\eta}, \quad i = 0, 1. \tag{49}$$

The partition procedure generates $M > 1$ distinct subdomains $\mathcal{V}_{B^m}$, $1 \leq m \leq M$.[6] Each of these subdomains is the result of a splitting of a 'parent' subdomain $\mathcal{V}_{B_l} \supset \mathcal{V}_{B^m}$ (for some $B_l, 0 \leq l \leq L-1$). As B_l above was arbitrary, we can successively set $\mathcal{V}_{B_l}$ to be the parent of each of the M 'leaf' subdomains and conclude that

$$|\mathcal{V}_{B^m}| > \frac{\epsilon^1_{\text{tol}}\sqrt{\underline{\alpha}}(\eta-1)}{2c\eta}, \quad 1 \leq m \leq M. \tag{50}$$

We define $\delta_M \equiv \min_{1\leq m\leq M} |\mathcal{V}_{B^m}|$; hence in particular $\delta_M \geq \dfrac{\epsilon^1_{\text{tol}}\sqrt{\underline{\alpha}}(\eta-1)}{2c\eta}$.

We complete the proof by a contradiction argument. Assume that $M > \dfrac{|\mathcal{D}|2c\eta}{\epsilon^1_{\text{tol}}\sqrt{\underline{\alpha}}(\eta-1)}$. Thus

$$M\delta_M > \frac{|\mathcal{D}|2c\eta}{\epsilon^1_{\text{tol}}\sqrt{\underline{\alpha}}(\eta-1)} \frac{\epsilon^1_{\text{tol}}\sqrt{\underline{\alpha}}(\eta-1)}{2c\eta} = |\mathcal{D}|, \tag{51}$$

which is clearly a false statement. We conclude that $M = M(\epsilon^1_{\text{tol}}) \leq C(\eta, |\mathcal{D}|)/\epsilon^1_{\text{tol}}$ with $C(\eta,|\mathcal{D}|) = \frac{|\mathcal{D}|2c\eta}{\sqrt{\underline{\alpha}}(\eta-1)}$. We finally note that Algorithm 4.3 is convergent because the POD/Greedy (line 13) will be able to satisfy the error bound tolerance ϵ^2_{tol} within each of the M final subdomains. □

Remark 4.3: The requirement $\eta > 1$ reappears in the proof in Equation (47). We note that we cannot obtain a positive lower bound for the distance between the two anchor points if $\eta \leq 1$.

Remark 4.4: If we assume only $f \in X'$ (and not in L^2) and furthermore a_{II} only X-continuous in both arguments (and not L^2-continuous in the second argument), then we can still obtain Proposition 4.1 albeit with an additional factor $1/\Delta t$ in the 'constant' C. However, we note that this $1/\Delta t$ factor is in this case relatively 'benign': we cannot in any event let '$\Delta t \to 0$' in practice because of the increase in Online computational cost. (In contrast, we can let '$\mathcal{N} \to \infty$' because larger $\mathcal{N}$ affects only Offline cost.)

We recall that all the hypotheses of Proposition 4.1 are satisfied by our numerical example in Section 5.1.

Remark 4.5: Proposition 4.1 guarantees that the partition algorithm (*h*-refinement) is convergent. However, the convergence is very slow and hence subsequent *p*-refinement is in practice necessary. But note that with only a global Lipschitz constant c in our proof, our bound (32) is very pessimistic and in particular does not reflect any adaptivity in the partition. In practice, we expect that the algorithm adaptively generates smaller subdomains in areas of $\mathcal{D}$ for which the field exhibits larger variations with the parameters.

5. Numerical results

We now present numerical results for two model problems. We demonstrate that in both cases the *hp*-RB method yields significant Online computational savings relative to a standard (*p*-type) RB approach; we also show that the partitions of $\mathcal{D}$ may reflect the underlying parametric sensitivity of the problems. All our computational results are obtained via `rbOOmit` [19], which is an RB plugin for the open-source FE library `libMesh` [20]. All computations are performed on a 2.66-GHz processor. For the *hp*-RB approximations below, we have used a 'scaled' Euclidean distance for the distance function $\delta(\cdot,\cdot)$: we map $\mathcal{D}$ (a rectangle in both our examples) to $\hat{\mathcal{D}} = [0,1]^P$ (via an obvious affine transformation) and compute the Euclidean distance on $\hat{\mathcal{D}}$. For the constant η in Algorithm 4.3, we choose $\eta = 1.1$.

5.1. *Convection–[diffusion problem]*

We consider the non-dimensional temperature u, which satisfies the convection–diffusion equation in the spatial domain $\Omega = \{(x_1, x_2) : x_1^2 + x_2^2 < 2\}$ for the discrete time levels $t^k = 0.01k$, $0 \le k \le 100$; we employ Backward Euler temporal discretization (hence $\theta = 1$). We impose a parameter-dependent velocity field $V(\mu) \equiv V(\nu,\varphi) \equiv (\nu\cos\varphi, \nu\sin\varphi)$ and we prescribe a constant forcing term $q = 10$. We specify homogeneous Dirichlet boundary conditions and zero initial conditions. We denote a particular parameter value $\mu \in \mathcal{D}$ by $\mu = (\nu,\varphi)$ and we introduce the parameter domain $\mathcal{D} = [0,10] \times [0,\pi] \subset \mathbb{R}^{P=2}$. For this problem, we focus for simplicity on the RB field approximation and thus we do not consider any particular outputs.

We next introduce the forms

$$
\begin{aligned}
m(w,v;\mu) &= \int_\Omega wv,\\
a(w,v;\mu) &= \int_\Omega (\nabla w\cdot\nabla v + (V(\mu)\cdot\nabla w)v),\\
f(v;\mu) &= q\int_\Omega v = 10\int_\Omega v,
\end{aligned}
\tag{52}
$$

for $v, w \in X$, where $X = H_0^1(\Omega)$. Our problem can then be expressed in the form (4) with $b = 0$; note that our only parameter-dependent form is a, which admits an affine expansion (3) with $Q_a = 3$. We note that this problem satisfies all the theoretical hypothesis of Proposition 4.1.[7] For our truth approximation, we choose a $\mathbb{P}_2$ FE space $X^{\mathcal{N}} \subset X$ of dimension $\mathcal{N} = 1889$.

