Stabilized Weighted Reduced Basis Methods for Parametrized Advection Dominated Problems with Random Inputs

SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 6, No. 4, pp. 1475–1502

Stabilized Weighted Reduced Basis Methods for Parametrized Advection Dominated Problems with Random Inputs*

Davide Torlo[†‡], Francesco Ballarin[†], and Gianluigi Rozza[†]

Abstract. In this work, we propose viable and efficient strategies for stabilized parametrized advection dominated problems, with random inputs. In particular, we investigate the combination of the wRB (weighted reduced basis) method for stochastic parametrized problems with the stabilized RB (reduced basis) method, which is the integration of classical stabilization methods (streamline/upwind Petrov–Galerkin (SUPG) in our case) in the offline–online structure of the RB method. Moreover, we introduce a reduction method that selectively enables online stabilization; this leads to a sensible reduction of computational costs, while keeping a very good accuracy with respect to high-fidelity solutions. We present numerical test cases to assess the performance of the proposed methods in steady and unsteady problems related to heat transfer phenomena.

Key words. random inputs, reduced basis methods, uncertainty quantification, stochastic parametrized advection–diffusion equations, advection dominated problems

AMS subject classifications. 35J15, 65C30, 65N35, 60H15, 60H35

DOI. 10.1137/17M1163517

1. Introduction.

Advection–diffusion equations are very important in many engineering applications because they are used to model, for example, heat transfer phenomena [25] or diffusion phenomena, such as of pollutants in the atmosphere [13]. We are interested in studying related advection–diffusion partial differential equations (PDEs) when their Péclet numbers, representing, roughly, the ratio between the advection and the diffusion field, are high. Moreover, in such applications, we often need very fast evaluations of the approximated solution, depending on some input parameters, which may be deterministic or uncertain. This happens, for example, in the case of *real-time* simulation or if we need to perform repeated approximations of solutions for different input parameters. We find such *many-query* situations in optimization problems, in which the objective function to be optimized depends on the parameters through the solution of a PDE or a system of PDEs.

The aim of this work is to study a stabilized reduced basis (RB) method suitable for

*Received by the editors January 2, 2018; accepted for publication (in revised form) September 5, 2018; published electronically October 25, 2018.

http://www.siam.org/journals/juq/6-4/M116351.html

Funding: This work was funded by European Union Funding for Research and Innovation (project H2020 ERC CoG 2015 AROMA-CFD project 681447, "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics") and by the INDAM-GNCS projects "Metodi numerici avanzati combinati con tecniche di riduzione computazionale per PDEs parametrizzate e applicazioni" and "Numerical methods for model order reduction of PDEs."

[†]mathLab, Mathematics Area, SISSA, Via Bonomea 265, I-34136 Trieste, Italy (davidetorlo@gmail.com, francesco.ballarin@sissa.it, gianluigi.rozza@sissa.it).

[‡]Current address: Institut für Mathematik, UZH, Universität Zürich, Winterthurerstrasse 190, CH-8057, Zürich, Switzerland.

the approximation of parametrized advection–diffusion PDEs in advection dominated cases, including a stochastic context, by considering random inputs. Indeed, the RB method [20] has been devised to reduce the computational effort required by the repeated solution of parametrized problems. It provides rapid approximation of solution of PDEs and is able to guarantee the *reliability* of the solution with a sharp and accurate a posteriori error bound. In the literature we can find many works on the application of the RB method to advection–diffusion problems, in particular, problems with low Péclet numbers [16, 40, 44].

In contrast, problems characterized by high Péclet numbers are far more complex and may exhibit instabilities even with classical high-fidelity numerical approximations, such as the finite element or the finite difference method. To deal with this issue we have to resort to some stabilization techniques [7, 42], such as streamlined/upwind Petrov–Galerkin (SUPG) stabilization. A similar stabilization needs to be accounted for also at the reduced order level, resulting in a stabilized version of the RB algorithm [37, 38, 39]. In particular, in [37, 38, 39] it was shown that a double stabilization in the *offline* stage and the *online* stage was necessary to obtain an accurate approximation. Nevertheless, stabilizations in the online phase can be a bothersome computational cost that may damage the efficiency of the method (for example, in the *many-query* context), while in some other situation an *offline-only* stabilized method is preferred. Stabilization of problems characterized by strong convection effects is an active topic of research in the model order reduction community; see, e.g., [1, 2, 3, 8, 17, 23, 24, 31, 32, 37, 38, 47] for several different proposed methods with applications in heat transfer and computational fluid dynamics.

When dealing with stochastic equations, i.e., with random input parameters, we can modify the RB method according to probability laws that rule our parameters. In this direction, the wRB (weighted reduced basis) method [10] wants to exploit all the information that random variables give us (a review is provided in [12]). The main novelties of these papers are (i) the synergy of wRB with a stabilized formulation, suitable for stochastic advection dominated problems; and (ii) the resulting capability to enable adaptive toggling of the stabilization depending on the stochastic Péclet number. In particular, we will apply the weighted method to stabilized RB strategies and prove the accuracy of the combined method. Throughout this work we will test these methods on some steady and time-dependent problems.

The outline of the paper is as follows. In section 2 we will briefly introduce elliptic coercive parametrized PDEs, their associate RB method, and some classical stabilization methods for FE (finite element) approximation of advection dominated problems; then we will study two RB stabilization methods by testing them on some examples. We will consider next stochastic PDEs; in section 3 we present the wRB method and combine it with proper stabilization techniques. Moreover, we will provide a method that selectively enables stabilization to optimize computational costs. In section 4 we will extend these ideas to parabolic problems by introducing the general wRB method for these problems, combining it with a suitable stabilization technique (based on stabilization for the FE approximation of advection dominated parabolic problems) and testing it on a few examples. Finally, section 5 will provide some conclusions and future perspectives.

2. Stabilized reduced basis method for deterministic elliptic equations.

2.1. A brief introduction to the reduced basis method. The RB method is a reduced order modeling (ROM) technique which provides rapid and reliable solutions for parametrized PDEs (PPDEs) [20], in which the parameters can be either physical or geometrical, or either deterministic or stochastic.

The need to solve this kind of problem arises in many engineering applications, in which the evaluation of some *output* quantities is required. These *outputs* are often functionals of the solution of a PDE, which can in turn depend on some *input* parameters. The aim of the RB method is to provide a very fast computation of this *input-output* evaluation, and so it turns out to be very useful—especially in the real-time or many-query context.

Roughly speaking, given a value of the parameter, the (Lagrange) RB method consists of a Galerkin projection of the continuous solution on a particular subspace of a high-fidelity approximation space, e.g., an FE space with a large number of degrees of freedom. This subspace is spanned by some precomputed high-fidelity global solutions (snapshots) of the continuous parametrized problem, corresponding to some properly chosen values of the parameter.

For a complete presentation of the RB method we refer the reader to [20]; here we just recall its main features in order to introduce some notation.

2.1.1. The continuous problem. Let $\boldsymbol{\mu}$ belong to the parameter domain $\mathcal{D} \subset \mathbb{R}^p$, $p \in \mathbb{N}$. Let Ω be a regular bounded open subset of $\mathbb{R}^d$, $d = 1, 2, 3$, and let X be a suitable Hilbert space. For any $\boldsymbol{\mu} \in \mathcal{D}$, let $a(\cdot, \cdot; \boldsymbol{\mu}) : X \times X \to \mathbb{R}$ be a bilinear form, and let $F(\cdot; \boldsymbol{\mu}) : X \to \mathbb{R}$ be a linear functional. As we will focus on advection–diffusion equations that are second order elliptic PDEs, the space X will be such that $H_0^1(\Omega) \subset X \subset H^1(\Omega)$. Formally, our problem can be written as follows:

$$\begin{aligned} &\text{For any } \boldsymbol{\mu} \in \mathcal{D}, \text{ find } u(\boldsymbol{\mu}) \in X : \\ &a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = F(v; \boldsymbol{\mu}) \quad \forall v \in X. \end{aligned} \tag{1}$$

We require a to be coercive and continuous, i.e., respectively,

$$\exists\ \alpha_0 \text{ such that } \alpha_0 \le \alpha(\boldsymbol{\mu}) = \inf_{v \in X} \frac{a(v, v; \boldsymbol{\mu})}{||v||_X^2} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{2}$$

and

$$+\infty > \gamma(\boldsymbol{\mu}) = \sup_{v \in X} \sup_{w \in X} \frac{|a(v, w; \boldsymbol{\mu})|}{||v||_X ||w||_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{3}$$

For the sake of online efficiency, we assume an *affine* dependence of a on the parameter $\boldsymbol{\mu}$; i.e., we assume that

$$a(v, w; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(v, w) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{4}$$

Here, $\Theta_a^q(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_a$, are smooth functions, while $a^q : X \times X \to \mathbb{R}$, $q = 1, \dots, Q_a$, are $\boldsymbol{\mu}$-independent continuous bilinear forms.

In a similar way, we assume that also the functional F is continuous and depends "affinely" on parameters:

$$F(v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) F^q(v) \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{5}$$

where, also in this case, $\Theta_F^q(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_F$, are smooth functions, while $F^q : X \to \mathbb{R}$, $q = 1, \dots, Q_F$, are $\boldsymbol{\mu}$-independent continuous linear functionals.

Let $X^{\mathcal{N}} \subset X$ be a conforming FE space with $\mathcal{N}$ degrees of freedom. We can now set the *truth* approximation of the problem (1):

$$\begin{aligned} &\text{For any } \boldsymbol{\mu} \in \mathcal{D}, \text{ find } u^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\ &a(u^{\mathcal{N}}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = F(v^{\mathcal{N}}; \boldsymbol{\mu}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}. \end{aligned} \tag{6}$$

As we are considering the conforming FE case, conditions similar to (2) and (3) are fulfilled by restriction. More precisely, as regards the coercivity of the restriction of a to $X^{\mathcal{N}} \times X^{\mathcal{N}}$, we define

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) := \inf_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{a(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu})}{||v^{\mathcal{N}}||_X^2} \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{7}$$

and, as we are considering a restriction, it easily follows that $\alpha(\boldsymbol{\mu}) \leq \alpha^{\mathcal{N}}(\boldsymbol{\mu})\ \forall \boldsymbol{\mu} \in \mathcal{D}$. Similarly, for the continuity, we can define

$$+\infty > \gamma^{\mathcal{N}}(\boldsymbol{\mu}) = \sup_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \sup_{w^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{|a(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu})|}{||v^{\mathcal{N}}||_X ||w^{\mathcal{N}}||_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{8}$$

As we have already mentioned, the domain of the equation also can depend on the parameter. In this case we need to map the parametric domain $\Omega_p(\boldsymbol{\mu})$ onto a reference domain, denoted with Ω, via a suitable parameter-dependent transformation $T(\cdot;\boldsymbol{\mu}) : \Omega \to \Omega_p(\boldsymbol{\mu})$; see [4, 20, 29, 33]. This allows us to track back on the reference domain Ω all the involved bilinear and linear forms, so that (4) and (5) are defined on a common reference domain Ω. In this work we use only affine mappings [20, 33] that allow us to easily recover the affinity assumptions (4) and (5). In [33, 43] it is possible to find, in particular, a detailed treatment of the advection–diffusion operators.

2.1.2. The reduced basis method: Main features. Let us suppose that we are given a problem in the form (1) and its truth approximation (6). We recall that the dimension of the FE space $X^{\mathcal{N}}$ is $\mathcal{N}$. Given an integer $N \ll \mathcal{N}$, suppose that we are given a set of N suitable parameter values, $S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$: this allows us to define the *RB space* as $X_N^{\mathcal{N}} = \text{span}\{u^{\mathcal{N}}(\boldsymbol{\mu}^n) : 1 \leq n \leq N\}$. To be more precise, a Gram–Schmidt orthonormalization process on $\{u^{\mathcal{N}}(\boldsymbol{\mu}^n) : 1 \leq n \leq N\}$ is usually carried out for the sake of numerical stability, and the resulting orthonormal functions are considered as bases of the reduced space [20, 43].

