## Chapter 1

# Weighted Reduced Order Methods for Uncertainty Quantification

sec:chapter_13

*Davide Torlo*[1], *Maria Strazzullo*[1,4], *Francesco Ballarin*[2], *Gianlugi Rozza*[1]

## 1.1 ▪ Introduction

Partial differential equations (PDEs) represent an effective tool to model phenomena in applied sciences. Realistic problems usually depend on several physical and geometrical parameters that can be calibrated exploiting real data. In real scenarios, however, these parameters are affected by *uncertainty* due to measurement errors or scattered data information. To deal with more reliable models which take into account this issue, the numerical approximation of stochastic PDEs can be exploited. In the *Uncertainty Quantification* (UQ) context, many simulations are run to better understand the system at hand and to compute statistics of outcomes over quantities of interest. In particular, the input parameters of the stochastic PDEs are assumed to be random finite–dimensional variables.

Classical numerical approximations, i.e., the *high fidelity* solutions, can lead to unbearable computational costs to compute statistical momenta. These are the leading motivations of this contribution. Indeed, we will focus on projection-based reduction techniques which can lighten this issue: *reduced order methods* (ROMs) [8, 17]. In many applications, the parameterized solutions can be sought in a low-dimensional subset of the solution space. If one applies a Galerkin-projection in this *reduced space*, the problem is solved more rapidly with respect to the original discretization, still being accurate. These approaches, thus, might accelerate standard statistical analysis techniques, such as the Monte-Carlo methods (for alternative methods for forward UQ with random inputs, e.g. Stochastic Galerkin, Stochastic collocation and Karhunen–Loève approximation, we refer to [21]). More precisely, this Chapter focuses on *weighted* ROMs for forward uncertainty problems, where the reduced algorithms are modified to comply with some previous knowledge on the distribution of the parameters and to exploit this information to accelerate even more the reduced simulations, see e.g. [4, 24] and the references therein. Moreover, the proposed weighted ROMs are not intrusive with respect to the classical ROM approaches,

[1] mathLab, Mathematics Area, SISSA, Scuola Internazionale Superiore di Studi Avanzati, Trieste, Italy

[4] Department of Mathematical Sciences "G.L. Lagrange", Politecnico di Torino, Italy

[2] Department of Mathematics and Physics, Catholic University of the Sacred Heart, Brescia, Italy

i.e., given a reduced model the wROMs just weigh its outcome without changing its form. We refer to [13] for inverse problems applications and to [9] for polynomial chaos expansion reduced techniques.
The contribution is outlined as follows: in section 1.2 we will present stochastic PDEs and their approximation. Then, we will move toward weighted ROMs, describing standard algorithms such as Weighted Proper Orthogonal Decomposition [4] and Weighted Reduced Basis [24] and the related sampling strategies, in sections 1.3 and 1.4, respectively. Then, in section 1.5, we will validate the proposed algorithms in several contexts, from heat transfer, Stokes problems and advection dominated phenomena to optimal control for environmental sciences. Conclusions follow in section 1.6.

## 1.2 ▪ Stochastic Partial Differential Equations and Discretized Approximations

ch3:Stocastic_problem

In this section, we introduce the stochastic setting for PDEs in their high fidelity discretization. Consider a complete probability space $(\mathcal{A}, \mathcal{F}, P)$, with $\mathcal{A}$ the set of the possible outcomes, $\mathcal{F}$ is a $\sigma$–algebra of events and $P$ is a probability measure. Let $\mu = \mu(\omega)$, $\mu : (\mathcal{A}, \mathcal{F}) \to (\Gamma, \mathcal{B})$ where $\Gamma \subset \mathbb{R}^K$ is a compact set, $\mathcal{B}$ is the Borel measure and $\mu(\omega) = (\mu_1(\omega), \dots, \mu_K(\omega))$ is a random vector whose components are independent absolutely continuous random variables and parameterizes the physical problem. Denote with $\rho : \mathbb{R}^K \to \mathbb{R}$ the probability density function of $\mu$. Let $D \subset \mathbb{R}^d$ with $d = 1, 2, 3$ be a bounded physical domain. The physical problems we are interested in can be modeled by PDEs over $D$, as follows

$$\text{find } u : \Gamma \to \mathbb{V} \text{ such that } \mathcal{P}(u(\mu(\omega)); \mu(\omega)) = 0, \tag{1.1}$$

for almost every $\omega \in \mathcal{A}$. The solution $u$ is usually sought in a Sobolev space, that we generically define as $\mathbb{V}$. Here, $\mathcal{P}$ represent a parameterized PDE problem in its weak formulation, for example an elliptic problem or a parabolic initial–valued problem, Navier–Stokes equations or a shallow water problem.

We suppose, furthermore, that a high fidelity method, namely finite element or finite volume, provides an approximation $u_N$ of the solution $u$. These solutions are characterized by an error which is inversely proportional to a power of the number of degrees of freedom $N$ of the discretized functional space $\mathbb{V}_N$ and a computational cost which is proportional to a polynomial in $N$. Let us define the high fidelity solver as

$$\text{find } u_N : \Gamma \to \mathbb{V}_N \text{ such that } \mathcal{P}_N(u_N(\mu(\omega)); \mu(\omega)) = 0, \quad \text{for a.e. } \omega \in \mathcal{A}. \lceil^{\text{eq:FOM}} \tag{1.2}$$

The computation of the high fidelity solution can be unfeasible in real–time or many–query situations, where a faster but still reliable solver is desired. In particular, we focus in this chapter on stochastic problems, where statistical momenta are the objective of the study. Typically, averaging algorithms, as Monte Carlo, are used to obtain such quantities and hence the computation of many simulations for different parameters $\mu \in \Gamma$ are necessary.