To obtain a benchmark for comparison, we first perform a standard (*p*-type) POD/Greedy: we specify $\epsilon = 10^{-5}$ for the target tolerance, $\Delta N = 1$ for the number of POD modes to include at each greedy iteration, $\mu^* = (0,0)$ for the initial parameter value

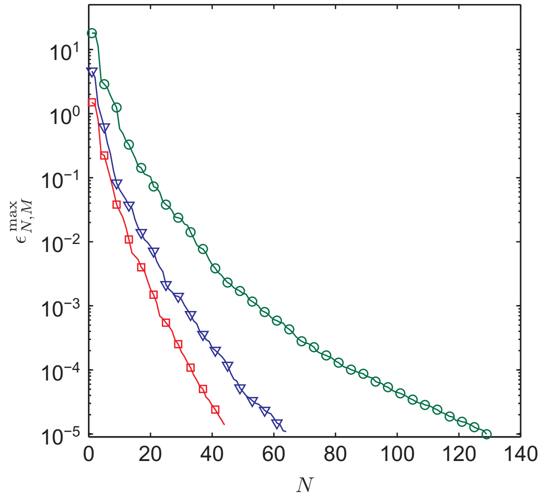


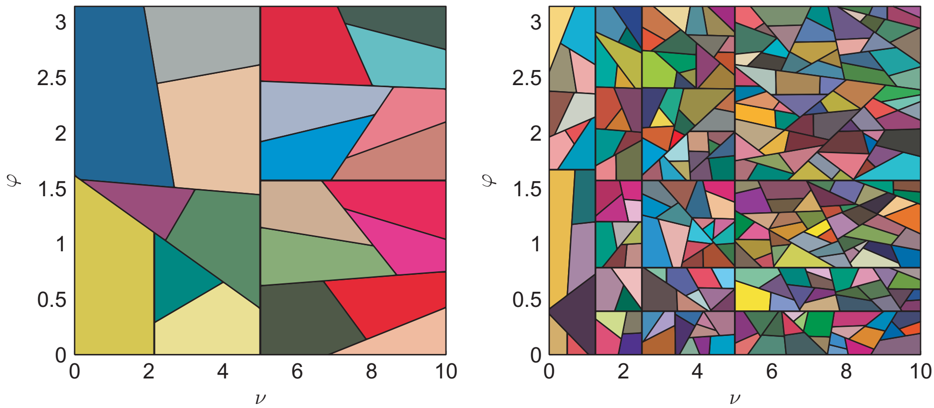
Figure 2. Convergence: *hp*-RB (triangles ($M = 22$) and squares ($M = 278$)) and *p*-type RB (circles). In the *hp*-RB cases, the error bound is the maximum over all subdomains for a given N.

and a train sample $\Xi_{\text{train}} \subset \mathcal{D}$ of size 900. We then execute Algorithm 4.2 (we also 'specify' $\overline{N} = \infty$ such that the POD/Greedy terminates for ϵ satisfied over Ξ_{train}). The tolerance is in this case satisfied for $N_{\max} = \tilde{N}_{\max} = 129$.

We next perform two *hp*-POD/Greedy computations. In the first, we specify $\epsilon_{\text{tol}}^1 = 5$, $\epsilon_{\text{tol}}^2 = 10^{-5}$, $\overline{N} = 65$, $\Delta N = 1$, $\hat{\mu}_{(1)}^0 = (0, 0)$ and a train sample $\Xi_{\text{train},(1)}$ of size 64. In this case, Algorithm 4.3 terminates for $M = 22$ subdomains with $N_{\max} = \overline{N} = 65$ (recall that $N_{\max} \equiv \max_{1\le m\le M} N_{\max,B^m}$). In the second case, we specify $\epsilon_{\text{tol}}^1 = 1.5$, $\epsilon_{\text{tol}}^2 = 10^{-5}$, $\overline{N} = 45$, $\Delta N = 1$, $\hat{\mu}_{(1)}^0 = (0, 0)$ and a train sample $\Xi_{\text{train},(1)}$ of size 25. In this case, Algorithm 4.3 terminates for $M = 278$ subdomains with $N_{\max} = \overline{N} = 45$. The maximum RB $L^2(\Omega)$ error bound $\epsilon_{N,M}^{\max}$ (over the train samples) over all M subdomains for each of the cases $M = 22$ and 278, as well as the *p*-type reference case $M = 1$, are plotted in Figure 2 as functions of N. We note that larger M yields smaller N, as desired.

We show the two partitions of $\mathcal{D}$ in Figure 3.[8] Note that the field variable exhibits larger variations with φ for larger ν, and hence we would expect the subdomain size to decrease with increasing ν. However, this is not the case in Figure 3(b) except for smaller ν. By way of explanation, we note that when the field varies significantly with *time*, which is indeed the case for large ν, R – the number of POD modes in the temporary space $\tilde{X}_{R,B_l}$ – will be larger. We suspect that the additional POD modes included in the $\tilde{X}_{R,B_l}$ associated with subdomains for ν larger than approximately 5 may also represent some parametric variations in the field and hence account for the 'non-monotonic' (in ν) subdomain size.

We note that the *hp*-RB method indeed yields a significant Online speedup. Online *p*-type RB calculation of the RB solution coefficients and error bound for $N = 129$ basis functions requires 1.4×10^{-2} seconds. In contrast, Online *hp*-RB calculation of the RB solution coefficients and error bound for the case with $M = 22$ subdomains and $N = 65$ requires 3.3×10^{-3} seconds, and for the case with $M = 278$ subdomains and $N = 45$ requires 1.8×10^{-3} seconds; in both cases, the search for the subdomain containing the new online parameter is negligible ($\mathcal{O}(10^{-6})$ seconds). (The timing results are averages over 100 Online calculations for randomly selected $\mu \in \mathcal{D}$.)



(a) $M = 22$ (b) $M = 278$

Figure 3. Parameter domain partitions $\mathcal{V}_{Bm}$, $1 \leq m \leq M$, for the convection–diffusion problem.