Given a value $\boldsymbol{\mu} \in \mathcal{D}$, we define the RB solution $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ such that

$$a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = F(v_N; \boldsymbol{\mu}) \quad \forall v_N \in X_N^{\mathcal{N}}. \tag{9}$$

Recalling that $N \ll \mathcal{N}$, we emphasize the fact that to find the RB solution we need only solve an $N \times N$ linear system, instead of the $\mathcal{N} \times \mathcal{N}$ system of the FE method. Moreover, we can also guarantee that the error for a parameter $\boldsymbol{\mu} \in \mathcal{D}$ is bounded by an error estimator $\Delta_N(\boldsymbol{\mu})$:

$$|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \Delta_N(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{10}$$

where $|||\cdot|||_{\boldsymbol{\mu}}$ is the norm induced by the symmetric part $a_S(\cdot,\cdot;\boldsymbol{\mu})$ of the bilinear form $a(\cdot,\cdot;\boldsymbol{\mu})$. The error estimator is defined as $\Delta_N(\boldsymbol{\mu}) := \frac{||\hat{r}(\boldsymbol{\mu})||_X}{\sqrt{\alpha_{LB}(\boldsymbol{\mu})}}$, where $\hat{r}$ is the Riesz representer for the functional $r(v^{\mathcal{N}}, \boldsymbol{\mu}) = F(v^{\mathcal{N}}; \boldsymbol{\mu}) - a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu})$, $||\cdot||_X$ is the norm associated to the scalar product in X, and $\alpha_{LB}(\boldsymbol{\mu})$ is a lower bound for the coercivity constant $\alpha(\boldsymbol{\mu})$, possibly dependent on $\boldsymbol{\mu} \in \mathcal{D}$.

The set S_N is built in the offline stage using a greedy algorithm on a training set Ξ_{train} that spans $\mathcal{D}$ [20, 43]. It is an iterative method that, at each step, chooses the parameter value which maximizes the a posteriori error estimator $\boldsymbol{\mu} \mapsto \Delta_N(\boldsymbol{\mu})$ in the training set. The algorithm stops when a prescribed tolerance ε_{tol}^* is reached, that is, when $\Delta_N(\boldsymbol{\mu}) \leq \varepsilon_{tol}^*$ for each parameter value $\boldsymbol{\mu}$ in the training set $\Xi_{train} \subset \mathcal{D}$. We assume in this section that Ξ_{train} is a collection of randomly selected parameter values according to a uniform distribution. The error estimator Δ_N is sharp in order to avoid an unnecessarily high dimension N for the RB space. Moreover, it must be computationally inexpensive in order to speed up the greedy algorithm (within which it is computed many times) and to allow the certification of the RB solution during the online stage.

We want to point out that all the expensive computations (i.e., those with costs depending on the FE space dimension $\mathcal{N}$) are performed during the offline stage. Indeed, the affinity assumptions (4) and (5) are crucial for the offline–online decoupling, as extensively shown in [20, 43]. The affinity assumptions allow the storage, during the offline stage, of the matrices corresponding to the parameter-independent forms $a_q, q = 1, \ldots, Q_a$, restricted to $X_N^{\mathcal{N}}$. Thanks to this fact, during the online stage the assembly of the RB system only consists of a linear combination of these precomputed matrices. A similar strategy can also be applied to the computation of the error estimator [20, 43]. Indeed, thanks to the affine decomposition of F (5) and a (4), $\hat{r}$ can be computed in an online phase, with a complexity that depends only on N but not on $\mathcal{N}$ [20]. Also, the $\alpha_{LB}(\boldsymbol{\mu})$ can be efficiently computed in an online phase thanks to suitable algorithms, such as the successive constraint method [20, 22]. Therefore, at each step of the greedy algorithm, the error estimator $\Delta_N(\boldsymbol{\mu})$ can be efficiently evaluated (with computational complexity independent from $\mathcal{N}$) for any element in the training set, rather than relying on the computation of the error $|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$ (which would require an expensive truth solve for all parameters in the training set, such as in a proper orthogonal decomposition basis generation). If affinity assumptions are not fulfilled, it turns out to be necessary to use an interpolation strategy (e.g., empirical interpolation method (EIM) [6, 15]) in order to recover them. A weighted version of the EIM is provided in [11].

2.2. Stabilized reduced basis methods. The main goal of this section is to design an efficient stabilization procedure for the RB method. More specifically, we will make a comparison between an offline–online stabilized method and an offline-only stabilized method as done in

[37]. We want to approximate the solution of a parametric advection–diffusion problem,

$$-\varepsilon\Delta u + \boldsymbol{\beta}\cdot\nabla u = f \quad \text{in } \Omega \subset \mathbb{R}^d, \tag{11}$$

given a parameter value $\boldsymbol{\mu} \in \mathcal{D}$ and suitable Dirichlet, Neumann, or mixed boundary conditions. Here $\varepsilon = \varepsilon(\boldsymbol{\mu}) : \Omega \to [0, +\infty)$ is a parametrized diffusion coefficient, while $\boldsymbol{\beta} = \boldsymbol{\beta}(\boldsymbol{\mu}) : \Omega \to \mathbb{R}^d$ is a parametrized advection field such that $\operatorname{div}(\boldsymbol{\beta}) = 0$.

Let $\mathcal{T}_h$ be a triangulation of Ω, and let K be an element of $\mathcal{T}_h$. We say that a problem is advection dominated in K if the following condition holds:

$$\mathbb{P}\mathsf{e}_K(x) := \frac{|\boldsymbol{\beta}(x)|h_k}{2\varepsilon(x)} > 1 \quad \forall x \in K, \tag{12}$$

where h_K is the diameter of K. It is very well known from the literature (e.g., [42]) that the FE approximation of advection dominated problems can show significant instability phenomena, e.g., spurious oscillations near the boundary layers. Several recipes have been proposed to fix these issues. We choose to resort to the strongly consistent stabilization method of SUPG [7, 21, 27, 28]. The main idea of stabilization techniques is to add artificial diffusion to equation (11). To increase the accuracy of the resulting solution, SUPG adds diffusion only in the streamline direction—not everywhere as in a purely artificial diffusion scheme. Moreover, the resulting method is strongly consistent with the continuous PDE and, provided that the stabilization coefficients are properly chosen, retains the same order of accuracy as the underlying discretization scheme. For a detailed presentation of the stabilization method for the FE approximation of advection dominated problems, we refer the reader to [21, 42].

Let us now explain the basic ideas of the two RB stabilization methods mentioned before. As regards the offline–online stabilized method, the choice of the name reveals that the Galerkin projections are performed, in both the offline and the online stage, with respect to the SUPG stabilized bilinear form [7, 42], that is,

$$\begin{aligned} a_{stab}(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) &= a(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) + s(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}), \\ F_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= F(v^{\mathcal{N}}; \boldsymbol{\mu}) + r(v^{\mathcal{N}}; \boldsymbol{\mu}), \end{aligned} \tag{13}$$

$$\begin{aligned} a(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) &= \int_\Omega \varepsilon(\boldsymbol{\mu})\nabla w^{\mathcal{N}} \cdot \nabla v^{\mathcal{N}} + (\boldsymbol{\beta}(\boldsymbol{\mu})\cdot\nabla w^{\mathcal{N}})v^{\mathcal{N}}, \\ F(v^{\mathcal{N}}; \boldsymbol{\mu}) &= \int_\Omega f v^{\mathcal{N}}, \end{aligned} \tag{14}$$

$$s(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) = \sum_{K\in\mathcal{T}_h} \delta_K \int_K L w^{\mathcal{N}} \frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L_{SS} v^{\mathcal{N}}, \tag{15}$$

$$r(v^{\mathcal{N}}; \boldsymbol{\mu}) = \sum_{K\in\mathcal{T}_h} \delta_K \int_K f \frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L_{SS} v^{\mathcal{N}}, \tag{16}$$

where $w^{\mathcal{N}}, v^{\mathcal{N}}$ are chosen in a suitable piecewise polynomial space $X^{\mathcal{N}}$. In (15) L is the parameter-dependent advection–diffusion operator, that is, $Lv^{\mathcal{N}} = \varepsilon\Delta v^{\mathcal{N}} + \boldsymbol{\beta}\cdot\nabla v^{\mathcal{N}}$, which can be split into its symmetric part $L_S u^{\mathcal{N}} = -\varepsilon\Delta u^{\mathcal{N}}$ and its skew-symmetric part $L_{SS} u^{\mathcal{N}} = \boldsymbol{\beta}\cdot\nabla u^{\mathcal{N}}$. Moreover, h_K denotes the diameter of the element K, while δ_K is a positive real

number which may depend on K through the parameter $\boldsymbol{\mu}$ (but does not directly depend on h_K).

In contrast, in the offline-only stabilized method we use the stabilized form (13) only during the offline stage, while during the online stage we project with respect to the standard advection–diffusion bilinear form (14). An advantage of using the offline-only stabilized method would be a certain reduction of the online computational effort in the assembly of the reduced linear system; this reduction could be also significant if the number of affine stabilization terms is very high. Among possible disadvantages, we mention the inconsistency between the offline and online bilinear forms.

We will start with the study of some test problems, which we will keep as prototypes for each further extension that will be carried out in the next sections. The first is a Poiseuille–Graetz (PG) problem [25, 40, 37], while the second is a parametrized internal layer problem [37]. From here on, we will explicitly write the FE space dimension $\mathcal{N}$ only when it will be strictly necessary.

2.2.1. Numerical test: Poiseuille–Graetz problem. We consider a PG problem where we have two parameters: one physical (the inverse of diffusivity coefficient μ_1, which is proportional to the Péclet number) and one geometrical (the length of the domain being equal to $1+\mu_2$). The PG problem deals with steady forced heat convection (advective phenomenon) combined with heat conduction (diffusive phenomenon) in a duct with walls at different temperatures. Let us define $\boldsymbol{\mu} = (\mu_1, \mu_2)$ with both μ_1 and μ_2 positive, real numbers. Let $\Omega_p(\boldsymbol{\mu})$ be the rectangle $(0, 1+\mu_2) \times (0,1)$ in $\mathbb{R}^2$. The domain is shown in Figure 1. The problem is to find a solution $u(\boldsymbol{\mu})$, representing the temperature distribution, such that

$$
\text{(17)} \qquad \begin{cases} -\frac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + 4y(1-y)\partial_x u(\boldsymbol{\mu}) = 0 & \text{in } \Omega_p(\boldsymbol{\mu}), \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_{p,1}(\boldsymbol{\mu}) \cup \Gamma_{p,2}(\boldsymbol{\mu}) \cup \Gamma_{p,6}(\boldsymbol{\mu}), \\ u(\boldsymbol{\mu}) = 1 & \text{on } \Gamma_{p,3}(\boldsymbol{\mu}) \cup \Gamma_{p,5}(\boldsymbol{\mu}), \\ \frac{\partial u}{\partial \nu} = 0 & \text{on } \Gamma_{p,4}(\boldsymbol{\mu}). \end{cases}
$$

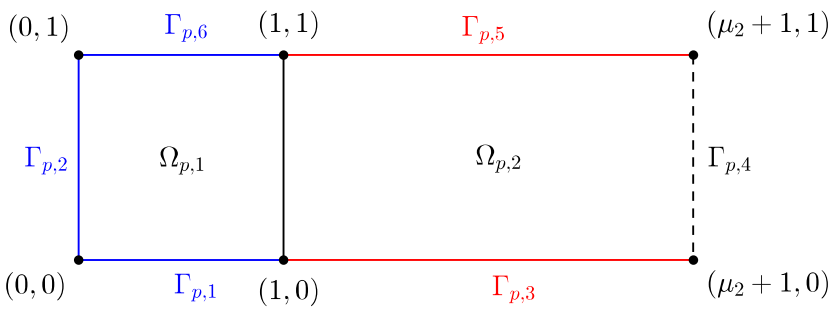


Figure 1. *Geometry of PG problem. Parametrized domain. Boundary conditions: homogeneous Dirichlet on the blue sides, $u = 1$ on the red sides, and homogeneous Neumann on the dashed side.*

We set the reference domain as $\Omega = (0,2) \times (0,1)$ and subdivide it into $\Omega^1 = (0,1) \times (0,1)$ and $\Omega^2 = (1,2) \times (0,1)$. The affine transformation that maps the reference domain into the parametrized one is

$$T^1(\boldsymbol{\mu}) : \Omega^1 \to \Omega_{p,1}(\boldsymbol{\mu}) \subset \mathbb{R}^2, \qquad T^2(\boldsymbol{\mu}) : \Omega^2 \to \Omega_{p,2}(\boldsymbol{\mu}) \subset \mathbb{R}^2, \tag{18}$$

$$T^1\left(\begin{pmatrix} x \\ y \end{pmatrix}; \boldsymbol{\mu}\right) = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad T^2\left(\begin{pmatrix} x \\ y \end{pmatrix}; \boldsymbol{\mu}\right) = \begin{pmatrix} \mu_2 x \\ y \end{pmatrix} + \begin{pmatrix} 1-\mu_2 \\ 0 \end{pmatrix}. \tag{19}$$

We define the continuous one-to-one transformation $T(\boldsymbol{\mu})$ by gluing together these two transformations.