To speed up these procedures, ROMs offer interesting strategies. The core idea of ROMs is to search the parameterized solution inside a reduced functional space $\mathbb{V}_n \subset \mathbb{V}_N$. This reduced space is characterized by a dimension $n \ll N$. Model order reduction algorithms consists of two phases: an *offline* phase, where the reduced space is built using some *snapshots*, i. e., high fidelity simulations for some parameters $\mu$, and an *online* phase where the reduced space structure is exploited and a reduced solver with computational costs proportional to $n$ is used to simulate the solution for new parameters $\mu$. The reduced problem will be denoted by

$$\text{find } u_n : \Gamma \to \mathbb{V}_n \text{ such that } \mathcal{P}_n(u_n(\mu(\omega)); \mu(\omega)) = 0, \quad \text{for a.e. } \omega \in \mathcal{A}. \lceil^{\text{eq:ROM}} \tag{1.3}$$

For the problem we deal with, the reduced space is built in order to control the error between the high–fidelity solutions and the reduced ones $\|u_N(\mu) - u_n(\mu)\|$, and a cheap and sharp error bound is available, i. e.,

$$\eta_n(u_n(\mu(\omega)); \mu(\omega)) \geq \|u_N(\mu(\omega)) - u_n(\mu(\omega))\|_{\mathbb{V}_N} =: e_{\mathbb{V}_n}(\mu(\omega)), \quad \text{for a.e. } \omega \in \mathcal{A}. \tag{1.4}$$

## 1.3 ▪ Weighted Reduced Order Methods

ch13:wROMs

In this chapter we will focus on how the *offline* phase of different algorithms can be modified to take into account the knowledge of the distribution of the random variable $\mu$. This is done mainly by the means of weighted algorithms that are derived from classical ROMs algorithms [8, 17]. We will present the weighted Reduced Basis (wRB) method [4] and the weighted Proper Orthogonal Decomposition (wPOD) [24]. Moreover, we will focus only on the *offline* modification of such algorithms, as the *online* phase stays unmodified with respect to the original algorithms. The difference is that now we are interested not only in one evaluation for a given $\mu$ but for the momenta of the random variable $u_n(\mu)$. We refer to [8, 16] for the development of such parts of the algorithms.

### 1.3.1 ▪ Weighted Reduced Basis Method

The wRB [4] is an extension of the (deterministic) Reduced Basis (RB) method [18]. In the deterministic setting, the construction of the reduced space $\mathbb{V}_n$ is done following the Greedy algorithm 1.1. It selects some of the high fidelity snapshots from a training set $\{u_N(\mu)\}_{\mu \in \Xi_t}$. The construction of the reduced space is progressive and at every iteration the dimension of the reduced space increases by adding the selected snapshot to the bases of the reduced space. The chosen snapshot is the one that is worst approximated by the RB space at each iteration (in a *greedy* fashion). This choice is quickly done through the use of the error bound, which allows not to compute the high fidelity solution for all the parameters $\mu \in \Xi_t$ but only the ROM ones.

**ALGORITHM 1.1. (Deterministic) Greedy Algorithm.**

algo:RB

```
Sample $\Xi_t \subset \Gamma$
Let $\mathbb{V}_0 = \emptyset$
Pick an arbitrary $\mu^1 \in \Xi_t$
$n = 0$
while err $>$ tol do
    $n := n + 1$
    Solve (1.2) for $\mu = \mu^n$ to get $u_N(\mu^n)$
    Update $\mathbb{V}_n := \mathbb{V}_{n-1} \bigoplus \text{span}\{u_N(\mu^n)\}$
    Find $\mu^{n+1} := \arg\max_{\mu \in \Xi_t} \eta_n(u_n(\mu^{n+1}); \mu^{n+1})$
    err $:= \eta_n(u_n(\mu^{n+1}); \mu^{n+1})$
end
```

In the stochastic context, we want to give different importance to different parameters $\mu$, i. e., different realizations, according to the underlying probability distribution. This must be done modifying the error measure that we used in the deterministic setting, introducing a different norm $\|\cdot\|_w$ that modifies the search of the worst represented snapshot in the greedy algorithm. In particular, the new norm is defined as

$$\|u(\mu)\|_w := w(\mu)\|u(\mu)\|_{\mathbb{V}_N}, \quad \forall u \in \mathbb{V}_N, \forall \mu \in \Gamma. \tag{1.5}$$

The weight function $w : \Gamma \to \mathbb{R}^+$ depends on the target norm that one wants to minimize. As an example, if we want to minimize the expected value of the 2-norm of the error, i. e.,

$$\mathbb{E}[\|u_N - u_n\|^2_{\mathbb{V}_N}] = \int_D \|u_N(\mu(\omega)) - u_n(\mu(\omega))\|^2_{\mathbb{V}_N} dP(\omega) \; {}^{\text{eq:wExpValueProb}} \tag{1.6}$$

$$= \int_\Gamma \|u_N(\mu) - u_n(\mu)\|^2_{\mathbb{V}_N} \rho(\mu) d\mu, \; {}^{\text{eq:wExpectedValue}} \tag{1.7}$$

the natural choice for the weight function is $w(\mu) = \sqrt{\rho(\mu)}$ [4]. Consequently, also the error estimator reflects this modification by introducing a new error bound $\eta_n^w(\mu) = w(\mu)\eta_n(\mu)$, resulting in

$$\mathbb{E}[\|u_N - u_n\|^2_{\mathbb{V}_N}] \leq \int_\Gamma \eta_n^w(\mu)^2 dy = \int_\Gamma \eta_n(\mu)^2 \rho(\mu) d\mu. \tag{1.8}$$

Other choices may be more meaningful in other contexts, e.g. different norms, output error minimization [22, 25]. The modified algorithm is reported in algorithm 1.2.