Of course, Offline cost and Online storage are larger for the *hp*-RB than for the standard (p-type) RB: the Offline stage requires 29.6 minutes and 3.5 hours for the *hp*-RB computations ($M = 22$ and $M = 278$, respectively) and only 13.4 minutes for the standard RB; the Online Dataset requires 25.3 MB and 142.9 MB for the *hp*-RB computations ($M = 22$ and $M = 278$, respectively) and only 5.7 MB for the standard RB. In particular, Offline cost for the $M = 278$ computation is admittedly very large compared with the Offline cost for the p-type computation. Of course, even in our 'real time' and 'many query' contexts, the larger Offline cost associated with the *hp*-RB method may be an issue; we must thus seek to balance the increase in Offline cost against the decrease in Online cost by appropriate choices of the parameters $\epsilon^1_{\mathrm{tol}}$ and $\overline{N}$. We note that for this problem, our $M = 22$ *hp*-RB computation provides significant Online speedup at only modest increase in Offline cost.

The additional splitting step – the "insurance" provided by lines 15–18 in Algorithm 4.3 – was never invoked for either *hp*-POD/Greedy computation. For the computation with specified $\overline{N} = 65$, the average of $N_{\max,B^m}$, $1 \leq m \leq M = 22$, is 57.3. For the computation with specified $\overline{N} = 45$, the average of $N_{\max,B^m}$, $1 \leq m \leq M = 278$, is 37.9. We conclude that in both cases we could have chosen $\epsilon^1_{\mathrm{tol}}$ somewhat larger (at the risk of invoking insurance) to obtain a more optimal partition with respect to the target $\overline{N}$.

We finally note that calculation of the truth (4) for this problem with $\mathcal{N} = 1889$ requires about 0.9 seconds. The average speedup relative to a truth calculation is approximately 64 for the p-type Online calculation with $N = 129$, and approximately 273 and 500 for the *hp*-RB Online calculations ($N = 65$, $M = 22$ and $N = 45$, $M = 278$, respectively).

5.2. *Boussinesq problem*

We consider natural convection in the two-dimensional enclosure $\Omega = (0,5)^2 \backslash \mathcal{P}$, where $\mathcal{P}$ is the 'pillar' $(2.5 - 0.1, 2.5 + 0.1) \times (0, 1)$, for the discrete time levels $t^k = 0.0016k$, $0 \leq k \leq 100$; we employ Crank–Nicolson temporal discretization (hence $\theta = 0.5$). The direction of the acceleration of gravity is defined by the unit vector $(-\sin\phi, -\cos\phi)$. We solve for the field variables V_1, V_2 (the x and y components of the fluid velocity) and ϑ

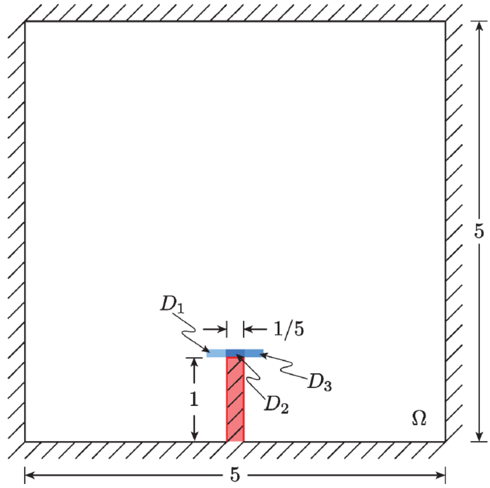


Figure 4. The computational domain; note that Ω does not include the pillar, which is shaded. The output regions D_1, D_2 and D_3 are also indicated.

(the temperature) over Ω; hence the field has dimension $d = 3$. The 'roof' of the enclosure is maintained at temperature $\vartheta = 0$, the sides and base of the enclosure are perfectly thermally insulated and the top and sides of the pillar are subject to a uniform heat flux of magnitude Gr (the Grashof number); we impose no-slip velocity conditions on all walls. We denote a particular parameter value $\mu \in \mathcal{D}$ by $\mu = (\mu_1, \mu_2) = (Gr, \phi)$ and we introduce the parameter domain $\mathcal{D} = [4000, 6000] \times [0, 0.2] \subset \mathbb{R}^{P=2}$. Note that we set the Prandtl number, Pr, here to 0.71 (for air).

Our goal is to study parametric dependence of the temperature in regions at or near the top of the heated pillar (or 'fin') in the presence of natural convection, and hence we are interested in local average-temperature outputs. These outputs can be expressed as $L^2(\Omega)$-bounded functionals of ϑ, namely,

$$s_n(t;\mu) = \ell_n(\vartheta(t;\mu), \mu) = \frac{1}{\mu_1 |D_n|} \int_{D_n} \vartheta(t;\mu)\,; \tag{53}$$

here $D_1 = [2.2, 2.4] \times [1, 1.1]$, $D_2 = [2.4, 2.6] \times [1, 1.1]$, $D_3 = [2.6, 2.8] \times [1, 1.1]$ are three small rectangles above the pillar. The domain geometry and output regions are depicted in Figure 4.

We introduce the forms

$$\begin{aligned}
m(w, v; \mu) &= \textstyle\int_\Omega w_i v_i, \\
a(w, v; \mu) &= \textstyle\int_\Omega \left(\frac{\partial w_1}{\partial x_j} \frac{\partial v_1}{\partial x_j} + \frac{\partial w_2}{\partial x_j} \frac{\partial v_2}{\partial x_j} + \frac{1}{Pr} \frac{\partial w_3}{\partial x_j} \frac{\partial v_3}{\partial x_j} \right), \\
b_1(w, v; \mu) &= -\sqrt{\mu_1 Pr} \sin \mu_2 \textstyle\int_\Omega w_3 v_1 - \sqrt{\mu_1 Pr} \cos \mu_2 \int_\Omega w_3 v_2, \\
b_2(w, z, v; \mu) &= \tfrac{1}{2\sqrt{\mu_1 Pr}} \textstyle\int_\Omega \left(\frac{\partial w_i z_j}{\partial x_j} + z_j \frac{\partial w_i}{\partial x_j} \right) v_i, \\
f(v; \mu) &= \tfrac{\mu_1}{Pr} \textstyle\int_{\partial\Omega_p} v_3,
\end{aligned} \tag{54}$$

for $w = (w_1, w_2, w_3) \in X$, $v = (v_1, v_2, v_3) \in X$ and $z = (z_1, z_2, z_3) \in X$; in these expressions, $i = 1, 2, 3$ and $j = 1, 2$. Here, $X = Z \times W$, where Z is the divergence-free subspace