Let us now define a mesh $\mathcal{T}_h$ on the reference domain Ω, and let us call $\mathcal{T}_h^1$ and $\mathcal{T}_h^2$ the restrictions $\mathcal{T}_h$ to Ω_1 and Ω_2, respectively. We use $\mathbb{P}^1$ FE discretization during the offline stage. Hence, the corresponding bilinear forms $a(\cdot,\cdot;\boldsymbol{\mu})$ and $s(\cdot,\cdot;\boldsymbol{\mu})$ are

$$\begin{aligned} a(u^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) := & \int_{\Omega^1} \frac{1}{\mu_1} \nabla u^{\mathcal{N}} \nabla v^{\mathcal{N}} + 4y(1-y)\partial_x u^{\mathcal{N}} v^{\mathcal{N}} \\ & + \int_{\Omega^2} \frac{1}{\mu_1 \mu_2} \partial_x u^{\mathcal{N}} \partial_x v^{\mathcal{N}} + \frac{\mu_2}{\mu_1} \partial_x u^{\mathcal{N}} \partial_y v^{\mathcal{N}} + 4\mu_2 y(1-y) \partial_x u^{\mathcal{N}} v^{\mathcal{N}} \end{aligned} \tag{20}$$

and

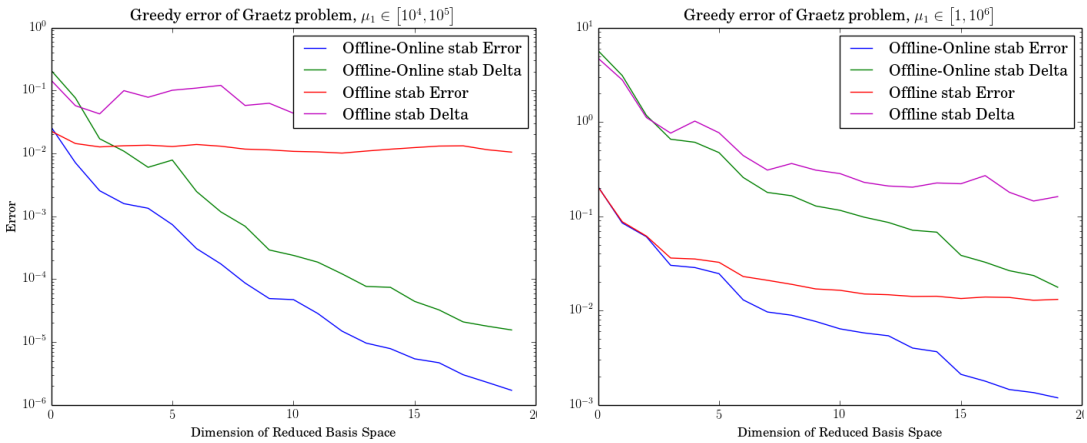
$$s(u^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) := \sum_{K \in \mathcal{T}_h^1} h_K \int_K (4y(1-y)\partial_x u^{\mathcal{N}}) \partial_x v^{\mathcal{N}} + \sum_{K \in \mathcal{T}_h^1} \frac{h_K}{\sqrt{\mu_2}} \int_K (4y(1-y)\partial_x u^{\mathcal{N}}) \partial_x v^{\mathcal{N}}. \tag{21}$$

The choice of the stabilization coefficient $\delta_{K_p} = \delta_{K_p}(\boldsymbol{\mu}) = \frac{1}{\sqrt{\mu_2}}$ for $K_p \in \mathcal{T}_h^2$ is motivated by the transformation to the reference domain.

We test the performance of the RB approximation for two choices of the parameter space, namely $\mathcal{D}^1 = \left[10^4, 10^5\right] \times [0.5, 4]$ and $\mathcal{D}^2 = \left[1, 10^6\right] \times [0.5, 4]$. The parameter space $\mathcal{D}^1$ features very large values of μ_1, so that the solution manifold is characterized by a solution with steep boundary layers. In contrast, the parameter space $\mathcal{D}^2$ features both small and large values of μ_1, resulting in a richer set of solutions. The range of variation for the geometrical parameter μ_2 is the same in both parameter spaces.

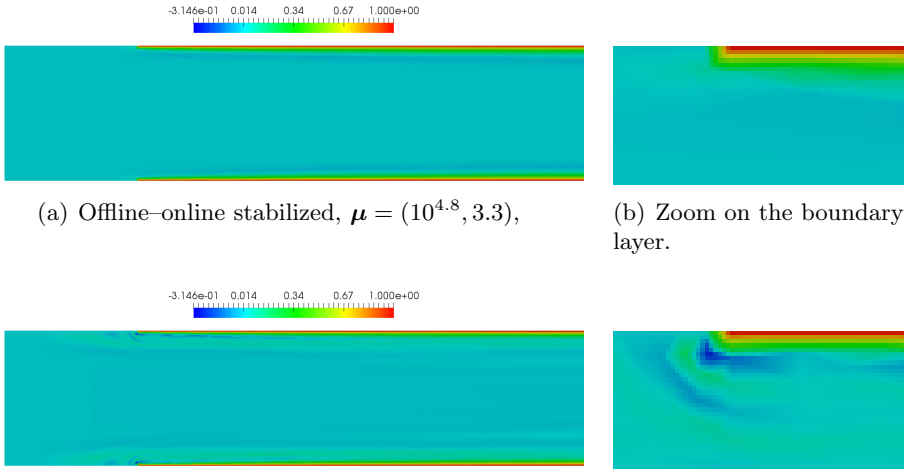
The comparison of offline-only and offline–online stabilized algorithms is shown in Figure 2 for $\mathcal{D}^1$ (a) and $\mathcal{D}^2$ (b). In each subfigure, the evolution of the greedy parameter selection is presented, plotting for comparison both the error bound $\max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_N(\boldsymbol{\mu})$, employed by the RB algorithm, and the energy norm error $\max_{\boldsymbol{\mu} \in \Xi_{train}} |||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$. For both $\mathcal{D}^1$ and $\mathcal{D}^2$, the greedy algorithm in the online–offline case is clearly converging as the RB space enriches its dimension. In contrast, the greedy algorithm does not converge in the offline-only case, being over 10^{-2} for both $\mathcal{D}^1$ and $\mathcal{D}^2$.

We show a representative online solution for both stabilization cases, characterized by large value of Péclet number, in Figure 3, obtained for $N = 20$. As we can see, the offline–online stabilized RB solution is showing marked boundary layers, while the offline-only stabilized RB solution still has some noise near the boundary layer and some peaks near discontinuities of the solution at the top and the bottom of the channel.



(a) $\mu_1 \in (10^4, 10^5)$. (b) $\mu_1 \in (1, 10^6)$.

Figure 2. *Error comparison between offline and online–offline stabilization.*



(a) Offline–online stabilized, $\boldsymbol{\mu} = (10^{4.8}, 3.3)$, (b) Zoom on the boundary layer.

(c) Offline stabilized, $\boldsymbol{\mu} = (10^{4.8}, 3.3)$. (d) Zoom on the boundary layer.

Figure 3. *RB solution, stabilized offline–online and offline,* $\boldsymbol{\mu} = (10^{4.8}, 3.3)$.

Moreover, if we compare the time used to perform one truth solution ($\mathcal{N} = 4369$) and an RB solution ($N = 20$), we can see that on average, on a test set the former lasts 0.0411 seconds, while the stabilized online RB solution lasts 0.000512 seconds. The nonstabilized solution in the online phase lasts even less time, namely 0.000151 seconds, even though it is less accurate (see Figure 3). The further speedup of the nonstabilized version is due to the lower number of affine terms to be assembled online. Even bigger gains can be observed in the parabolic case in section 4, or for problems characterized by a large number of affine terms Q_a and Q_F.

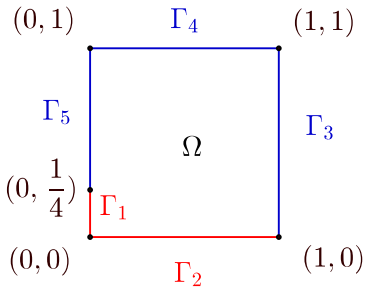


Figure 4. *Geometry PFS problem.*

2.2.2. Numerical test: Propagating front in a square (PFS). In this section we will test the reduced order stabilization method for a second test case where the parameter controls the angle of an internal layer. The problem we want to study is set over a unit square $\Omega \subset \mathbb{R}^2$, as sketched in Figure 4, has two parameters $\mu_1, \mu_2 \in \mathbb{R}$, and is as follows:

$$\begin{cases} -\frac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + (\cos\mu_2, \sin\mu_2)\cdot\nabla u(\boldsymbol{\mu}) = 0 & \text{in } \Omega, \\ u(\boldsymbol{\mu}) = 1 & \text{on } \Gamma_1 \cup \Gamma_2, \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_3 \cup \Gamma_4 \cup \Gamma_5. \end{cases} \tag{22}$$

Let us note that μ_1 is proportional to the Péclet number of the advection–diffusion problem, while μ_2 is the angle between the x axis and the direction of the constant advection field. The bilinear form associated to the problem is

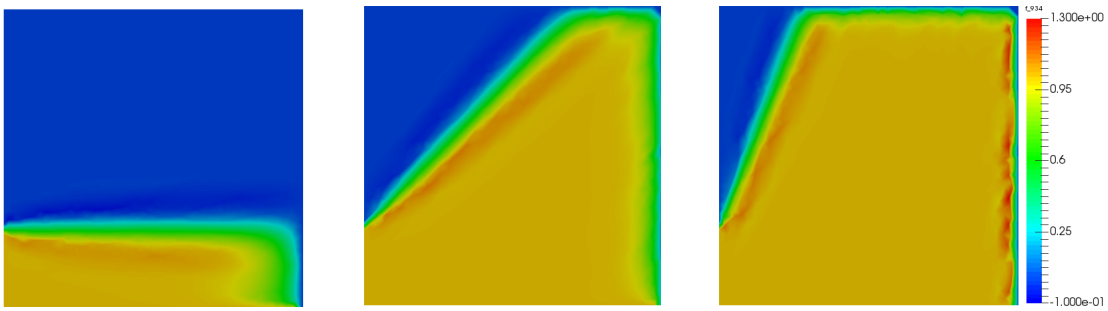
$$a(u, v; \boldsymbol{\mu}) = \int_\Omega \frac{1}{\mu_1}\nabla u \cdot \nabla v + (\cos\mu_2\, \partial_x u + \sin\mu_2\, \partial_y u)v. \tag{23}$$

We introduce again a triangulation $\mathcal{T}_h$ on the domain Ω, and we consider a $\mathbb{P}^1$ discretization. The corresponding stabilization term is

$$s(u^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) = \sum_{K\in\mathcal{T}^{\mathcal{N}}} \delta_K \int_K (\cos\mu_2, \sin\mu_2)\cdot\nabla u^{\mathcal{N}}\ (\cos\mu_2, \sin\mu_2)\cdot\nabla v^{\mathcal{N}}, \tag{24}$$

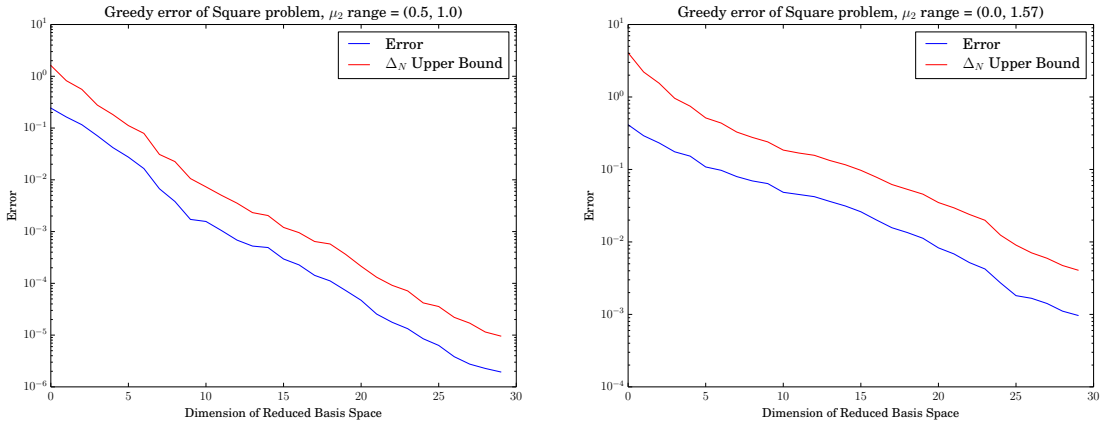
where δ_K is manually tuned according to μ_2. A few representative FE solutions are shown in Figure 5. The figure clearly shows that the direction of the advection fields largely affects the solution, which exhibits strong variations in energy norm [36]. For this reason, we test the RB method for two different choices of the parameter space, namely $\mathcal{D}^1 = \left[10^4, 10^5\right] \times [0.5, 1]$ and $\mathcal{D}^2 = \left[10^4, 10^5\right] \times [0, 1.57]$. Both choices are characterized by dominant advection; moreover, a wider range of angles is considered in $\mathcal{D}^2$ than in $\mathcal{D}^1$, resulting in a richer manifold of solutions.