**ALGORITHM 1.2. Weighted Greedy Algorithm.**

```
algo:wRB
Properly sample $\Xi_t \subset \Gamma$
Let $\mathbb{V}_0 = \emptyset$
Pick an arbitrary $\mu^1 \in \Xi_t$
$n = 0$
while err $>$ tol do
    $n := n + 1$
    Solve (1.2) for $\mu = \mu^n$ to get $u_N(\mu^n)$
    Update $\mathbb{V}_n := \mathbb{V}_{n-1} \bigoplus \text{span}\{u_N(\mu^n)\}$
    Find $\mu^{n+1} := \arg\max_{\mu \in \Xi_t} \eta_n^w(u_n(\mu^{n+1}); \mu^{n+1})$
    err $= \eta_n^w(u_n(\mu^{n+1}); \mu^{n+1})$
end
```

Another important aspect in this algorithm is the choice of the sampling strategy of the parameter training set $\Xi_t$. In Section 1.4 we will investigate in more details the different possibilities and their effects on the overall results.

### 1.3.2 ▪ Weighted Proper Orthogonal Decomposition

The other well–known algorithm in the ROMs community is the Proper Orthogonal Decomposition (POD) [8]. Its weighted version (wPOD), originally introduced in [24], is useful when an error estimator $\eta_n$ is not available.

The POD algorithm 1.3 finds the global minimum of the mean square error

$$ {}^{\text{eq:squareErrorPOD}} \int_\Gamma \|u_N(\mu) - u_n(\mu)\|^2_{\mathbb{V}_N} d\mu \tag{1.9}$$

over all the possible reduced spaces $\mathbb{V}_n \subset \mathbb{V}_N$ of dimension $n$. In practice, a tolerance on the error can be set and the dimension $n$ will be provided by the algorithm. In the discrete situation (1.9) becomes

$$\frac{1}{|\Xi_t|} \sum_{\mu \in \Xi_t} \|u_N(\mu) - u_n(\mu)\|^2_{\mathbb{V}_N}, \tag{1.10}$$

with $\Xi_t$ being the training set. The reduced space $\mathbb{V}_n$ is defined by the eigenvectors corresponding to the $n$ leading eigenvalues of the operator $C : \mathbb{V}_N \to \mathbb{V}_N$ defined by

$$C(v) := \sum_{i=1}^{|\Xi_t|} \langle v, u_N(\mu^i)\rangle_{\mathbb{V}_N} u_N(\mu^i). \tag{1.11}$$

In practice, this is computed by an eigenvalue analysis on the correlation matrix $\hat{C}_{ij} := \langle u_N(\mu^i), u_N(\mu^j)\rangle_{\mathbb{V}_N}$. The bases of the reduced space $\mathbb{V}_n$ are defined by $\xi^i = \sum_{j=1}^{|\Xi_t|} \psi_j^i u_N(\mu^j)$, where $\psi_j$ are the eigenvectors of $\hat{C}$.

**ALGORITHM 1.3. POD.**

algo:POD

```
Sample $\Xi_t \subset \Gamma$
for $\mu \in \Xi_t$ do
    Solve (1.2) for $\mu$ to get $u_N(\mu)$
end
Assemble the matrix $\hat{C}_{ij} = \langle u_N(\mu^i), u_N(\mu^j)\rangle_{\mathbb{V}_N}$
Compute the biggest $n$ eigenvalues $\lambda_k$ and the eigenvectors $\psi_k$ of $\hat{C}$ for $k = 1, \dots, n$
Define $\mathbb{V}_n :=$ span$\{\xi^1, \dots, \xi^n\}$, being $\xi^i = \sum_{j=1}^{|\Xi_t|} \psi_j^i u_N(\mu^j)$.
```

The stochastic setting modifies the mean square error (1.9) as the expected value (1.7). This leads to a modified correlation matrix

eq:wCorrMatrix

$$\hat{C}^w_{ij} := w(\mu^i)\langle u_N(\mu^i), u_N(\mu^j)\rangle_{\mathbb{V}_N}. \tag{1.12}$$

We remark that $\hat{C}^w$ is not diagonalizable in the usual sense, but it is with respect to the scalar product induced by $C$, therefore this allows to obtain $n$ orthogonal leading eigenvectors.

**ALGORITHM 1.4. wPOD.**

algo:wPOD

```
*Properly* sample $\Xi_t \subset \Gamma$
for $\mu \in \Xi_t$ do
    Solve (1.2) for $\mu$ to get $u_N(\mu)$
end
Assemble the matrix $\hat{C}^w$ as in (1.12)
Compute the biggest $n$ eigenvalues $\lambda_k$ and the eigenvectors $\psi_k$ of $\hat{C}^w$ for $k = 1, \dots, n$
Define $\mathbb{V}_n :=$ span$\{\xi^1, \dots, \xi^n\}$, being $\xi^i = \sum_{j=1}^{|\Xi_t|} \psi_j^i u_N(\mu^j)$.
```

## 1.4 ▪ Sampling Strategies

sec:sampling

In the previously proposed algorithms, the choice of the training set $\Xi_t$ over which the minimization occurs is fundamental. There are different strategies that one could consider. In the stochastic setting it is very common to use Monte Carlo algorithms to approximate the momenta of interest. In the numerical analysis setting, it can be seen as a collection of quadrature points for a specific quadrature formula. These two perspectives lead to different methods that one can use, with different pro and cons:

- Monte Carlo uniform sampling to approximate (1.7);
- Monte Carlo with sampling lead by the underlying distribution of the random variable $\mu$, approximating directly (1.6);