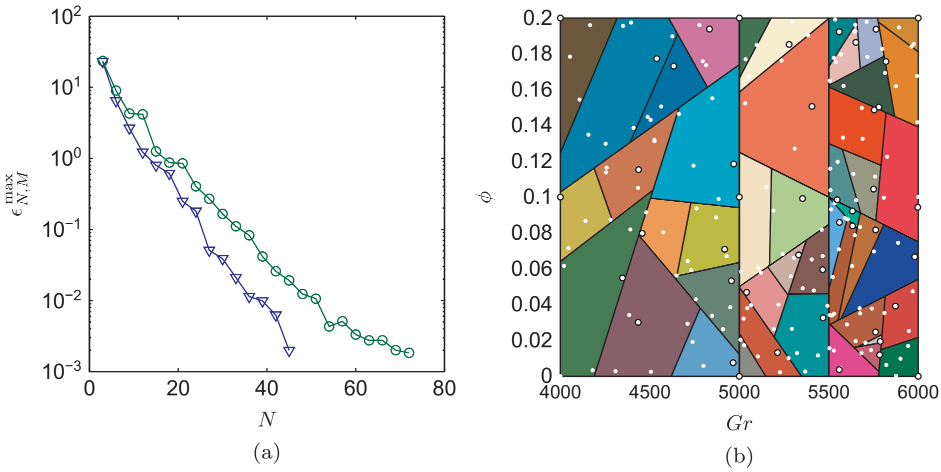


Figure 5. (a) Convergence: *hp*-RB (triangles) and *p*-type RB (circles). (b) Parameter domain partition: we show the anchor point (a circled white dot) and the Greedily selected parameters (white dots) in each subdomain; note that, within a subdomain, parameters are often selected more than once by the POD/Greedy algorithm.

of $(H_0^1(\Omega))^2$, and $H_0^1(\Omega) \subset W \subset H^1(\Omega)$ is the subspace of $H^1(\Omega)$ of functions that vanish on the enclosure roof.

Our problem can then be expressed in the form (4) with $b(w,z,v;\mu) = b_1(w,v;\mu) + b_2(w,z,v;\mu)$ (we have used a skew-symmetric form of the non-linear convection operator $b_2(w,z,v;\mu)$ to generate certain discrete stability properties [18]); note that all forms satisfy the 'affine' assumption. For our truth FE space, we choose $X^{\mathcal{N}} = Z^{\mathcal{N}} \times W^{\mathcal{N}}$ of dimension $\mathcal{N} = 7248$, where $Z^{\mathcal{N}}$ denotes a discretely divergence-free $\mathbb{P}_2$ space for the velocity (developed from the $\mathbb{P}_2 - \mathbb{P}_1$ Taylor–Hood velocity–pressure approximation) and $W^{\mathcal{N}}$ is a standard $\mathbb{P}_2$ FE space for the temperature. For further details on the formulation of this problem, see [11].

We note that for the computational results for this problem, we consider a '*relative* $L^2(\Omega)$ error bound' version of Algorithm 4.2 and hence Algorithm 4.3. To obtain a benchmark for comparison, we first perform a standard (*p*-type) POD/Greedy computation: we specify $\epsilon = 2 \times 10^{-3}$ for the target tolerance, $\Delta N = 3$ for the number of POD modes to include at each Greedy iteration, $\mu^* = (6000, 0)$ for the initial parameter value and a train sample Ξ_{train} of size 200. In this case, Algorithm 4.2 terminates for $N_{\max} = \tilde{N}_{\max} = 72$. Recall that in the quadratically non-linear case, the POD/Greedy terminates when the *nominal* error bound reaches the prescribed tolerance.

We then perform an *hp*-POD/Greedy computation: we specify $\epsilon_{\text{tol}}^1 = 1.2$, $\epsilon_{\text{tol}}^2 = 2 \times 10^{-3}$, $\overline{N} = 45$, $\Delta N = 3$, $\hat{\mu}_{(1)}^0 = (6000, 0)$ and a train sample $\Xi_{\text{train},(1)}$ of size 9. In this case, Algorithm 4.2 terminates after generation of $M = 45$ subdomains with $N_{\max} = 45$. The maximum relative RB $L^2(\Omega)$ error bound $\epsilon_{N,M}^{\max}$ (over the train samples) over all subdomains for the *hp*-RB approximation as well as for the *p*-type RB approximation are shown in Figure 5(a). As in the linear case, the *hp* approach trades reduced N for increased M. We show the *hp*-RB parameter domain partition in Figure 5(b).

In Figure 6, we show for $N = 45$ the RB output approximations to the three outputs (53) for three parameter values $(Gr, \phi) = (4000, 0.05)$, $(Gr, \phi) = (5000, 0.1)$ and $(Gr, \phi) = (6000, 0.2)$. We also indicate the corresponding error bars $[s_{N,j}^k(\mu) - \Delta_{N,s_j}^k(\mu), s_{N,j}^k(\mu) +$

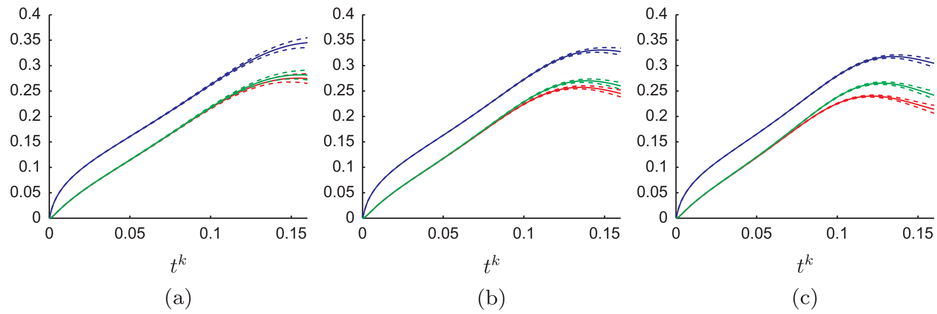


Figure 6. The RB outputs $s_{N,1}(t^k;\mu)$ (bottom, solid line), $s_{N,2}(t^k;\mu)$ (top, solid line), $s_{N,3}(t^k;\mu)$ (middle, solid line), and associated error bars (dashed lines) as functions of time for three values of μ. (a) $(Gr,\phi) = (4000, 0.05)$ (b) $(Gr,\phi) = (5000, 0.1)$ (c) $(Gr,\phi) = (6000, 0.2)$

$\Delta^k_{N,s_j}(\mu)]$, $1 \le k \le K$, $1 \le j \le 3$, in which the true result $s_j^{\mathcal{N}\,k}$ *must reside*. We recall that the RB output error bounds Δ_{N,s_j} are obtained as the product of the RB field error bound Δ_N and the dual norm of the output functional (Equation (11)).[9] We remark that the accuracy of these *hp*-RB outputs is comparable with the accuracy of the *p*-type RB outputs because the *hp*-POD/Greedy and *p*-type POD/Greedy calculations terminate for the same specified tolerance. Note that time is measured in diffusive units and hence the final time of 0.16 is sufficient to observe (at these *Gr*) significant non-linear effects.