The performance of the RB algorithm is shown in Figure 6 for $\mathcal{D}^1$ (a) and $\mathcal{D}^2$ (b). Only the offline–online stabilization case is reported, since the offline-only case gave poor results as in the previous test case. In both cases the stabilized reduced order method converges, reaching an error of around 10^{-6} for $\mathcal{D}^1$ and around 10^{-3} for $\mathcal{D}^2$. Computational times are 0.461346 seconds on average for a truth solution ($\mathcal{N} = 15626$), 0.034271 seconds for an RB solution ($N = 20$) with online stabilization, and 0.001862 seconds for an RB solution ($N = 20$) without online stabilization.



(a) $\mu_2 = 0, \delta_K = 2.1$. (b) $\mu_2 = 0.8, \delta_K = 1.4$. (c) $\mu_2 = 1.2, \delta_K = 0.7$.

Figure 5. *FE solution comparison, varying δ_K and μ_2.*



(a) $\mu_2 \in (0.5, 1)$. (b) $\mu_2 \in (0, 1.57)$.

Figure 6. *RB error and Δ_N error bound, varying μ_2 range.*

3. Stabilized weighted reduced basis algorithm for problems with uncertain parameters. The RB method formulated in section 2 assumed deterministic parameters; in contrast, for random parameters, a *weighted* reduced basis (wRB) has been proposed [9, 10] as an extension of the standard RB approach. The main idea of this method is to suitably assign a larger weight to those samples that are more "important." In this section, we will deal with problems with random distributed parameters, and we will compare the weighted method to the standard RB method for advection–diffusion problems with high Péclet numbers. Moreover, we will also provide an offline–online stabilization approach that can be useful in the case when stabilization involves large computations.

3.1. A brief introduction to the weighted reduced basis method. To discuss the wRB method [10], we introduce stochastic PDEs. Let Ω be an open set of $\mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$, and let $H_0^1(\Omega) \subset X \subset H^1(\Omega)$ be a functional space. Let $(A, \mathcal{F}, P)$ denote a complete probability space, where A is a set of outcomes $\omega \in A$, $\mathcal{F}$ is a σ-algebra of events,

and $P : \mathcal{F} \to [0,1]$ with $P(A) = 1$ is a probability measure [14]. A real-valued *random variable* is defined as a measurable function $Y : (A, \mathcal{F}) \to (\mathbb{R}, \mathcal{B})$, where $\mathcal{B}$ is the Borel σ-algebra on $\mathbb{R}$. Let $dF_Y(y)$ denote the distribution measure; i.e., $\forall\, B \subset \mathcal{D}$, $P(F \in B) = \int_B dF_Y(y)$. Provided that $dF_Y(y)$ is absolutely continuous with respect to the Lebesgue measure dy, which we assume hereafter to be the case, there exists a probability density function $\rho : \mathcal{D} \to \mathbb{R}$ such that $\rho(y)dy = dF_Y(y)$. Note that the new measure space $(\mathcal{D}, \mathcal{B}(\mathcal{D}), \rho(y)dy)$ is isometric to $(A, \mathcal{F}, P)$ under the random variable Y.

We define the probability Hilbert space $L^2(A) := \{v : A \to \mathbb{R} : \int_A v^2(\omega) dP(\omega) < \infty\}$ and $L^2_\rho(\mathcal{D}) := \{u : \mathcal{D} \to \mathbb{R} | \int_{\mathcal{D}} u^2(y)\rho(y)dy < \infty\}$, equipped with the equivalent norms (by noting that $v(\omega) = u(y(\omega))$)

$$||v||_{L^2(A)} := \left(\int_A v^2(\omega) dP(\omega)\right)^{1/2} = \left(\int_{\mathcal{D}} u^2(y)\rho(y)dy\right)^{1/2} =: ||u||_{L^2_\rho(\mathcal{D})}. \tag{25}$$

Let $v : \Omega \times A \to \mathbb{R}$ be a real-valued *random field*, which is a real-valued random variable defined on A for each $x \in \Omega$. We define the Hilbert space $S(\Omega) := L^2(A) \bigotimes H^1(\Omega)$, equipped with the inner product

$$(u, v) = \int_A \int_\Omega (uv + \nabla u \cdot \nabla v)\ dx\ dP(\omega) \quad \forall u, v \in S(\Omega), \tag{26}$$

where ∇ is the spatial gradient in Ω. The associated norm is defined as $||v||_{S(\Omega)} = \sqrt{(v, v)}$.

Now we can introduce *stochastic PDEs*. Given a random vector field $\boldsymbol{\mu} : A \to \mathbb{R}^p$, our stochastic advection–diffusion problem will be to find a random field $u(x; \boldsymbol{\mu}(\omega))$ such that

$$-\varepsilon(\boldsymbol{\mu}(\omega))\Delta u(\boldsymbol{\mu}(\omega)) + \boldsymbol{\beta}(\boldsymbol{\mu}(\omega)) \cdot \nabla u(\boldsymbol{\mu}(\omega)) = f(\boldsymbol{\mu}(\omega)) \quad \text{in } \Omega(\boldsymbol{\mu}(\omega)), \tag{27}$$

accompanied by suitable boundary conditions.

Now, we want to develop an algorithm that gives more importance to parameters with a higher probability of being chosen. The basic idea is to assign different weights to every value of parameter $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^p$ according to a prescribed weight function $w(\boldsymbol{\mu}) > 0$, and to use them during the construction of the RB space. The motivation is that when the parameter $\boldsymbol{\mu}$ has nonconstant weight function $w(\boldsymbol{\mu})$, e.g., stochastic problems with random inputs obeying probability distribution far from uniform type, the weighted approach can considerably attenuate the computational effort for large-scale computational problems. The wRB method consists of the same elements, namely greedy algorithm, a posteriori error estimate, and offline–online decomposition, as presented in section 2.1. In this section, we only highlight the new weighted steps.

Let $X^{\mathcal{N}}$ be a high-fidelity approximation space of X, equipped with the norm $|||.|||_{\boldsymbol{\mu}}$ defined in section 2.1.2. Moreover, let us define an equivalent weighted norm,

$$||u(\boldsymbol{\mu})||_w = w(\boldsymbol{\mu})||u(\boldsymbol{\mu})||_{\boldsymbol{\mu}} \quad \forall u \in X^{\mathcal{N}}, \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{28}$$

where $w : \mathcal{D} \to \mathbb{R}^+$ is a weighted function taking positive real values, which we assume to be continuous and bounded. We will denote by X_w the space X endowed with $||\cdot||_w$.

The greedy algorithm is thus modified to take the weighting into account, that is, to solve an optimization problem in $L^\infty(\mathcal{D}; X_w)$: at each step we are seeking a new parameter $\boldsymbol{\mu}^N \in \mathcal{D}$ such that

$$\boldsymbol{\mu}^N = \arg \sup_{\boldsymbol{\mu} \in \Xi_{train}} ||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})||_w, \tag{29}$$

where again u_N is the RB approximation of the truth solution $u^{\mathcal{N}}$. Here, Ξ_{train} is the discretized version of the parameter space $\mathcal{D}$. Instead of performing the true error, we use a weighted a posteriori error estimator Δ_N^w such that

$$||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})||_w \leq \Delta_N^w(\boldsymbol{\mu}). \tag{30}$$

The choice of the weight function $w(\boldsymbol{\mu})$ is made based on the desire to minimize the squared norm error of the RB approximation in the space $L^\infty(\mathcal{D}; X_w)$, i.e.,

$$\begin{aligned} \mathbb{E}[||u^{\mathcal{N}} - u_N||^2] &= \int_A \int_\Omega ||u^{\mathcal{N}}(\boldsymbol{\mu}(\omega)) - u_N(\boldsymbol{\mu}(\omega))||^2_{\boldsymbol{\mu}} dx \; dP(\omega) \\ &= \int_{\mathcal{D}} \int_\Omega ||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})||^2_{\boldsymbol{\mu}} \rho(\boldsymbol{\mu}) \; dx \; d\boldsymbol{\mu}, \end{aligned} \tag{31}$$

that we can bound with

$$\mathbb{E}\left[||u^{\mathcal{N}} - u_N||^2\right] \leq \int_{\mathcal{D}} \Delta_N(\boldsymbol{\mu})^2 \rho(\boldsymbol{\mu}) d\boldsymbol{\mu}, \tag{32}$$

where Δ_N is the RB error estimator introduced in section 2.1. This motivates us in the choice $w(\boldsymbol{\mu}) = \sqrt{\rho(\boldsymbol{\mu})}$. Finally, we set $\Delta_N^w(\boldsymbol{\mu}) := \Delta_N(\boldsymbol{\mu})\sqrt{\rho(\boldsymbol{\mu})}$ [10].

Another important aspect in the RB algorithm is the choice of the training set Ξ_{train}. While in the deterministic case we used uniform Monte Carlo sampling methods to choose elements from $\mathcal{D}$, in the stochastic context we can use a Monte Carlo sampling according to the distribution $\rho(\boldsymbol{\mu})$. We will see in the numerical test that this choice is important to improve the convergence of the error.

We refer the reader to [9, 10, 12] for further details on weighted reduced basis methods.

3.2. Stabilized weighted reduced basis methods. In this section we study a variant of the wRB method well suited for stochastic advection–diffusion equations with high Péclet numbers. In order to do so, we combine the stabilization of advective terms, introduced in section 2, with the weighting procedure of section 3.1.

As in section 2, for the moment we need to add SUPG stabilization terms to the weak form of the problem. This results in the following formulation:

$$\begin{aligned} &\text{Find } u^{\mathcal{N}}(\boldsymbol{\mu}(\omega)) \in X^{\mathcal{N}} \text{ s.t.} \\ &a_{stab}(u^{\mathcal{N}}(\boldsymbol{\mu}(\omega)), v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) = F_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) \quad v^{\mathcal{N}} \in X^{\mathcal{N}} \quad \forall \omega \in A, \end{aligned} \tag{33}$$

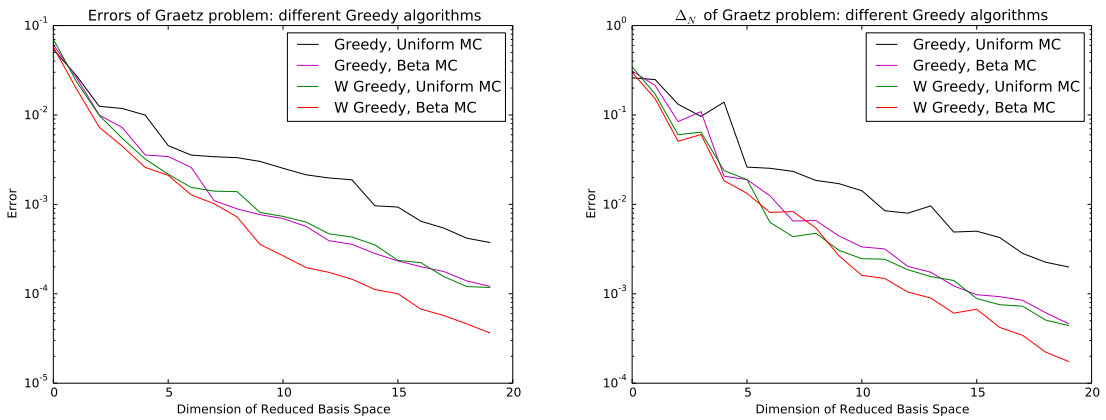
where a_{stab} and F_{stab} are defined in section 2. The most relevant difference with respect to the previous section is that $\boldsymbol{\mu} : A \to \mathcal{D}$ is a random vector instead of a deterministic parameter.

We test the proposed method with stochastic versions of the previous test cases (PG problem in section 2.2.1 and the PFS problem in section 2.2.2). In order to do so, we need to prescribe the distribution of $\boldsymbol{\mu}$; this will be done for each test case in the following sections. For the sake of exposition, results are presented only for the offline–online stabilization.

3.2.1. Numerical test: Poiseuille–Graetz problem. For the PG problem, we consider the range $\mathcal{D} = [10^1, 10^6] \times [0.5, 4]$ for the parameter $\boldsymbol{\mu}$. To give more importance to parameters with $\mu_1 \approx 10^5$, we use $X_1 \sim \text{Beta}(4,2)$ and $\mu_1 \sim 10^{1+5 \cdot X_1}$, while $X_2 \sim \text{Beta}(3,4)$ and $\mu_2 \sim 0.5 + 3.5 X_2$. We choose the Beta distribution because it takes values in a compact set,[1] resulting in $(\mu_1, \mu_2) \in \mathcal{D}$.