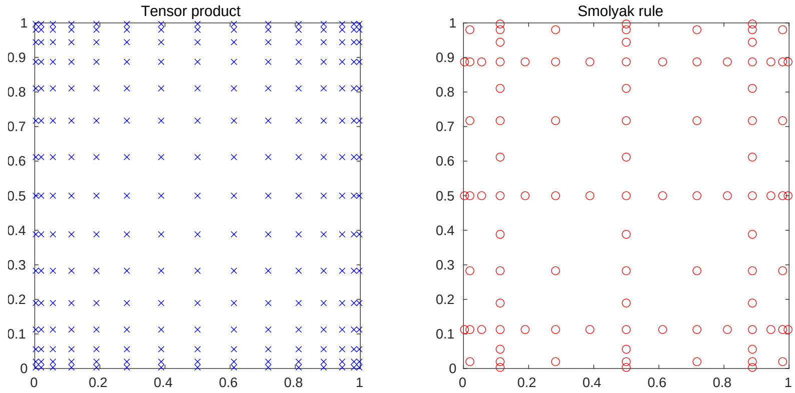

**Figure 1.1.** *Comparison of multivariate quadrature rules with order of accuracy 15.* fig:Smolyak

- Tensor product quadrature rules of univariate quadrature rules;
- Sparse Smolyak quadrature rule.

The Monte Carlo rules, very common in the stochastic world, suffer of a slow decay of the error of the integral, which scales as an $\mathcal{O}(|\Xi_t|^{-1/2})$. The sampling using the underlying distribution may lead to some advantages, in the case of very concentrated distributions, but it is strongly not recommended for rare events prediction.

The classical quadrature rules that generalize with tensor products in more dimension suffer of the curse of the dimensionality. In particular, according to the smoothness of the function we want to approximate, they will behave as $\mathcal{O}(|\Xi_t|^{-\bar{n}/K})$, where $\bar{n}$ is the minimum between the regularity of the function and the accuracy degree of the quadrature rule, and $K$ is the dimension of the (parameter) space $\Gamma$.

Sparse quadrature rules can improve this by putting the quadrature points in different locations. The Smolyak quadrature rule [10, 2, 26] allows to not fall into the curse of dimensionality, using different refinement levels of the grids on the different parameters. Consider $\Gamma := \Pi_{j=1}^K \Gamma_j$ and define by $U_i^{(j)}$ the univariate quadrature rule at the refinement level $i$ on the interval $\Gamma_j \subset \mathbb{R}$, introducing the differences opeators on $\Gamma_j$ as

$$\Delta_0^{(j)} = 0, \qquad \Delta_i^{(j)} := U_{i+1}^{(j)} - U_i^{(j)} \text{ for } i \geq 0, \tag{1.13}$$

the Smolyak quadrature rule of order $q$ in $\Gamma$ is defined as

eq:SmolyakQuadrature

$$Q_q^K := \sum_{|\alpha|_1 \leq q,\, \alpha \in \mathbb{N}^K} \bigotimes_{i=1}^K \Delta_{\alpha_i}^{(i)}. \tag{1.14}$$

The difference could be observed in Figure 1.1, where a 2 dimensional example shows the tensor product and the Smolyak rule.

In the next section, we will alternate different quadrature strategies to show the advantages of choosing samples distributed with the underlying probability law and the cost reduction without loss of quality of sparse quadratures.

## 1.5 ▪ Applications

ch13:applications

In this section we apply the strategies presented above to different problems. In a first section we applied directly the presented strategies on a heat equation problem and on Stokes problems.

In Section 1.5.2 we will exploit the proposed methods for a selective stabilization technique in the context of uncertainty quantification. Finally, in Section 1.5.3 we will apply the weighted algorithms to parameterized optimal control problems in environmental sciences.

### 1.5.1 ▪ Averaging Uncertain Parameter Simulations for Heat Equation and Stokes Problems

The first problem we consider is a heat transfer problem on a squared domain $D = [0,1]^2$. The equation that we solve is

$$-\mu(x_1,x_2)\Delta u(x_1,x_2) = 1 \tag{1.15}$$

where $\mu : [0,1]^2 \to \mathbb{R}$ is defined piecewise constant on 9 subsquares defined by $D_{ij} = [i/3,(i+1)/3] \times [j/3,(j+1)/3]$, $i,j = 0,\dots,2$ by 9 parameters $\mu_{ij} \in \mathbb{R}$ with $i,j = 0,\dots,2$. In particular, we consider these heat transfer coefficients to behave like some Beta(20,10) random variables, i.e., $\mu_{ij} \sim$ Beta(20,10) rescaled on the interval $[1,10]$. The Beta distribution is defined through the Beta function $B(\alpha,\beta) = \int_0^1 \mu^{\alpha-1}(1-\mu)^{\beta-1} d\mu$ with the probability density function

$$\rho(\mu) = \frac{\mu^{\alpha-1}(1-\mu)^{\beta-1}}{B(\alpha,\beta)}. \tag{1.16}$$

We compare the reduced algorithm presented before in their weighted and classical formulations and we compare the Uniform Monte Carlo (MC) sampling strategy with the Beta(20,10) Monte Carlo sampling one, in both cases with $|\Xi_t| = 100$.

To assess the quality of the algorithms we compute reduced solutions onto a Beta$(20,10)$ distributed test set and we average the error obtained with such solutions.