The standard (*p*-type) RB method yields a significant Online speedup relative to the expensive Boussinesq truth FE solves (one truth solve requires 239 seconds); nevertheless, these *p*-type RB computations are still rather expensive due to the $O(N^4)$ complexity of the RB error bound for quadratically non-linear problems. The *hp*-POD/Greedy method of this article provides a significant additional speedup in the *hp*-RB Online stage due to the direct control of $N_{\max}$ and hence reduction in N: Online *p*-type RB calculation of the output and error bound with $N = 72$ basis functions requires 6.48 seconds, whereas Online *hp*-RB calculation of the output and error bound with $M = 45$ subdomains and $N = 45$ requires only 0.845 seconds. Of course, Offline cost and Online storage are larger for the *hp*-RB than for the standard RB: the Offline stage requires about 69 hours for the *hp*-RB and only about 5.2 hours for the standard RB; the Online Dataset requires 2.3 GB for the *hp*-RB and only 481 MB for the standard RB.

We finally note that the additional splitting step ('insurance') was invoked for 10 subdomains for the *hp*-POD/Greedy computation, and the average of $N_{\max,B^m}$, $1 \le m \le M$, is 40.1. This suggests that ϵ^1_{tol} in this case was reasonably well chosen with respect to the target $\overline{N}$.

Appendix A. Proof of Lemma 4.1

From Equations (16), (18) and (20), we obtain Equation (24) with $c_a = Q_a \max_{2\le q\le Q_a}(\gamma^q L^q_a)$. From Equations (15), (19) and (21), we obtain Equation (25) with $c_f = Q_f \max_{1\le q\le Q_f}(\| f^q \|_{L^2} L^q_f)$.

Appendix B. Proof of Lemma 4.2

From Equation (4) with $v = u^{\mathcal{N}\,k}(\mu)$, we obtain

$$\frac{1}{\Delta t}m(u^{\mathcal{N}k}(\mu),u^{\mathcal{N}k}(\mu))+a(u^{\mathcal{N}k}(\mu),u^{\mathcal{N}k}(\mu);\mu)=\frac{1}{\Delta t}m(u^{\mathcal{N}k-1}(\mu),u^{\mathcal{N}k}(\mu))+f(u^{\mathcal{N}k}(\mu);\mu). \tag{B1}$$

We next recall Young's inequality $AB \leq (A^2/\kappa + \kappa B^2)/2$ (for $A, B, \kappa \in R$). For the first term on the right, we first invoke the Cauchy–Schwarz inequality and then Young's inequality for $A = m(u^{\mathcal{N}k-1}(\mu), u^{\mathcal{N}k-1}(\mu))^{1/2}$, $B = m(u^{\mathcal{N}k}(\mu), u^{\mathcal{N}k}(\mu))^{1/2}$ and $\kappa = 1$ to obtain

$$\begin{aligned} m(u^{\mathcal{N}k-1}(\mu),u^{\mathcal{N}k}(\mu)) &\leq m(u^{\mathcal{N}k-1}(\mu),u^{\mathcal{N}k-1}(\mu))^{1/2}m(u^{\mathcal{N}k}(\mu),u^{\mathcal{N}k}(\mu))^{1/2}\\ &\leq \frac{1}{2}\Big(m(u^{\mathcal{N}k-1}(\mu),u^{\mathcal{N}k-1}(\mu))+m(u^{\mathcal{N}k}(\mu),u^{\mathcal{N}k}(\mu))\Big). \end{aligned} \tag{B2}$$

For the second term on the right, we first invoke boundedness of $f(\cdot;\mu)$ and then Young's inequality with $A = \| f(\cdot;\mu)\|_{X'}$, $B = \| u^{\mathcal{N}k}(\mu)\|_X$ and $\kappa = \alpha(\mu)$ to obtain

$$\begin{aligned} f(u^{\mathcal{N}k}(\mu);\mu) \leq \| f(\cdot;\mu)\|_{X'}\| u^{\mathcal{N}k}(\mu)\|_X &\leq \frac{1}{2}\left(\frac{\| f(\cdot;\mu)\|^2_{X'}}{\alpha(\mu)}+\alpha(\mu)\| u^{\mathcal{N}k}(\mu)\|^2_X\right)\\ &\leq \frac{1}{2}\left(\frac{\| f(\cdot;\mu)\|^2_{X'}}{\alpha(\mu)}+a(u^{\mathcal{N}k}(\mu),u^{\mathcal{N}k}(\mu);\mu)\right), \end{aligned} \tag{B3}$$

where the last step follows from coercivity of $a(\cdot,\cdot;\mu)$. We combine Equations (B2) and (B3) with (B1), invoke Equation (22), substitute k' for k and sum over k' to obtain Equation (27).