Next, we compare the performance of the reduction method for the different choices discussed in section 3, namely related to using a weighted or standard greedy algorithm, and to the sampling of the training set Ξ_{train}. We present in Figure 7 numerical results for the following four cases:

1. Classical greedy with uniform Monte Carlo sampling (black line);
2. Classical greedy with Beta Monte Carlo sampling (purple line);
3. Weighted greedy with uniform Monte Carlo sampling (green line);
4. Weighted greedy with Beta Monte Carlo sampling (red line).



(a) Error comparison. (b) Δ_N comparison.

Figure 7. *Greedy algorithms comparison for the Graetz problem.*

We used 200 samples for Ξ_{train} in each algorithm during the offline stage. We can see in Figure 7 the comparison between the average errors and the average Δ_N for these algorithms for a test set of size 100, with the same distribution as the training set. The results show that both weighting and a correct sampling are essential to obtain the best convergence results [48, 49]. Indeed, putting together these two aspects we get the best results, reaching an error that is one-tenth of the error of the classical greedy algorithm on uniform distribution.

In a similar way, instead of computing the average of the errors on the test set, we can

[1] The weighted approach would work as well for an unbounded (e.g., Gaussian) distribution. We use a Beta distribution in order to be able to present the comparison between a weighted and the classical approach. The latter would not be possible for Gaussian random variables unless the parameter domain were cut. Such a cut would be somehow arbitrary, since the classical approach does not exploit the underlying probability distribution.

also compute the mean of the error in a probability sense, i.e.,

$$\mathbb{E}[|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}] = \int_{\mathcal{A}} |||u^{\mathcal{N}}(\boldsymbol{\mu}(\omega)) - u_N(\boldsymbol{\mu}(\omega))|||_{\boldsymbol{\mu}(\omega)} dP(\omega) \tag{34}$$

$$= \int_{\mathcal{D}} |||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \rho(\boldsymbol{\mu}) d\boldsymbol{\mu}, \tag{35}$$

that we can approximate using some quadrature method. In particular, we will use the Monte Carlo method; i.e., we approximate (34) with

$$\mathbb{E}[|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}] \approx \frac{1}{M} \sum_{i=1}^{M} |||u^{\mathcal{N}}(\boldsymbol{\mu}_i) - u_N(\boldsymbol{\mu}_i)|||_{\boldsymbol{\mu}_i}, \tag{36}$$

where $\boldsymbol{\mu}_i$, $i = 1, \ldots, M$, are random parameters in the test drawn from a Beta distribution, while we approximate (35) with

$$\mathbb{E}[|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}] \approx \frac{1}{M} \sum_{j=1}^{M} |||u^{\mathcal{N}}(\boldsymbol{\mu}_j) - u_N(\boldsymbol{\mu}_i)|||_{\boldsymbol{\mu}_j} \rho(\boldsymbol{\mu}_j), \tag{37}$$

where $\boldsymbol{\mu}_j$, $i = 1, \ldots, M$, are drawn from a uniform distribution (on the same support) instead.

Results are nevertheless similar to those presented in Figure 7, and the same conclusions can be drawn. For instance, the probabilistic mean of the errors in the classical greedy method with uniform sampling and the weighted reduced mean with Beta sampling are $4.5485 \cdot 10^{-4}$ and $1.2807 \cdot 10^{-4}$, respectively.

3.2.2. Numerical test: Propagating front in a square. We can proceed in the same way for the PFS problem of section 2.2.2. In this section, the parameter range $\mathcal{D}$ is $\left[10^4, 10^5\right] \times [0, 1.5]$. Also, in this case μ_1 and μ_2 depend on randomly distributed Beta variables, i.e., $\mu_1 \sim 10^4 + 9 \cdot 10^4 \cdot X_1$ and $\mu_2 \sim 1.5 \cdot X_2$, where $X_1 \sim \text{Beta}(3, 4)$, while $X_2 \sim \text{Beta}(4, 2)$.

As for the previous test case, we compare the classical greedy method with uniform Monte Carlo to the wRB method with Beta Monte Carlo distribution. The comparison, shown in Figure 8, provides results which are very similar to the PG problem. Indeed, the wRB method with Beta distribution is converging faster than the classical method. Also, the mean errors in the probabilistic sense of (36) show a similar behavior: for an RB space of dimension $N = 20$, the stabilized weighted method with Beta distribution produces a mean error of $1.7803 \cdot 10^{-3}$, while the classical approach gives a mean error of $7.9362 \cdot 10^{-3}$.

3.3. Selective online stabilization of the weighted reduced basis approach. In this section we want to optimize computational costs in the *online* phase of the RB method. Indeed, the stabilization procedure can lead to an increase in the number Q_a and/or Q_f of affine terms, which in turn may lead to larger online times required for the assembly of the linear system or for the evaluation of the error estimator. In this section we propose a procedure to selectively enable online stabilization only when required. In the entire section we keep the RB produced in the previous section for $N = 20$.

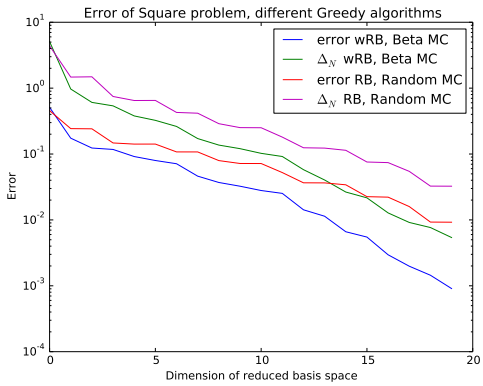
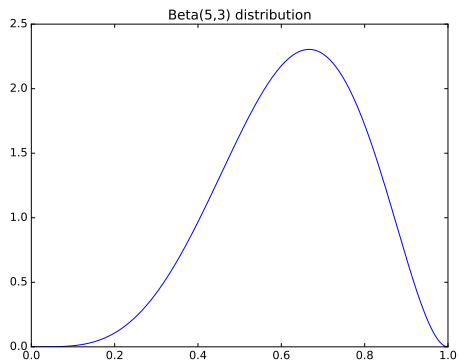
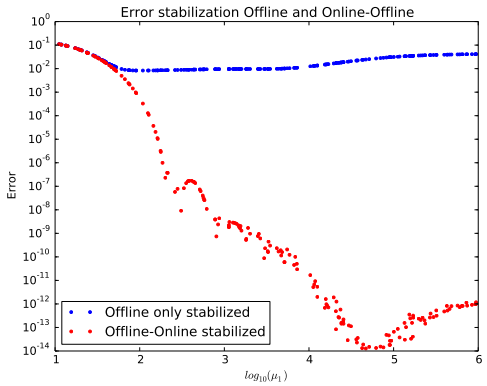


Figure 8. *Greedy algorithms comparison for the PFS problem.*



(a) Beta(5,3) distribution.



(b) Errors with stabilization offline and offline–online.

Figure 9. *Error and density of uniform Monte Carlo test set.*

3.3.1. Numerical test: Poiseuille–Graetz problem. Let us consider first the PG example, with Beta distribution over parameter $\boldsymbol{\mu}$, similarly to section 3.2.1. In what follows, we assume that $\mu_1 \in [10, 10^6]$, $\mu_1 \sim 10^{1+5 \cdot X_1}$ where $X_1 \sim \text{Beta}(5, 3)$. To simplify the discussion of the results we further assume that $\mu_2 \equiv 1$.

While carrying out the online stage of the proposed stabilized wRB method, we can choose whether to apply online stabilization or not. Figure 9(b) shows the resulting error on a test set (that we have taken with a uniform Monte Carlo sampling), sorted by increasing values of μ_1, considering both options. We can observe that for low Péclet number ($\mu_1 \leq 10^2$), offline–online stabilization and offline-only stabilization produce very similar results. Thus, we would prefer the less expensive offline-only stabilization procedure. There, the error is high because the samples selected from the weighted greedy in the offline phase are all concentrated where the density of probability is higher (high Péclet). For this reason the low Péclet number zone is poorly represented. Moreover, in the regions where the density of $\boldsymbol{\mu}$ is very small, even a

large error would be less relevant in terms of the probabilistic mean error (34). So, we should consider the idea of enabling the more expensive online stabilization only for parameters with high density (which would have more of an effect on the mean error) or parameters with large Péclet numbers (where the more expensive assembly is fully justified by the convection dominated regime).

Let us start by considering the case where we want to stabilize online solutions depending on Péclet numbers. First, we establish a threshold at a certain Péclet number $\widetilde{\mu}_1$. For parameters $\mu_1 > \widetilde{\mu}_1$ we will use both online and offline stabilization, while for parameter $\mu_1 \leq \widetilde{\mu}_1$ we will use only offline stabilization. See Figure 10 for a graphical representation for $\widetilde{\mu}_1 = 10^3$.

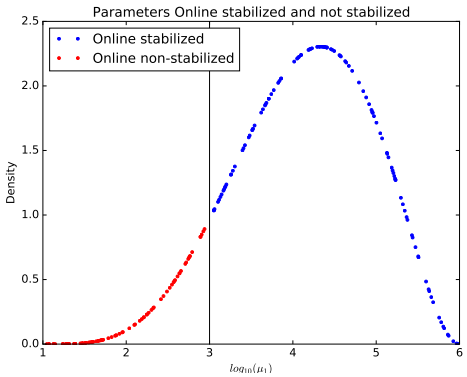


Figure 10. *Péclet discriminant; black line is the Péclet threshold.*

For different thresholds $\widetilde{\mu}_1$ we can compute the error in the sense of (34), as we can see in the following table:

Threshold $\widetilde{\mu}_1$	Error	Percentage nonstabilized
10^1	$7.9673 \cdot 10^{-4}$	0%
$10^{1.5}$	$8.0704 \cdot 10^{-4}$	10%
10^2	$10.0060 \cdot 10^{-4}$	20%
$10^{2.5}$	$18.2806 \cdot 10^{-4}$	33%
10^3	$33.4593 \cdot 10^{-4}$	45%
10^6	0.021128	100%

Considering that the best attainable error was of $7.967 \cdot 10^{-4}$, we can say that until $\widetilde{\mu}_1 = 10^2$, we are not worsening considerably the error (less than an order of magnitude). At the same time, we can save online time on the assembly of terms related to the stabilization coefficient for 20% of our test set (that was uniformly distributed).

The other natural gauge to decide whether or not to stabilize online is the density $\rho(\boldsymbol{\mu})$. Let $\widetilde{\nu}$ be a prescribed tolerance; we will not stabilize parameters $\boldsymbol{\mu}$ on the tail I of the distribution such that

$$\int_I \rho(\boldsymbol{\mu}) d\boldsymbol{\mu} = \widetilde{\nu}, \tag{38}$$

where I is a set $\{\boldsymbol{\mu} : \rho(\boldsymbol{\mu}) \leq \widetilde{\rho}\}$ for some suitable $\widetilde{\rho}$ which can be easily found numerically as a function of $\widetilde{\nu}$. In Figure 11 we can see an example for $\widetilde{\nu} = 10\%$.

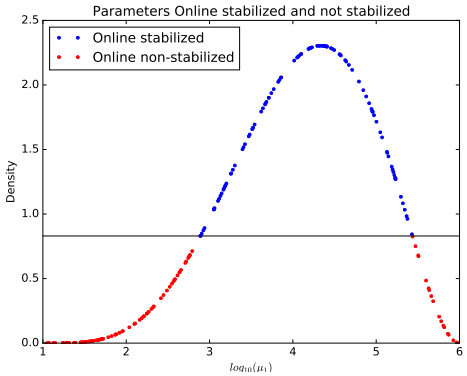


Figure 11. *Density discriminant; the black line is the density threshold.*

In the following table, we summarize some results for different thresholds $\widetilde{\nu}$ (and, correspondingly, $\widetilde{\rho}$):

Threshold $\widetilde{\nu}$	Threshold $\widetilde{\rho}$	Error	Percentage nonstabilized
0	0	$7.9673 \cdot 10^{-4}$	0%
0.001	0.02233	$9.3222 \cdot 10^{-4}$	15%
0.002	0.04423	$9.6456 \cdot 10^{-4}$	17%
0.005	0.09094	$14.7861 \cdot 10^{-4}$	21%
0.01	0.13877	$15.9482 \cdot 10^{-4}$	25%
0.02	0.21433	$25.6017 \cdot 10^{-4}$	30%
0.05	0.38244	$49.1931 \cdot 10^{-4}$	38%
0.1	0.89068	$66.7488 \cdot 10^{-4}$	45%
1	∞	0.021128	100%

We have that errors computed using the density discriminant are less accurate than ones computed with the Péclet discriminant. Indeed, for the same percentage of nonstabilized solutions (for example, 45%) we have bigger errors in the density discriminant approach ($66 \cdot 10^{-4}$ instead of $33 \cdot 10^{-4}$). This is due to the enormous difference between online stabilized and online nonstabilized solutions for high Péclet numbers (Figure 9(b)), with the latter resulting in considerably larger errors.