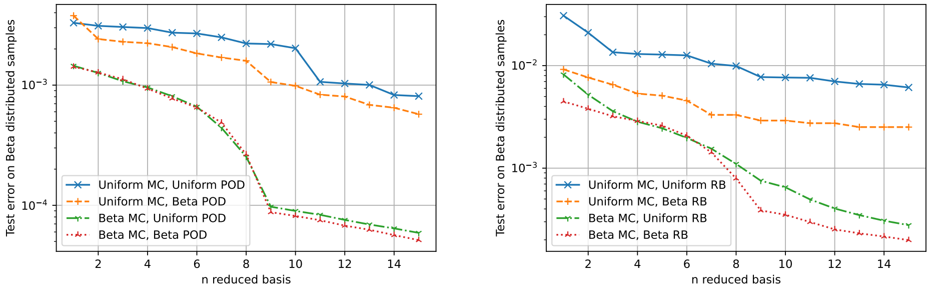

**Figure 1.2.** *Heat transfer problem: error decay with respect to the number of reduced basis functions, comparison between weighted and not weighted algorithms and between uniform and Beta(20,10) distributed training samples. Left POD, right Greedy algorithm.* `fig:ROMUQWeightHeatTransferErrorDecay`

In figure 1.2, we notice that for the POD the weighting of the algorithm does not lead to great improvement in the decay of the error, while sampling of the training set with the underlying distribution extremely reduces the error, by even a factor 10. Conversely, for the greedy algorithm we see already a great improvement with only the weighted algorithm of a factor 5. Also here the sampling strategy plays a bigger role in the training phase and gives an overall improvement of at least a factor 10. The combination of the 2 strategies slightly improves this result. We remark that a uniform sample considers parameters which are not at all well represented in the reduced space, hence, this leads to worse average errors.

We now focus on the stochastic steady Stokes equations results shown in [7]: here we extend the deterministic problem presented in [6] following the strategies first discussed in [5]. In this case, we will deal with physical and geometrical parameterization. Namely, we consider the random domain $D(\mu(\omega)) : \mathcal{A} \to \mathbb{R}^2$ and a random inlet condition $\mathbf{g}_{in}(\mathbf{x}, \mu(\omega)) : \partial D_{in} \times \mathcal{A} \to \mathbb{R}^2$. From now on, for the sake of notation, we will use $\mu$ or $\mu(\omega)$ indistinctly. The random boundary of the domain is given by the following partition

$$\partial D(\mu) = \partial D_{D,\mathbf{0}}(\mu) \cup \partial D_{in}(\mu) \cup \partial D_N(\mu),$$

where $\partial D_{D,\mathbf{0}}(\mu)$ and $\partial D_{in}(\mu)$ represent the portions of the boundary with homogeneous and non-homogeneous Dirichlet conditions, respectively, while $\partial D_N(\mu)$ is characterized by Neumann boundary conditions. The stochastic Stokes problem reads: given an outcome $\omega \in \mathcal{A}$, find the $\mu-$dependent pair $(\mathbf{u}, p) : D \times \mathcal{A} \to \mathbb{R}^2 \times \mathbb{R}$ such that

$$\begin{cases} -\Delta \mathbf{u}(\mathbf{x}) + \nabla p(\mathbf{x}) = 0 & \text{in } D(\mu), \\ \nabla \cdot \mathbf{u}(\mathbf{x}) = 0 & \text{in } D(\mu), \\ \mathbf{u}(\mathbf{x}) = \mathbf{0} & \text{on } \partial D_{D,\mathbf{0}}(\mu), \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}_{in}(\mathbf{x}, \mu) & \text{on } \partial D_{in}(\mu), \\ \nu \dfrac{\partial \mathbf{u}}{\partial \mathbf{n}}(\mathbf{x}) - p(\mathbf{x})\mathbf{n} = 0 & \text{on } \partial D_N(\mu). \end{cases} \tag{1.17}$$

The parameter we consider is

$$\mu = (\mu_0, \mu_1, \mu_2, \mu_3, \mu_4) \in (0.2, 1.9) \times (0.2, 2.) \times (0.2, 2.) \times (0.2, 2.) \times (0.2, 20.),$$

where the first four parameters are related to the shape of the considered computational domain, while the last one is related to the diffusivity of the flow since it changes the inlet condition: indeed, we define $\mathbf{g}_{in}(\mathbf{x}, \mu) = (-\mu_4 x_1(x_1 - 3), 0)$.

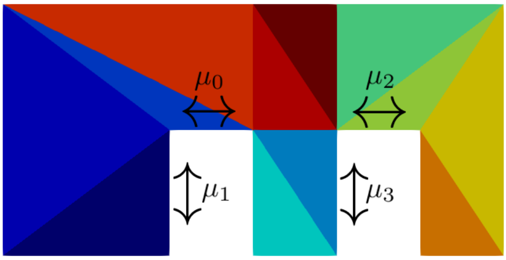

**Figure 1.3.** *Reference domain D. The reference parameter is* $\mu = (1., 1.5, 1., 1.5, \mu_4)$. fig:domain_julien

The domain is given by

$$D(\mu) = \{[0, 6] \times [0, 3]\} \setminus \{([2, 2 + \mu_0] \times [0, \mu_1]) \cup ([3 + \mu_0, 3 + \mu_0 + \mu_2] \times [0, \mu_3])\}.$$

The boundary portions are $\partial D_N(\mu) = \{6\} \times [0, 3]$ and $\partial D_{in}(\mu) = \{0\} \times [0, 3]$, while $\partial D_{D,\mathbf{0}}(\mu) = \partial D(\mu) \setminus \{\partial D_N(\mu) \cup \partial D_{in}(\mu)\}$. In order to deal with geometrical parameterization, we need to consider a *reference domain*, say $D$, that we represent in Figure 1.3. At each new parameter evaluation the system (1.17) is traced back to $D$, see e.g. [17]. To build the reduced model we applied wPOD with *supremizer enrichment* [1, 19], which guarantees the well-posedness of the Stokes problem at the reduced level for each new realization of $\mu$. Furthermore, we assume that the parameters verify $\mu_i \sim$ Beta(75,75), for $i = 0, \dots, 4$. The testing set has cardinality