Appendix C. Proof of Lemma 4.3

From linearity of Equation (5), we obtain, for $1 \leq k \leq K$,

$$\begin{aligned} \frac{1}{\Delta t}m(\Delta u_N^k-\Delta u_N^{k-1},v)+a(\Delta u_N^k,v;\mu_2) &= f(v;\mu_1)-f(v;\mu_2)\\ &\quad + a(u_N^k(\mu_1),v;\mu_2)-a(u_N^k(\mu_1),v;\mu_1), \quad \forall v\in X_N. \end{aligned} \tag{C1}$$

Next, from Lemma 4.1 we obtain

$$\begin{aligned} &\frac{1}{\Delta t}m(\Delta u_N^k-\Delta u_N^{k-1},v)+a(\Delta u_N^k,v;\mu_2)\\ &\quad = f(v;\mu_1)-f(v;\mu_2)+a(u_N^k(\mu_1),v;\mu_2)-a(u_N^k(\mu_1),v;\mu_1)\\ &\quad \leq c_f|\mu_1-\mu_2|\,\| v\|_X + c_a|\mu_1-\mu_2|\,\| u_N^k(\mu_1)\|_X\| v\|_X . \end{aligned} \tag{C2}$$

For the first term on the right, we invoke Young's inequality for $A = c_f|\mu_1-\mu_2|$, $B = \| v\|_X$ and $\kappa = \underline{\alpha}/2$ to note that

$$c_f|\mu_1-\mu_2|\,\| v\|_X \leq \frac{1}{2}\left(\frac{2c_f^2}{\underline{\alpha}}|\mu_1-\mu_2|^2+\frac{\underline{\alpha}}{2}\| v\|^2_X\right) \leq \frac{c_f^2}{\underline{\alpha}}|\mu_1-\mu_2|^2+\frac{1}{4}a(v,v;\mu_2), \tag{C3}$$

where the second inequality follows from coercivity of $a(\cdot,\cdot;\mu_2)$. For the second term on the right, we invoke Young's inequality for $A = c_a|\mu_1-\mu_2|\,\| u_N^k(\mu_1)\|_X$, $B = \| v\|_X$ and $\kappa = \underline{\alpha}/2$ to note that

$$c_a|\mu_1 - \mu_2| \parallel u_N^k(\mu_1) \parallel_X \parallel v \parallel_X \leq \frac{1}{2}\left(\frac{2c_a^2}{\underline{\alpha}}|\mu_1 - \mu_2|^2 \parallel u_N^k(\mu_1) \parallel_X^2 + \frac{\underline{\alpha}}{2} \parallel v \parallel_X^2\right)$$
$$\leq \frac{c_a^2}{\underline{\alpha}^2}|\mu_1 - \mu_2|^2 a(u_N^k(\mu_1), u_N^k(\mu_1); \mu_1) + \frac{1}{4}a(v, v; \mu_2), \tag{C4}$$

where the second inequality follows from coercivity of $a(\cdot, \cdot; \mu)$. With Equations (C2)–(C4), we obtain for $v = \Delta u_N^k$,

$$m(\Delta u_N^k, \Delta u_N^k) + \frac{\Delta t}{2}a(\Delta u_N^k, \Delta u_N^k; \mu_2) \leq m(\Delta u_N^{k-1}, \Delta u_N^k)$$
$$+ \frac{\Delta t}{\underline{\alpha}^2}|\mu_1 - \mu_2|^2 \left(\underline{\alpha}c_f^2 + c_a^2 a(u_N^k(\mu_1), u_N^k(\mu_1); \mu_1)\right). \tag{C5}$$

For the first term on the right, we note by the Cauchy–Schwarz inequality and Young's inequality for $A = m(\Delta u_N^{k-1}, \Delta u_N^{k-1})^{1/2}$, $B = m(\Delta u_N^k, \Delta u_N^k)^{1/2}$ and $\kappa = 1$ that

$$m(\Delta u_N^{k-1}, \Delta u_N^k) \leq m(\Delta u_N^{k-1}, \Delta u_N^{k-1})^{1/2} m(\Delta u_N^k, \Delta u_N^k)^{1/2}$$
$$\leq \frac{1}{2}m(\Delta u_N^{k-1}, \Delta u_N^{k-1}) + \frac{1}{2}m(\Delta u_N^k, \Delta u_N^k). \tag{C6}$$

Hence

$$m(\Delta u_N^k, \Delta u_N^k) - m(\Delta u_N^{k-1}, \Delta u_N^{k-1}) + \Delta t\, a(\Delta u_N^k, \Delta u_N^k; \mu_2)$$
$$\leq \frac{2\Delta t}{\underline{\alpha}^2}|\mu_1 - \mu_2|^2 \left(\underline{\alpha}c_f^2 + c_a^2 a(u_N^k(\mu_1), u_N^k(\mu_1); \mu_1)\right). \tag{C7}$$

We now substitute k' for k and sum over k' to obtain

$$|||\Delta u_N^k|||_{\mu_2}^2 \leq \frac{2}{\underline{\alpha}^2}|\mu_1 - \mu_2|^2 \left(\underline{\alpha}c_f^2 t^k + c_a^2 \Delta t \sum_{k'=1}^{k} a(u_N^{k'}(\mu_1), u_N^{k'}(\mu_1); \mu_1)\right). \tag{C8}$$

We finally note that $\Delta t \sum_{k'=1}^{k} a(u_N^{k'}(\mu_1), u_N^{k'}(\mu_1); \mu_1) \leq |||u_N^k(\mu_1)|||_{\mu_1}^2$. Hence, by Lemma 4.2 we obtain Equation (28) for $\tilde{C}$ given in Equation (29).

Appendix D. Proof of Lemma 4.4

From linearity of Equation (5) we obtain, for $1 \leq k \leq K$,

$$\frac{1}{\Delta t}m(\Delta u_N^k - \Delta u_N^{k-1}, v) + a(\Delta u_N^k, v; \mu_2)$$
$$= f(v; \mu_1) - f(v; \mu_2) + a(u_N^k(\mu_1), v; \mu_2) - a(u_N^k(\mu_1), v; \mu_1), \quad \forall v \in X_N. \tag{D1}$$

We choose $v = (\Delta u_N^k - \Delta u_N^{k-1})/\Delta t \in X_N$ and obtain

$$\begin{aligned}
&\frac{1}{\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L_2}^2 + \frac{1}{\Delta t} a(\Delta u_N^k, \Delta u_N^k - \Delta u_N^{k-1}; \mu_2) \\
&\qquad = \frac{1}{\Delta t} f(\Delta u_N^k - \Delta u_N^{k-1}; \mu_1) - \frac{1}{\Delta t} f(\Delta u_N^k - \Delta u_N^{k-1}; \mu_2) \\
&\qquad + \frac{1}{\Delta t} a(u_N^k(\mu_1), \Delta u_N^k - \Delta u_N^{k-1}; \mu_2) \\
&\qquad - \frac{1}{\Delta t} a(u_N^k(\mu_1), \Delta u_N^k - \Delta u_N^{k-1}; \mu_1), \quad \forall v \in X_N.
\end{aligned} \tag{D2}$$