3.3.2. Numerical test: Propagating front in a square. Let us now consider the PFS problem with fixed $\mu_1 \equiv 10^5$, while $\mu_2 \sim 0.5 + 3.5X_2 \in [0, 1.5]$ where $X_2 \sim \text{Beta}(4, 2)$. We have decided to fix the Péclet number since results in section 2.2.2 show that the solution is most sensible for the parameter μ_2, which represents the angle of the propagating front.

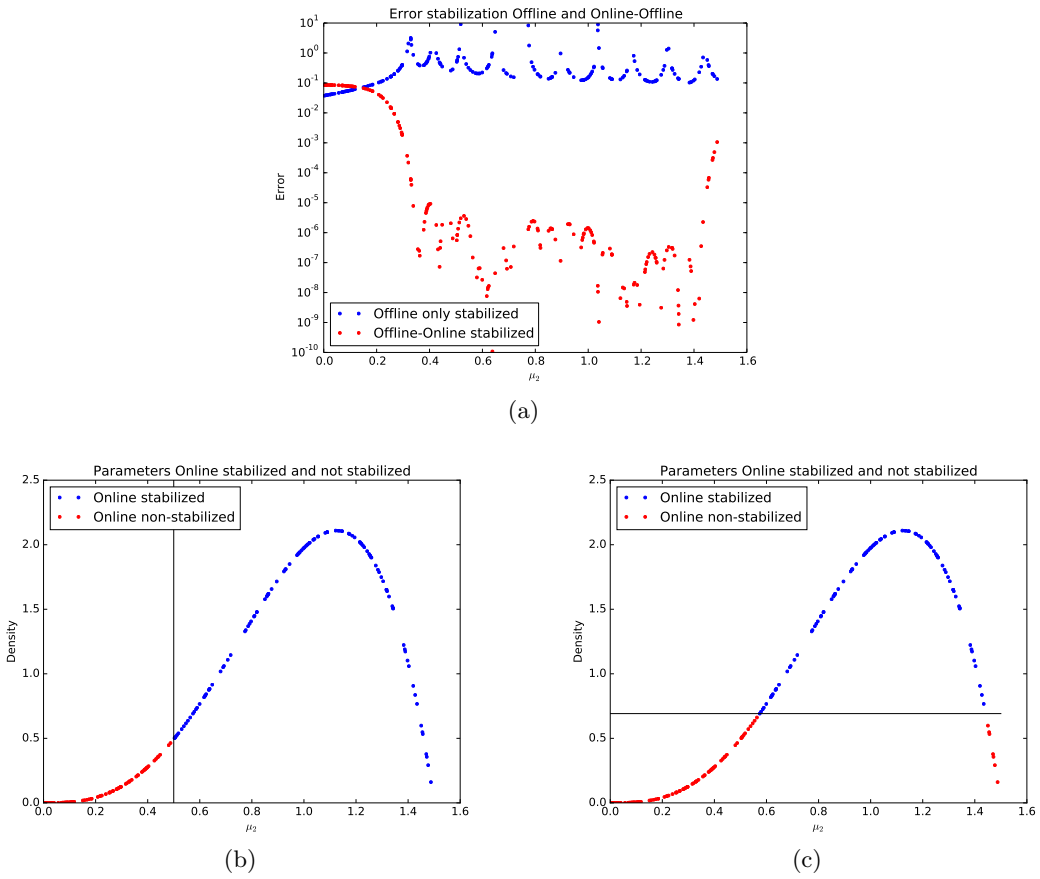


Figure 12. (a) *Errors with stabilization offline and offline–online.* (b) *Angle discriminant; the black line is the angle threshold.* (c) *Density discriminant; the black line is the density threshold.*

Errors for online stabilized and online nonstabilized solutions over a uniform Monte Carlo test set of 200 elements are provided in Figure 12(a) for increasing values of μ_2. We can notice that offline–online stabilized errors of solutions with small angles (Figure 12(a), $\mu_2 \lesssim 0.2$) are bigger than offline-only stabilized errors. This is due to the fact that the density of that region of the parameter range is very small, and thus the weighted Greedy algorithm picks very few parameters in that region. In a similar way, we also notice that solutions for $\mu_2 \approx 1.5$ are not well approximated. Indeed, in the offline-only stabilized case the lack of stabilization has a poor effect on the reduced order solution for any $\mu_2 \gtrsim 0.2$, while in the offline–online stabilized case the low density of $\mu_2 \gtrsim 1.4$ leads the wRB selection to choose few parameters $\mu_2 \approx 1.5$ during the offline stage.

Thus, in a similar way to the previous test case, we propose selective online stabilization criteria depending on either a threshold on the parameter (the angle μ_2 in this case, rather than the Péclet number) or on the probability distribution. Let us start with a discussion of the former choice, leading to online stabilization for angles greater than a certain threshold

$\widetilde{\mu}_2$ (see, e.g., Figure 12(b)). The error for different thresholds $\widetilde{\mu}_2$ is tabulated as follows:

Threshold $\widetilde{\mu}_2$	Error	Percentage nonstabilized
0	0.01416	0%
0.1	0.01400	6%
0.2	0.01506	16%
0.3	0.04056	23%
0.4	0.11810	30%
0.5	0.20365	37%
1.5	0.82998	100%

We can observe that at the beginning the error decreases as the threshold increases, while it slowly increases after a critical angle between 0.1 and 0.2. Due to this, we consider a threshold $\widetilde{\mu}_2 = 0.2$ to be optimal in order not to increase the error and to save 16% of online stabilization computations.

As for the PG example, we can also test a criterion based on a density threshold (see, e.g., Figure 12(c)). The following table shows different errors for different density thresholds:

Threshold $\widetilde{\nu}$	Threshold $\widetilde{\rho}$	Error	Percentage nonstabilized
0	0	0.01416	0%
0.001	0.02271	0.01400	13%
0.002	0.04600	0.01506	16%
0.005	0.10237	0.02269	20%
0.01	0.13598	0.04658	25%
0.02	0.26309	0.11158	30%
0.05	0.51855	0.20613	38%
0.1	0.72557	0.32034	46%
1	∞	0.82998	100%

In this case, a negligible increase of the error is obtained for $\widetilde{\nu} = 0.002$, allowing us to save more than 15% of stabilized online computations. Further computational savings up to 25% can be obtained for $\widetilde{\nu} = 0.01$ at the expense of a larger error. We notice that in this case both criteria give similar results; this is due to the fact that errors are large for both offline-only and offline–online stabilization methods when μ_2 is large or when density ρ is small.

Remark 3.1. Let I be the region of the parameter space where an offline-only stabilized solution is selected, and let $\mathcal{D} \setminus I$ denote the complement region in which the offline–online stabilized method is queried. Let $u_N^I(\boldsymbol{\mu})$ denote the corresponding reduced order solution for $\boldsymbol{\mu} \in I$, and similarly $u_N^{\mathcal{D}\setminus I}(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \mathcal{D} \setminus I$. To ease the notation, we will denote the online solution by $u_N(\boldsymbol{\mu})$ when no confusion arises.

The selective procedure for online stabilization can be automatically tuned according to a prescribed tolerance on the probabilistic mean error $\mathbb{E}\left[|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}\right]$. In order to estimate the mean error, we recall the standard error estimation (10) for $\boldsymbol{\mu} \in \mathcal{D} \setminus I$, and the following error estimation:

$$
\begin{aligned}
|||u_N^I(\boldsymbol{\mu}) - u^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \Delta_N^I(\boldsymbol{\mu}) := h_{max}(\boldsymbol{\mu}) C(\boldsymbol{\mu}) ||\boldsymbol{\beta} \cdot \nabla u^{\mathcal{N}}(\boldsymbol{\mu})||_{L^2(\Omega_p(\boldsymbol{\mu}))} \\
+ (1 + h_{max}(\boldsymbol{\mu}) C(\boldsymbol{\mu})^2 ||\boldsymbol{\beta}||_{L^\infty(\Omega_p(\boldsymbol{\mu}))}) \varepsilon^*
\end{aligned} \tag{39}
$$

for $\boldsymbol{\mu} \in I$ [37], where $C(\boldsymbol{\mu})$ is the constant of the equivalence between H^1 and $|||\cdot|||_{\boldsymbol{\mu}}$ norms, h_{max} is the maximum mesh size, while ε^* is the tolerance of the Greedy algorithm [37].

Thus, combining these two error estimators, we get that

$$\mathbb{E}\left[|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N|||_{\boldsymbol{\mu}}\right] \leq (1-\widetilde{\nu}) \max_{\boldsymbol{\mu}\in\mathcal{D}\setminus I} \Delta_N(\boldsymbol{\mu}) + \widetilde{\nu} \max_{\boldsymbol{\mu}\in I} \Delta_N^I(\boldsymbol{\mu}), \tag{40}$$

which, for a given tolerance $\widetilde{e}$ on the mean error, allows us to compute $\widetilde{\nu}$ such that

$$(1-\widetilde{\nu}) \max_{\boldsymbol{\mu}\in\mathcal{D}\setminus I} \Delta_N(\boldsymbol{\mu}) + \widetilde{\nu} \max_{\boldsymbol{\mu}\in I} \Delta_N^I(\boldsymbol{\mu}) < \widetilde{e}.$$

Remark 3.2. We remark that this selective approach for online stabilization is peculiar to stochastic problems. Indeed, it is the density distribution and the relative importance of each sample in the computation of the probabilistic mean that drive the selection process. Such a weighting is lacking in a deterministic setting, where all samples are equally probable during the online stage.

4. Stabilized weighted reduced basis method for stochastic parabolic equations. In this section we extend our investigation to stochastic time-dependent advection–diffusion equations. Stabilization of advection–diffusion parabolic equations with high Péclet numbers has been studied in several works with different stabilization methods [7]. We will adapt SUPG stabilization for FE methods on parabolic equations to the RB method, as suggested in [36, 37, 38, 39]. The reduction will employ a proper orthogonal decomposition (POD)-greedy procedure [19, 35, 40] during the offline stage. We refer the reader to [45, 46] for very recent wRB variants for stochastic heat equations.

As we did for stochastic elliptic equations, we define a *parameter domain* $\mathcal{D}$ as a closed subset of $\mathbb{R}^p$, and we call $\boldsymbol{\mu}$ a random field with values in $\mathcal{D}$. Again, let Ω be a bounded open subset of $\mathbb{R}^d$ $(d = 1, 2, 3)$ with regular boundary $\partial\Omega$, and let X be a functional space such that $H_0^1(\Omega) \subset X \subset H^1(\Omega)$. For each outcome $\omega \in A$, and corresponding realization $\boldsymbol{\mu}(\omega) \in \mathcal{D}$, we define the continuous, coercive bilinear form a and the continuous, bilinear, symmetric form m such that they satisfy the *affinity* assumption (4), and define a linear form F which satisfies the *affine* assumption (5). Finally, we denote the time domain as $I = [0, T]$, where T is the final time.

We can now define the weak form of the continuous stochastic problem as follows:

(41)
Find $u(t; \boldsymbol{\mu}(\omega)) \in X \quad \forall t \in I, \quad \forall \omega \in A \quad$ continuous in t
s.t. $m(\partial_t u(t; \boldsymbol{\mu}(\omega)), v) + a(u(t; \boldsymbol{\mu}(\omega)), v; \boldsymbol{\mu}(\omega)) = g(t)F(v; \boldsymbol{\mu}(\omega)) \quad \forall v \in X, \quad \forall t \in I, \quad \forall \omega \in A$
given the initial value $u(0; \boldsymbol{\mu}(\omega)) = u_0 \in L^2(\Omega)$,

where $g : I \to \mathbb{R}$ is a *control function* such that $g \in L^2(I)$. We choose a right-hand side of the form $g(t)F(v; \boldsymbol{\mu})$, as is usual in the RB framework [18, 40], in order to ease the offline–online computational decoupling.

4.1. Discretization and reduced basis formulation. To discretize the time-dependent problem (41), we follow the approach used in [18, 20, 34, 40], that is, we use finite differences

in time and FE in space discretization [41]. We start by discretizing the spatial part of the problem (resulting in a mesh denoted by $\mathcal{T}_h$) and the temporal part (resulting in discrete time steps $\{t_j = j \cdot \Delta t\}_{j=0}^{J}$). We thus define the FE truth approximation space $X^{\mathcal{N}}$ and denote its basis with $\{\phi_i\}_{i=1}^{\mathcal{N}}$. The fully *discretized* problem reads

$$
\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u_j^{\mathcal{N}}(\boldsymbol{\mu}(\omega)) \in X^{\mathcal{N}} \\
&\quad \text{s.t. } \frac{1}{\Delta t} m(u_j^{\mathcal{N}}(\boldsymbol{\mu}(\omega)) - u_{j-1}^{\mathcal{N}}(\boldsymbol{\mu}(\omega)), v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) + a(u_j^{\mathcal{N}}(\boldsymbol{\mu}(\omega)), v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) \\
&\qquad = g(t_j) F(v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \quad \forall \omega \in A, \\
&\quad \text{given the initial condition } u_0^{\mathcal{N}} \\
&\quad \text{s.t. } (u_0^{\mathcal{N}}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.
\end{aligned} \tag{42}
$$

The latter problem uses the *backward Euler–Galerkin* discretization, but we can resort to other theta-methods (e.g., Crank–Nicolson) or to high order method (e.g., Runge–Kutta) [41].