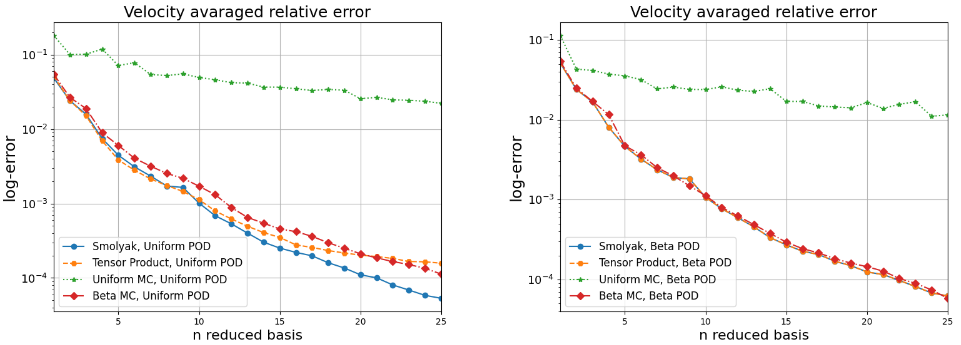

**Figure 1.4.** *Stokes problem: error decay with respect to the number of reduced basis functions, comparison between not weighted (left) and weighted (right) algorithms and between Smolyak, tensor product, uniform, Beta(75,75) distributed training samples. For the weighted algorithm, Smolyak rule, tensor product rule and Beta(75,75) sampling coincide.* fig:CH13StokesErrorDecay

$|\Xi_t| = 241$ when dealing with the Smolyak rule, while $|\Xi_t| = 243$ for all the other test cases. Increasing the value $n$, we compare several sampling techniques: Smolyak rule, tensor product rule, uniform and Beta(75, 75) sampling over a testing set of 100 outcomes from a Beta(75, 75) sampling. In figure 1.4, we show results for the not weighted algorithm (left plot) and the weighted one (right plot). For the sake of brevity we report only the velocity averaged log-errors since the pressure has a comparable behavior. First of all, we notice that, apart from the uniform sampling, we reach very good results in term of relative errors with values around $10^{-4}$ for both the algorithms. Focusing on the not weighted algorithm, we see how the sampling strategies might strongly help in accuracy with respect to uniform MC sampling. Indeed, uniform MC suffers from the curse of dimensionality. As already seen in the left plot of figure 1.2, the wPOD performs slightly better for the uniform sampling. We remark that the two approaches are not equivalent as the resulting reduced bases are different. For the other sampling strategies, the weighted algorithm gives comparable results with respect to the not weighted.

### 1.5.2 ▪ Stabilization of Advection Dominated Problems Conditioned to Parameters

sec:stabAdvectionDominatedUQ

In this section we will make use of the weighted and distributed algorithms presented above to further reduce computational costs in the online phase.

Let us consider the Graetz problem, an advection-diffusion problem

$$10^{\mu} x_2(1 - x_2)\partial_x u - \Delta u = 0, \tag{1.18}$$

on the domain $D = [0, 2] \times [0, 1]$ with Dirichlet boundary condition equal to 1 on the boundaries with $x_1 \leq 1$ and equal to 0 on the boundaries with $x_1 > 1$. We suppose that the parameter $\mu$ is distributed as a Beta(5,3) rescaled on the interval $[0, 6]$. The considered problem will be advection dominated as $\mu$ increases and a standard finite element method would fail in providing an accurate solution, as it would be lead by oscillations due to the instability of the problem. In order to stabilize the method, we introduce in the offline phase a stabilization term, represented by streamline upwind Petrov-Galerkin (SUPG) [15], which keeps the consistency of the method, but adds a stabilization effect.

In this simple example the stabilization is not computationally expensive, but in more realistic cases the stabilization term might include more elaborated nonlinear terms.

In the online phase we can choose whether we want to apply again the stabilization or not. As for the high fidelity model, it is necessary to use it in the advection dominated regime. We see this behavior running an online error decay study with respect to the dimension of $n$ in figure 1.5. There, it is clear that, both with POD and Greedy without online stabilization, the error has a plateau at $10^{-1}$ which is the distance between the non stabilized solutions and the stabilized ones [22].

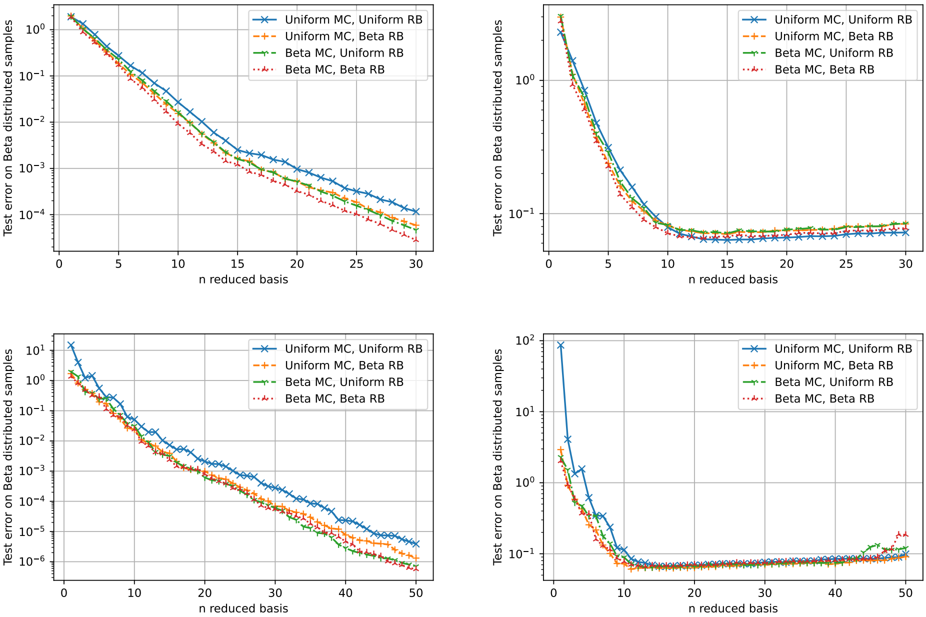

**Figure 1.5.** *Graetz problem: error decay with respect to the number of reduced basis functions, comparison between weighted and not weighted algorithms and between uniform and Beta(5,3) distributed training samples. With online stabilization (left), without online stabilization (right). POD algorithm (top), greedy algorithm (bottom).* fig:ROMUQWeightAdvectionErrorDecay

In figure 1.5 for stabilized methods, we observe the same order of convergence but smaller errors for weighted algorithms and distributed training samples. Hence, for the rest of this section we consider only this type of algorithms.