From Lemma 4.1 we obtain

$$\begin{aligned}
&\frac{1}{\Delta t} f(\Delta u_N^k - \Delta u_N^{k-1}; \mu_1) - \frac{1}{\Delta t} f(\Delta u_N^k - \Delta u_N^{k-1}; \mu_2) \\
&\qquad \leq \frac{c_f}{\Delta t} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} |\mu_1 - \mu_2|
\end{aligned} \tag{D3}$$

and

$$\begin{aligned}
&\frac{1}{\Delta t} a(u_N^k(\mu_1), \Delta u_N^k - \Delta u_N^{k-1}; \mu_2) - \frac{1}{\Delta t} a(u_N^k(\mu_1), \Delta u_N^k - \Delta u_N^{k-1}; \mu_1) \\
&\qquad \leq \frac{c_a}{\Delta t} \parallel u_N^k(\mu_1) \parallel_X \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} |\mu_1 - \mu_2|.
\end{aligned} \tag{D4}$$

We thus obtain

$$\begin{aligned}
&\frac{1}{\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L_2}^2 + \frac{1}{\Delta t} a(\Delta u_N^k, \Delta u_N^k - \Delta u_N^{k-1}; \mu_2) \\
&\qquad \leq \frac{c_f}{\Delta t} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} |\mu_1 - \mu_2| + \frac{c_a}{\Delta t} \parallel u_N^k(\mu_1) \parallel_X \parallel \Delta u_N^k \\
&\qquad - \Delta u_N^{k-1} \parallel_{L^2} |\mu_1 - \mu_2|.
\end{aligned} \tag{D5}$$

We now recall from Equation (16) that $a(\cdot, \cdot; \mu) = a^1(\cdot, \cdot) + a_{\mathrm{II}}(\cdot, \cdot; \mu)$. We may thus write

$$\begin{aligned}
&\frac{1}{\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L_2}^2 + \frac{1}{\Delta t} a^1(\Delta u_N^k, \Delta u_N^k) \\
&\qquad \leq \frac{1}{\Delta t} a^1(\Delta u_N^k, \Delta u_N^{k-1}) + \frac{1}{\Delta t} |a_{\mathrm{II}}(\Delta u_N^k, \Delta u_N^k - \Delta u_N^{k-1}; \mu_2)| \\
&\qquad + \frac{c_f}{\Delta t} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} |\mu_1 - \mu_2| \\
&\qquad + \frac{c_a}{\Delta t} \parallel u_N^k(\mu_1) \parallel_X \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2} |\mu_1 - \mu_2|.
\end{aligned} \tag{D6}$$

Next, we apply the Cauchy–Schwarz inequality to the first term on the right and continuity to the second term on the right; we then apply Young's inequality to each term on the right to obtain

$$
\begin{aligned}
\frac{1}{\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L_2}^2 + \frac{1}{\Delta t} a^1(\Delta u_N^k, \Delta u_N^k) \\
&\leq \frac{1}{2\Delta t}\left(a^1(\Delta u_N^k, \Delta u_N^k) + a^1(\Delta u_N^{k-1}, \Delta u_N^{k-1})\right) \\
&+ \frac{\overline{\gamma}}{2}\left(\frac{1}{3\overline{\gamma}\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2}^2 + 3\overline{\gamma} \parallel \Delta u_N^k \parallel_X^2\right) \\
&+ \frac{1}{2}\left(\frac{1}{3\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2}^2 + 3c_f^2|\mu_1 - \mu_2|^2\right) \\
&+ \frac{1}{2}\left(\frac{1}{3\Delta t^2} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L^2}^2 + 3c_a^2 \parallel u_N^k(\mu_1) \parallel_X^2 |\mu_1 - \mu_2|^2\right),
\end{aligned} \tag{D7}
$$

or

$$
\begin{aligned}
\frac{1}{\Delta t} \parallel \Delta u_N^k - \Delta u_N^{k-1} \parallel_{L_2}^2 + a^1(\Delta u_N^k, \Delta u_N^k) - a^1(\Delta u_N^{k-1}, \Delta u_N^{k-1}) \\
&\leq 3\overline{\gamma}^2 \Delta t \parallel \Delta u_N^k \parallel_X^2 + 3|\mu_1 - \mu_2|^2 (c_f^2 \Delta t + c_a^2 \Delta t \parallel u_N^k(\mu_1) \parallel_X^2).
\end{aligned} \tag{D8}
$$

We then substitute k' for k and sum over k' to obtain

$$
\begin{aligned}
\frac{1}{\Delta t}\sum_{k'=1}^{k} \parallel \Delta u_N^{k'} - \Delta u_N^{k'-1} \parallel_{L_2}^2 + a^1(\Delta u_N^k, \Delta u_N^k) \\
&\leq 3\overline{\gamma}^2 \sum_{k'=1}^{k} \Delta t \parallel \Delta u_N^{k'} \parallel_X^2 + 3|\mu_1 - \mu_2|^2 (c_f^2 t^k + c_a^2 \Delta t \sum_{k'=1}^{k} \parallel u_N^{k'}(\mu_1) \parallel_X^2).
\end{aligned} \tag{D9}
$$

Finally, we first invoke coercivity of $a(\cdot, \cdot; \mu_2)$, and then Lemmas 4.2 and 4.3 to obtain

$$
\begin{aligned}
&\frac{1}{\Delta t}\sum_{k'=1}^{k} \parallel \Delta u_N^{k'} - \Delta u_N^{k'-1} \parallel_{L_2}^2 + a^1(\Delta u_N^k, \Delta u_N^k) \\
&\leq \frac{3\overline{\gamma}^2}{\underline{\alpha}} \Delta t \sum_{k'=1}^{k} a(\Delta u_N^{k'}, \Delta u_N^{k'}; \mu_2) + \frac{3}{\underline{\alpha}}|\mu_1 - \mu_2|^2 \left(\underline{\alpha} c_f^2 t^k + c_a^2 \Delta t \sum_{k'=1}^{k} a(u_N^{k'}(\mu_1), u_N^{k'}(\mu_1); \mu_1)\right) \\
&\qquad \leq |\mu_1 - \mu_2|^2 \frac{3}{\underline{\alpha}^2}(\overline{\gamma}^2 \underline{\alpha} \tilde{C}^2 + t^k \underline{\alpha}^2 c_f^2 + t^k c_a^2 \max_{\mu \in \mathcal{D}} \parallel f(\cdot; \mu) \parallel_{X'}).
\end{aligned} \tag{D10}
$$

The desired result thus follows because $a^1(\Delta u_N^k, \Delta u_N^k) \geq 0$.