The RB formulation of the problem (42) is based on hierarchical RB space, as for the steady case, employing a POD reduction over the time trajectory and a greedy selection over the parameter space [19, 35]. The algorithm can be seen as a greedy algorithm in the parameter space with a further compression by POD for the space trajectory.

At each step of the greedy algorithm we search the parameter $\boldsymbol{\mu}^*$ which maximizes, over the training set Ξ_{train}, an error estimator for the following quantity:

$$
|||\boldsymbol{e}_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} = \left(m(e_{N,J}^{\mathcal{N}}(\boldsymbol{\mu}), e_{N,J}^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}) + \sum_{j=1}^{J} a(e_{N,j}^{\mathcal{N}}(\boldsymbol{\mu}), e_{N,j}^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \Delta t \right)^{\frac{1}{2}}, \tag{43}
$$

where $e_{N,j}^{\mathcal{N}}(\boldsymbol{\mu}) = u_j^{\mathcal{N}}(\boldsymbol{\mu}) - u_{N,j}^{\mathcal{N}}(\boldsymbol{\mu})$. We remark that, as in section 2.1, an inexpensive a posteriori error bound for (43) can be derived (see [18]), which in particular does not require any $\mathcal{N}$-dependent computation (e.g., it does not require the time trajectory to be computed for every $\boldsymbol{\mu}$ in the training set). We will continue denoting by Δ_N the resulting error estimator, even though its expression is different from the one in section 2.1; we refer the reader to [18] for more details.

Once the parameter is chosen, we project the time evolution of the solution of this parameter on the orthogonal space of the current RB space $X_N^{\mathcal{N}}$. This projection ensures that, at each Greedy iteration, only *new* information is added to the RB. To set the notation, denote by $\mathcal{P}_N : X^{\mathcal{N}} \to X_N^{\mathcal{N}}$ the projection onto the current RB$X_N^{\mathcal{N}}$. We then define $u_j^{\perp}(\boldsymbol{\mu}^*) = u_j(\boldsymbol{\mu}^*) - \mathcal{P}_N(u_j(\boldsymbol{\mu}^*))$ for $j = 1, \dots, J$.

As a further compression of the resulting time trajectory, we compute a POD on $\{u_j^{\perp}(\boldsymbol{\mu}^*)\}_{j=1}^{J}$ and collect the first few POD modes (up to a prescribed tolerance) into a space denoted by $Y_N^{\mathcal{N}}$. The resulting RB space to be used at the $(N+1)$th greedy iteration is then defined as $X_{N+1}^{\mathcal{N}} = X_N^{\mathcal{N}} \oplus Y_N^{\mathcal{N}}$.

The RB formulation of the problem can be obtained by substituting the RB space $X_N^{\mathcal{N}}$ by $X^{\mathcal{N}}$ in (42).

4.2. Streamlined/upwind Petrov–Galerkin stabilization method for parabolic problems.

In this section we briefly introduce the SUPG method for time-dependent problems [7, 28].

The idea is the same as in the steady case: we have to add terms to bilinear forms in order to improve stability. The stabilization term is almost the same as in the steady case, but now we have to consider also the time dependency to guarantee the strong consistency. We thus set

$$s(w^{\mathcal{N}}(t), v^{\mathcal{N}}) = \sum_{K \in \mathcal{T}_h} \delta_K \left(\partial_t w^{\mathcal{N}}(t) + L w^{\mathcal{N}}(t), \frac{h_k}{|\boldsymbol{\beta}(\boldsymbol{\mu}(\omega))|} L_{SS} v^{\mathcal{N}} \right)_K, \tag{44}$$

where $w^{\mathcal{N}}(t) \in X^{\mathcal{N}}$ for each $t \in I$, $v^{\mathcal{N}} \in X^{\mathcal{N}}$, and $(\cdot,\cdot)_K$ is the usual L^2 scalar product, restricted to the element K. Here L is the steady advection–diffusion operator and L_{SS} is its skew-symmetric part.

Thus, we can define the backward Euler–SUPG formulation of the problem by substituting the forms m, a, and F in (42) with

$$\begin{aligned}
m_{stab}(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) &= m(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) + \sum_{K \in \mathcal{T}_h} \delta_K \left(w^{\mathcal{N}}, \frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu}(\omega))|} L_{SS} v^{\mathcal{N}} \right)_K, \\
a_{stab}(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) &= a(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) + \sum_{K \in \mathcal{T}_h} \delta_K \left(L w^{\mathcal{N}}, \frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu}(\omega))|} L_{SS} v^{\mathcal{N}} \right)_K, \\
F_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) &= F(v^{\mathcal{N}}; \boldsymbol{\mu}(\omega)) + \sum_{K \in \mathcal{T}_h} \delta_K \left(f, \frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu}(\omega))|} L_{SS} v^{\mathcal{N}} \right)_K,
\end{aligned} \tag{45}$$

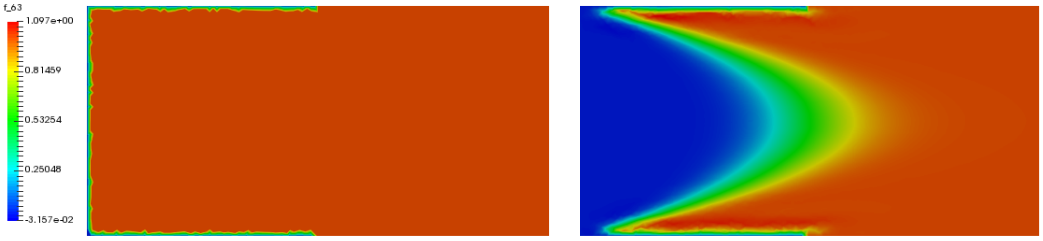
where K are the elements which form the mesh $\mathcal{T}_h$ and f can be a source term of the advection–diffusion equation or a lifting of the Dirichlet boundary data. For the analysis of stability and convergence of the method we refer the reader to [26].

4.3. Numerical tests for stochastic parabolic problems. We now show some numerical results of the stabilized RB method for stochastic parabolic PDEs, extending to the time-dependent case the problems in sections 3.2.1 and 3.2.2. For the sake of exposition we will show the results only for the offline–online stabilization. A few representative FE solutions are provided in Figure 13 for the parabolic PG problem and in Figure 14 for the parabolic front propagation test.

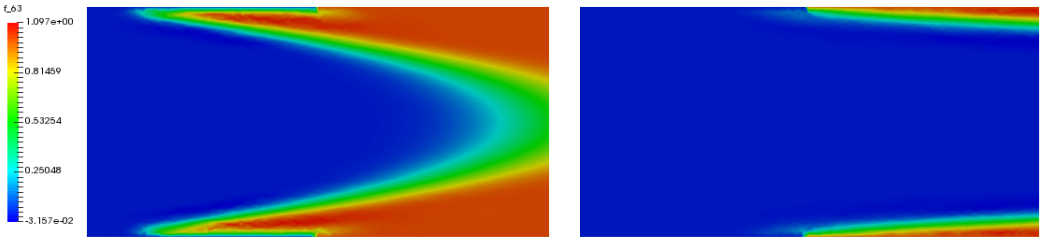
We show in Figures 15 and 16 the average error on a test set for both the parabolic PG problem (a) and the parabolic PFS test (b), respectively, in the deterministic and the stochastic case. The error is defined in (43), while the error estimator Δ_N is as in [18]. We compare in Figure 16 the classical RB algorithm (with uniform Monte Carlo sampling) and the wRB algorithm (with sampling according to the distribution of $\boldsymbol{\mu}$). The comparison shows that, also for the parabolic problem, proper weighting and suitable sampling allow us to improve the accuracy of the resulting reduced order model (especially in the case of the parabolic front problem) and the reliability of the error estimator (in both test cases).

Similar results hold for the probabilistic mean indicator introduced in (34), which we extend to the unsteady case as

$$\mathbb{E}[||\boldsymbol{u}^{\mathcal{N}} - \boldsymbol{u}_N^{\mathcal{N}}||^2] := \sum_{j=1}^{J} \int_{\mathcal{D}} ||u_j^{\mathcal{N}}(\boldsymbol{\mu}) - u_{N,j}^{\mathcal{N}}(\boldsymbol{\mu})||_{\boldsymbol{\mu}}^2 \rho(\boldsymbol{\mu}) d\boldsymbol{\mu} \tag{46}$$

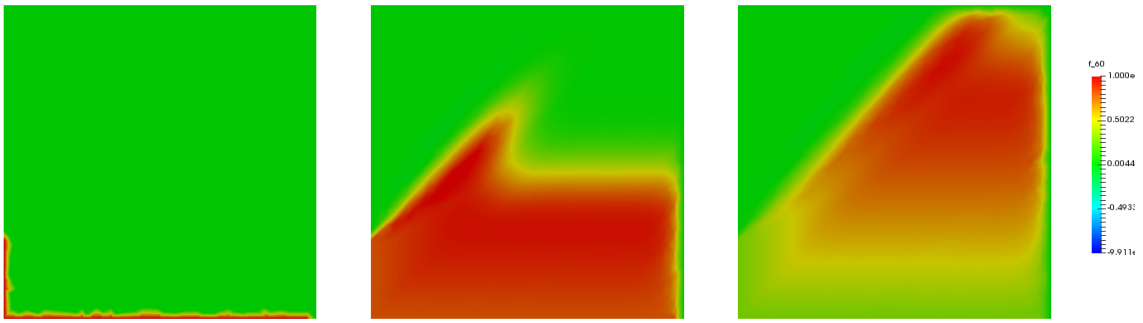


(a) $t = 0$ sec. (b) $t = 1$ sec.



(c) $t = 2$ sec. (d) $t = 7$ sec.

Figure 13. *Plot of FE solution for a parabolic PG problem at different times at* $\mu_1 = 1$ *and* $\mu_2 = 1 \cdot 10^4$.

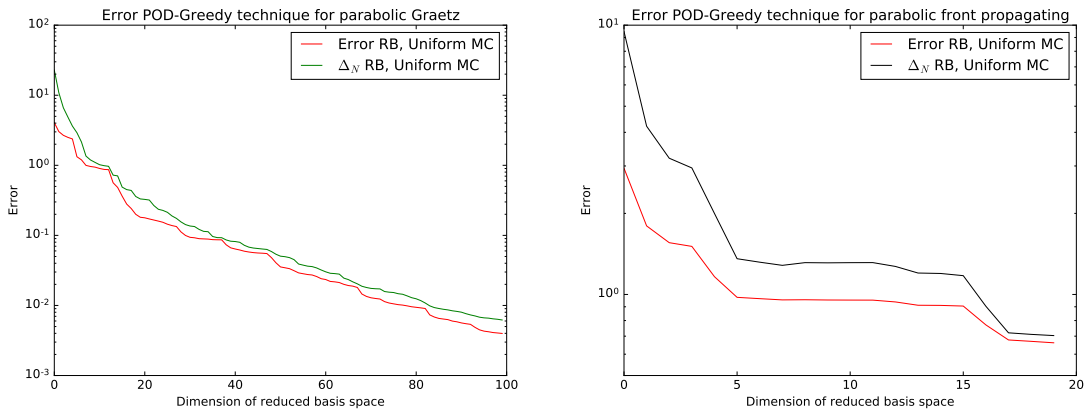


(a) $t = 0$ sec. (b) $t = 0.64$ sec. (c) $t = 1.28$ sec.

Figure 14. *Plot of FE solution for parabolic PFS problem at different times,* $\mu_1 = 2 \cdot 10^4$, $\mu_2 = 0.8$.