We can observe in figure 1.6 that the unstabilized and stabilized online solutions show differences in the error in particular in the advection dominated regime, while, in the diffusive regime, the two solutions show similar errors. Hence, one may be more interested in solving these solutions without the stabilization terms to save computational time.

The task that we aim to solve in this section is to compute some statistical quantities of the stochastic solution $u(\mu)$, for example the average. A classical strategy could be to approximate the average via Monte Carlo algorithm, i.e.,

$$\mathbb{E}(u(\mu)) = \int_{\mathcal{A}} u(\mu(\omega))dP(\omega) = \int_{\mathcal{A}} u(\mu(\omega))\rho(\mu(\omega))d\omega \approx \sum_{i=1}^{N_{MC}} \frac{u(\mu^i)\rho(\mu^i)}{N_{MC}}, \qquad \text{eq:averageMC} \quad (1.19)$$

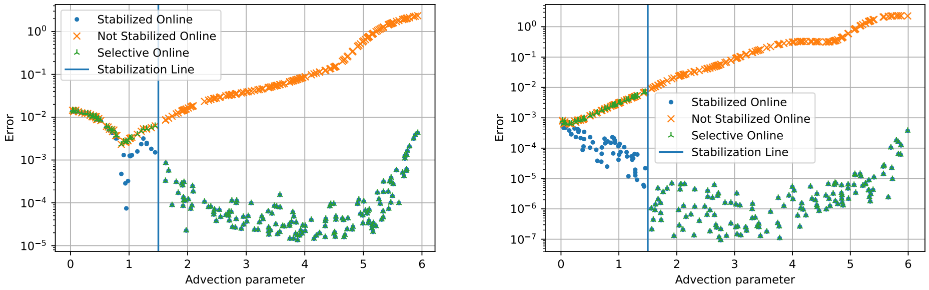

**Figure 1.6.** *Graetz problem: weighted algorithms and Beta(5,3) distributed training samples. Error of stabilized and not stabilized online with respect to the advection parameter $\mu$. Comparison with selective online strategy. Left POD, right Greedy.* fig:ROMErrorWrtParameter

POD algorithm

| $\varepsilon$ | Mean Error | Stabilized Rate |
|---|---|---|
| 0.0 | 5.43329 $10^{-5}$ | 100 % |
| 1.0 | 5.62446 $10^{-5}$ | 83% |
| 1.5 | 6.79605 $10^{-5}$ | 78% |
| 2.0 | 3.61034 $10^{-4}$ | 71% |
| 2.5 | 1.46119 $10^{-3}$ | 65% |
| 3.0 | 4.17433 $10^{-3}$ | 56.5% |
| 6.0 | 2.42411 $10^{-1}$ | 0.0% |

Greedy algorithm

| $\varepsilon$ | Mean Error | Stabilized Rate |
|---|---|---|
| 0.0 | 2.55161 $10^{-6}$ | 100 % |
| 1.0 | 8.02889 $10^{-6}$ | 85.5% |
| 1.5 | 8.069458 $10^{-5}$ | 75% |
| 2.0 | 5.031474 $10^{-4}$ | 68 % |
| 2.5 | 3.192200 $10^{-3}$ | 60 % |
| 3.0 | 1.066145 $10^{-2}$ | 52.5 % |
| 6.0 | 3.217842 $10^{-1}$ | 0% |

**Table 1.1.** *Tables for the selective stabilization approach, given a certain threshold advection coefficient $\varepsilon$, we obtain the mean error in table computing only a percentage of all the stabilization terms.* tab:advectionUQselectiveStab

where $\rho$ is the probability density function of the underlying distribution of $\mu$, and $\mu^i$ are Monte Carlo samples uniformly distributed in the domain of interest.

Computing many high fidelity simulations $u_N(\mu^i)$ is too expensive for this task, so we consider the reduced solutions instead $u_n(\mu^i)$. Now, the stabilization term is important to guarantee an accurate solution in the advection dominated regime, i.e., $\mu \gg 0$. Conversely, in the diffusion dominated regime, the stabilization term is not necessary and would be time consuming.

What we present here is a selective approach on the reduced solutions, which will be computed with the stabilization term or not according to a rule lead by a threshold value $\varepsilon$. Namely, while computing (1.19), for every $u^i$ we will use $u_n(\mu^i)$ with the stabilization term if $\mu^i > \varepsilon$, else we will use $u_n(\mu^i)$ without the stabilization term. This idea is illustrated in figure 1.6 by the selective online plot.

In table 1.1 we see the result of this approach changing different values of the threshold $\varepsilon$. In the POD case we see that for an $\varepsilon = 1.5$ we save the computational costs of the 22% of the stabilization terms and we obtain a negligible worsening of the error. Already with $\varepsilon = 2$ the error is a bit larger though increasing to almost the 30% of stabilizations not computed. Higher thresholds make less sense. For the Greedy approach we observe similar results, even if the error increases a bit more even with small thresholds.