Acknowledgements

This work was supported by the Norwegian University of Science and Technology, AFOSR Grant No. FA9550-07-1-0425, and OSD/AFOSR Grant No. FA9550-09-1-0613.

Notes

1. In the linear case $b = 0$, and it thus follows from Equation (9) and the definition of τ_N^{LB} (we recall that $a(\cdot, \cdot; \mu)$ is coercive) that Equation (10) simplifies to $\Delta_N^k(\mu) = (\frac{\Delta t}{\alpha_{\mathrm{LB}}\sigma_{\mathrm{LB}}(\mu)} \sum_{k'=1}^{k} \epsilon_N(t^{k'}; \mu)^2)^{1/2}$.
2. We refer to [12,15] for details on the Construction–Evaluation procedure for the computation of lower bounds for the stability constants – a Successive Constraint Method (SCM).
3. We note that $(\chi^i, \chi^j)_X = \sum_{k=1}^{K} \sum_{l=1}^{K} \psi_k^i \psi_l^j (w^k, w^l)_X = K(\psi^i)^{\mathrm{T}} C \psi^j = \delta_{ij}$.
4. We note that after completion of the *hp*-POD/Greedy, we can apply the SCM algorithm independently for each parameter subdomain; we thus expect a reduction in the SCM (Online) evaluation cost because the size of the parameter domain is effectively reduced.
5. We suppose here for simplicity that $\|\cdot\|_X = \|\cdot\|_{H^1}$; hence $\| v \|_{L^2} \leq \| v \|_X$ for all $v \in X$.
6. In fact, we should interpret M here as the number of subdomains generated by Algorithm 4.3 *so far*; the $\mathcal{V}_{B^m}$, $1 \leq m \leq M$, are not necessarily the *final* M subdomains. With this interpretation, we thus do not presume termination of the algorithm.
7. Equation (16) is satisfied with $a_{\mathrm{II}}(w, v; \mu) = \int_\Omega (V(\mu) \cdot \nabla w) v$. We note that a_{II} is $L^2(\Omega)$ continuous in its second argument because by the Cauchy–Schwarz inequality $a_{\mathrm{II}}(w, v) \leq (\int_\Omega (V(\mu) \cdot \nabla w)^2)^{1/2} (\int_\Omega v^2)^{1/2}$.
8. To ensure a good spread over $\mathcal{D}$ of the rather few (25 or 64 for our two examples) *initial* train points, we use for $\Xi_{\mathrm{train},(1)}$ a deterministic initial regular grid. (For the train sample *enrichment*, we use random points.) As some train points belong to a regular grid, the procedure may produce 'aligned' subdomain boundaries, as seen in Figure 3.
9. We note that $\sup_{v \in X^{\mathcal{N}}} \frac{\ell_n(v;\mu)}{\|v\|_{L^2}} = \frac{1}{\mu_1 |D_n|} \sup_{v \in X^{\mathcal{N}}} \frac{\int_{D_n} v}{\|v\|_{L^2}} \leq \frac{1}{\mu_1 \sqrt{|D_n|}} \sqrt{\frac{\int_{D_n} v^2}{\int_\Omega v^2}} \leq \frac{1}{\mu_1 \sqrt{|D_n|}}$.

References

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

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

[3] B. Haasdonk and M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, Math. Model. Numer. Anal. 42 (2008), pp. 277–302.

[4] M.A. Grepl and A.T. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations*, M2AN Math. Model. Numer. Anal. 39 (2005), pp. 157–181.

[5] 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*, Arch. Comput. Methods Eng. 15 (2008), pp. 229–275.

[6] 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 *Large-Scale Inverse Problems and Quantification of Uncertainty*, Wiley Series in Computational Statistics, L. Biegler, G. Biro, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. van Bloemen Waanders, and K. Willcox, eds., John Wiley & Sons, Chichester, 2010.

[7] J.L. Eftang, A.T. Patera, and E.M. Ronquist, *An 'hp' certified reduced basis method for parametrized elliptic partial differential equations,* SIAM J. Sci. Comput. 32 (2010), pp. 3170–3200.

[8] 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. Rep. 28, SRC SimTech, 2010.

[9] 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*, Orlando, FL, January, 2009.

[10] J.L. Eftang, A.T. Patera, and E.M. Rønquist, *An hp certified reduced basis method for parametrized parabolic partial differential equations*, in *Spectral and High Order Methods*

for Partial Differential Equations, Lecture Notes in Computational Science and Engineering, Vol. 76, J.S. Hesthaven and E.M. Rønquist, eds., Springer, Trondheim, Norway (2009), pp. 179–187.

[11] D.J. Knezevic, N.C. Nguyen, and A.T. Patera, *Reduced basis approximation and a posteriori error estimation for the parametrized unsteady Boussinesq equations*, M3AS, Accepted 2010.

[12] 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.

[13] 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.

[14] J.L. Eftang, M.A. Grepl, and A.T. Patera, *A posteriori error bounds for the empirical interpolation method*, C. R. Math. 348 (2010), pp. 575–579.

[15] 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.

[16] L. Sirovich, *Turbulence and the dynamics of coherent structures. Part I. Coherent structures*, Quart. Appl. Math. 45 (1987), pp. 561–571.

[17] D.J. Knezevic and A.T. Patera, *A certified reduced basis method for the fokker-planck equation of dilute polymeric fluids: fene dumbbells in extensional flow*, SIAM J. Sci. Comput. (2010), pp. 793–817.

[18] A. Quarteroni and A. Valli, *Numerical approximation of partial differential equations*, Springer Series in Computational Mathematics Vol. 23, Springer-Verlag, Berlin, 1994.

[19] D.J. Knezevic and J.W. Peterson, *A high-performance parallel implementation of the certified reduced basis method*, Comput. Methods Appl. Mech. Eng., Submitted 2010.

[20] B.S. Kirk, J.W. Peterson, R.M. Stogner, and G.F. Carey, *libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations*, Eng. Comput. 23 (2006), pp. 237–254.