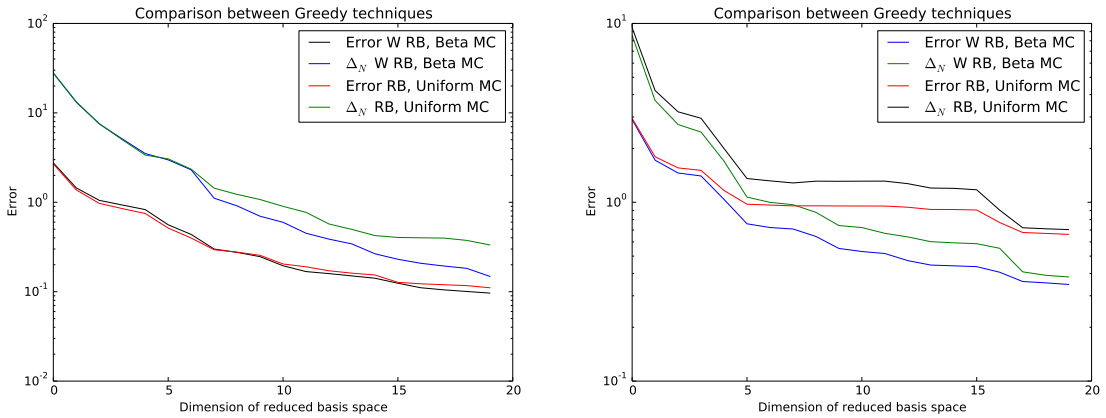
and approximate with the Monte Carlo quadrature procedure. By doing this we obtain, given the PG problem with an RB space of dimension 20, an error of $8.3248 \cdot 10^{-2}$ for the classic greedy algorithm and obtain $7.6318 \cdot 10^{-2}$ for the wRB algorithm, respectively. For the PFS problem we have that the classic greedy algorithm produces an error of 0.3196, while the weighted algorithm gives 0.2343.

We must make a short remark on computational times in the parabolic problem. In the PG problem, for one true parabolic solution we need 132.382 seconds, while for the RB problem with $N = 20$ basis functions we need only 0.356224 seconds. For a PFS true solution we need 17.2846 seconds but only 0.125266 seconds for an RB solution with $N = 20$ basis functions. These results justify all the computational costs of the offline phase.



(a) PG problem. (b) PFS problem.

Figure 15. *Greedy algorithms comparison for parabolic problems.*



(a) PG problem. (b) PFS problem.

Figure 16. *Greedy algorithms comparison for parabolic problems.*

5. Conclusions. In this work we have dealt with stabilization techniques for the approximation of advection dominated problems using an RB approach in a stochastic framework, both in the steady and unsteady case. To perform a stabilization in the RB algorithm, we have studied the SUPG [42] stabilization for the FE method and introduced two RB stabilization algorithms: the online–offline stabilization, which uses SUPG stabilized forms in both stages (offline and online), and the offline-only stabilization, which uses the original (nonstabilized) forms for the online stage. The underlying idea was to obtain a stable RB approximation from the stable FE approximation, with reasonable computational times and, at the same time, a very good accuracy.

We then introduced stochastic equations and the wRB method [10]. We formulated a stabilized wRB method for advection–diffusion problems with random input parameters. Numerical test cases clearly highlight the importance of the weighting procedure, as well as the necessity of a proper sampling of the parameter space, according to the probability distribution of $\boldsymbol{\mu}$. Moreover, we introduced a procedure to selectively enable online stabilization when required. This allows us to reduce the number of terms to be assembled in the affine expansion, with a negligible worsening of the error, which remains of the same order as that for the previous strategies.

Finally, we have generalized these methods to parabolic problems, producing a stabilized RB approach for unsteady cases [19, 37], starting from SUPG stabilized parabolic FE methods [7, 28].

Possible further developments of this topic could be the application of these methods to more complex geometries, e.g., non–affinely parametrized ones, requiring some empirical interpolation preprocessing [6, 29]. Moreover, the method could be tested on higher dimensional parameter spaces $\mathcal{D}$, using Monte Carlo or quasi–Monte Carlo strategies, and on other types of probability distributions.

Acknowledgments. The computations in this work have been performed with the RBniCS [5] library, developed at SISSA mathLab, which is an implementation in FEniCS [30] of several reduced order modeling techniques; we acknowledge the developers of and contributors to both libraries.

REFERENCES

[1] I. Akhtar, A. H. Nayfeh, and C. J. Ribbens, *On the stability and extension of reduced-order Galerkin models in incompressible flows*, Theoret. Comput. Fluid Dynam., 23 (2009), pp. 213–237.

[2] S. Ali, F. Ballarin, and G. Rozza, *Stabilized reduced basis methods for parametrized steady Stokes and Navier-Stokes equations*, submitted.

[3] J. Baiges, R. Codina, and S. Idelsohn, *Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations*, Internat. J. Numer. Methods Fluids, 72 (2013), pp. 1219–1243.

[4] F. Ballarin, E. Faggiano, S. Ippolito, A. Manzoni, A. Quarteroni, G. Rozza, and R. Scrofani, *Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization*, J. Comput. Phys., 315 (2016), pp. 609–628.

[5] F. Ballarin, A. Sartori, and G. Rozza, *RBniCS—Reduced Order Modelling in FEniCS*, http://mathlab.sissa.it/rbnics, 2015.

[6] M. Barrault, Y. Maday, N. Nguyen, and A. 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.

[7] A. Brooks and T. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 32 (1982), pp. 199–259.

[8] T. Chacón Rebollo, E. Delgado Ávila, M. Gómez Mármol, F. Ballarin, and G. Rozza, *On a certified Smagorinsky reduced basis turbulence model*, SIAM J. Numer. Anal., 55 (2017), pp. 3047–3067, https://doi.org/10.1137/17M1118233.

[9] P. Chen, *Model Order Reduction Techniques for Uncertainty Quantification Problems*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne EPFL, https://infoscience.epfl.ch/record/198689/files/EPFL_TH6118.pdf, 2014.

[10] P. Chen, A. Quarteroni, and G. Rozza, *A weighted reduced basis method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 51 (2013), pp. 3163–3185, https://doi.org/10.1137/130905253.

[11] P. Chen, A. Quarteroni, and G. Rozza, *A weighted empirical interpolation method: A priori convergence analysis and applications*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 943–953.

[12] P. Chen, A. Quarteroni, and G. Rozza, *Reduced basis methods for uncertainty quantification*, SIAM/ASA J. Uncertain. Quantif., 5 (2017), pp. 813–869, https://doi.org/10.1137/151004550.

[13] L. Dedè, *Reduced basis method for parametrized elliptic advection-reaction problems*, J. Comput. Math., 28 (2010), pp. 122–148.

[14] R. Durrett, *Probability Theory and Examples*, Cambridge University Press, Cambridge, UK, 2010.

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

[16] F. Gelsomino and G. Rozza, *Comparison and combination of reduced-order modelling techniques in 3D parametrized heat transfer problems*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 371–394.

[17] S. Giere, T. Iliescu, V. John, and D. Wells, *SUPG reduced order models for convection-dominated convection–diffusion–reaction equations*, Comput. Methods Appl. Mech. Engrg., 289 (2015), pp. 454–474.

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

[19] B. Haasdonk and M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, ESAIM M2AN, 42 (2008), pp. 277–302.

[20] J. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer, Berlin, 2016.

[21] T. Hughes and A. Brooks, *A multidimensional upwind scheme with no crosswind diffusion*, in Finite Element Methods for Convection Dominated Flows, AMD 34, American Society of Mechanical Engineers (ASME), New York, 1979, pp. 19–35.

[22] D. Huynh, G. Rozza, S. Sen, and A. 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.

[23] T. Iliescu, H. Liu, and X. Xie, *Regularized Reduced Order Models for a Stochastic Burgers Equation*, preprint, https://arxiv.org/abs/1701.01155, 2017.

[24] T. Iliescu and Z. Wang, *Variational multiscale proper orthogonal decomposition: Navier-Stokes equations*, Numer. Methods Partial Differential Equations, 30 (2014), pp. 641–663.

[25] F. Incropera and D. DeWitt, *Fundamentals of Heat and Mass Transfer*, John Wiley & Sons, New York, 1990.

[26] V. John and J. Novo, *Error analysis of the SUPG finite element discretization of evolutionary convection-diffusion-reaction equations*, SIAM J. Numer. Anal., 49 (2011), pp. 1149–1176, https://doi.org/10.1137/100789002.

[27] C. Johnson and U. Nävert, *An analysis of some finite element methods for advection-diffusion problems*, in Analytical and Numerical Approaches to Asymptotic Problems in Analysis (Proc. Conf., Univ. Nijmegen, Nijmegen, 1980), North-Holland Math. Stud. 47, North-Holland, Amsterdam, 1981, pp. 99–116.

[28] C. Johnson, U. Nävert, and J. Pitkäranta, *Finite element methods for linear hyperbolic problems*, Comput. Methods Appl. Mech. Engrg., 45 (1984), pp. 285–312.

[29] T. Lassila and G. Rozza, *Parametric free-form shape design with PDE models and reduced basis method*, Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 1583–1592.

[30] A. Logg, K. Mardal, and G. Wells, *Automated Solution of Differential Equations by the Finite Element Method*, Springer-Verlag, Berlin, 2012.

[31] S. Lorenzi, A. Cammi, L. Luzzi, and G. Rozza, *POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations*, Comput. Methods Appl. Mech. Engrg., 311 (2016), pp. 151–179.

[32] Y. Maday, A. Manzoni, and A. Quarteroni, *An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated problems*, C. R. Math. Acad. Sci. Paris, 354 (2016), pp. 1188–1194.

[33] A. Manzoni, A. Quarteroni, and G. Rozza, *Model reduction techniques for fast blood flow simulation in parametrized geometries*, Int. J. Numer. Methods Biomed. Engrg., 28 (2012), pp. 604–625.

[34] 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, Chichester, 2011, pp. 151–177.

[35] 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.

[36] P. Pacciarini, *Stabilized Reduced Basis Method for Parametrized Advection-Diffusion PDEs*, Master's thesis, Università degli Studi di Pavia, Pavia, Italy, 2012.

[37] P. Pacciarini and G. Rozza, *Stabilized reduced basis method for parametrized advection–diffusion PDEs*, Comput. Methods Appl. Mech. Engrg., 274 (2014), pp. 1–18.

[38] P. Pacciarini and G. Rozza, *Stabilized reduced basis method for parametrized scalar advection-diffusion problems at higher Péclet number: Roles of the boundary layers and inner fronts*, in Proceedings of the 11th World Congress on Computational Mechanics and 5th European Congress on Computational Mechanics and 6th European Congress on Computational Fluid Dynamics, Barcelona, Spain, 2014, pp. 5614–5624.

[39] P. Pacciarini and G. Rozza, *Reduced basis approximation of parametrized advection-diffusion PDEs with high Péclet number*, in Numerical Mathematics and Advanced Applications - ENUMATH 2013, Lecture Notes Comput. Sci. Eng. 103, Springer, Cham, 2015, pp. 419–426.

[40] A. Quarteroni, G. Rozza, and A. Manzoni, *Certified reduced basis approximation for parametrized partial differential equations and applications*, J. Math. Ind., 1 (2011), 3.

[41] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Texts Appl. Math. 37, Springer-Verlag, Berlin, 2007.

[42] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer, Berlin, 1994.

[43] G. Rozza, D. Huynh, and A. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics*, Arch. Comput. Methods Eng., 3 (2008), pp. 229–275.

[44] G. Rozza, C. N. Nguyen, A. T. Patera, and S. Deparis, *Reduced basis methods and a posteriori error estimators for heat transfer problems*, in Proceedings of the ASME 2009 Heat Transfer Summer Conference Collocated with the InterPACK09 and 3rd Energy Sustainability Conferences. Volume 2: Theory and Fundamental Research; Aerospace Heat Transfer; Gas Turbine Heat Transfer; Computational Heat Transfer, San Francisco, CA, 2009, pp. 753–762.

[45] C. Spannring, *Weighted Reduced Basis Methods for Parabolic PDEs with Random Input Data*, Ph.D. thesis, Graduate School CE, Technische Universität Darmstadt, Darmstadt, Germany, 2018.

[46] C. Spannring, S. Ullmann, and J. Lang, *A weighted reduced basis method for parabolic PDEs with random data*, in Recent Advances in Computational Engineering (ICCE 2017), Lecture Notes in Comput. Sci. Eng. 124, M. Schäfer et al., eds., Springer, Cham, pp. 145–161, https://doi.org/10.1007/978-3-319-93891-2_9.

[47] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza, *POD-Galerkin reduced order methods for CFD using finite volume discretisation: Vortex shedding around a circular cylinder*, Commun. Appl. Ind. Math., 8 (2017), pp. 210–236.

[48] L. Venturi, *Weighted Reduced Basis Methods for Parametrized PDEs in Uncertainty Quantification Problems*, Master's thesis, Università degli Studi di Trieste/SISSA, Trieste, Italy, 2016.

[49] L. Venturi, F. Ballarin, and G. Rozza, *A weighted POD method for elliptic PDEs with random inputs*, J. Sci. Comput., to appear, https://doi.org/10.1007/s10915-018-0830-7.