### 1.5.3 ▪ Weighted POD for Optimal Control for Environmental Sciences

sec:optimizationUQWeighted

This section focuses on the application of wROMs to parameterized Optimal Control Problems (OCP($\mu$)s) applied to environmental sciences. Namely, we extend the results of [20] in a stochastic setting, as presented in [3]. UQ combined with OCP($\mu$) is of utmost interest in the field of natural sciences, that are strongly related to *in situ* data. Furthermore, there is the need of reliable simulations based on PDEs which are similar to the collected measurements. OCP($\mu$)s comply with this task: they aim at changing the classical behavior of a system towards a desired configuration, namely, the collected information. This scope is reached through the employment of external factors called *controls*. The control, thus, acts on the system and guides it in order to lower as much as possible a cost functional dependent on the data observations of the physical phenomenon. The ingredients of an OCP($\mu$) are

- a state variable $y \in \mathbb{Y}$,
- a control variable $u \in \mathbb{U}$,
- an observation $y_d \in \mathbb{Z} \supset \mathbb{Y}$,

for $\mathbb{Y}, \mathbb{U}$ and $\mathbb{Z}$ Hilbert spaces. The problem we are dealing with is the minimization over $\mathbb{Y}$ and $\mathbb{U}$ of

eq:ocpfunctional

$$\mathcal{J}(y, u; \mu) := \frac{1}{2}\|y - y_d\|_{\mathbb{Z}}^2 + \frac{\alpha}{2}\|u\|_{\mathbb{U}}^2, \tag{1.20}$$

constrained to

eq:ocp_state

$$\mathcal{P}(y(\mu), u(\mu); \mu) = 0, \tag{1.21}$$

where $\mathcal{P}$ is a linear stochastic constraint in weak formulation. The interested reader may refer to [23] for a survey on optimal control theory. This problem formulation has been employed in data assimilation contexts, where the variable $u(\mu)$ steers the system towards the desired configuration $y_d$ making the solution $y(u(\mu))$ the most similar to the observation. The stochastic parameter still plays an important role since both the data and the system may be affected by uncertainty. In the following, we report some results concerning this framework applied to an hypothetical loss of pollutant in the Gulf of Trieste. Let us indicate this geographical region with $D \subset \mathbb{R}^2$, represented in figure 1.7. We call $D_u$ the subdomain where the pollutant loss is happening (green subdomain in figure 1.7). The goal is to determine the maximum loss allowed in $D_u$ to keep harmless the pollutant concentration in $D_{obs}$, the Miramare natural reserve (orange subdomain in figure 1.7). The boundary is partitioned in $\partial D_D$, the coastline, and $\partial D_N$, the open sea, where Dirichlet and Neumann boundary conditions have been applied, respectively. The state concentration is $y \in \mathbb{Y} := H^1_{\partial D_D}$, $u \in \mathbb{R}$ and $y_d = 0.2$ over $D_{obs}$. The observation represents a safe threshold for the pollutant concentration in the Miramare area.

For this specific test case, $\mathcal{P}$ is an advection-diffusion equation of the following form:

$$\int_\Omega \mu_1 \nabla y \cdot \nabla z dD + \int_\Omega \left([\mu_2, \mu_3] \cdot \nabla y\right) z dD - Lu \int_{\Omega_u} z dD_u = 0 \quad \forall z \in \mathbb{Y}. \tag{1.22}$$

Here, the random parameter $\mu = (\mu_1, \mu_2, \mu_3) \in (0.5, 1) \times (-1, 1) \times (-1, 1)$ models the sea dynamics under several meteorological phenomena. The constant $L = 1000$ is used to make the system non-dimensional [20]. Furthermore, we impose $\alpha = 10^{-7}$. We want to compare the performances of informed sampling with respect to standard POD. We will show how exploiting data information in such a context might help in monitoring and solve in real time potentially dangerous situation. Thus, we exploit the not weighted version of the POD with aggregated space strategy [11, 12, 14], since, as already said in the aforementioned applications, the POD

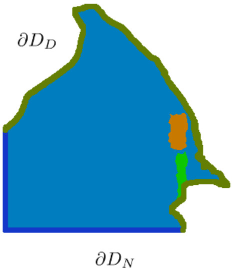

**Figure 1.7.** *Domain $D$, the Gulf of Trieste. Orange: Miramare reserve $D_{obs}$. Pollutant spill $D$* fig:gulf13

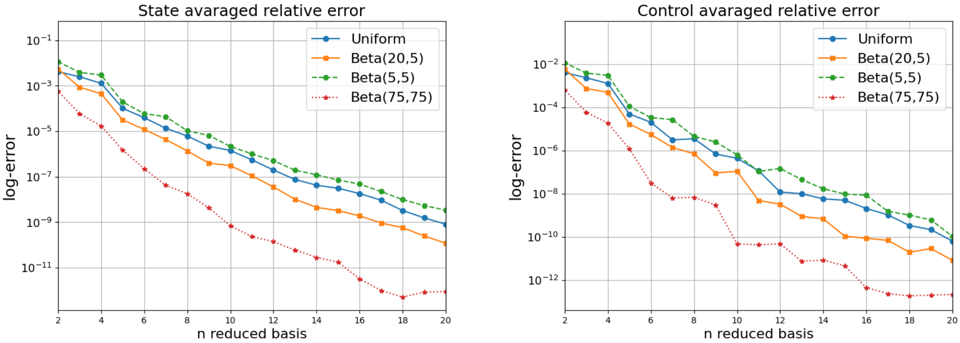

**Figure 1.8.** *Optimal control problem: error decay with respect to the number of reduced basis functions, comparison between state variable (left) and control variable (right) with a not weighted POD algorithm between uniform, Beta(20, 5), Beta(5, 5) and Beta(75,75) distributed training samples.* fig:CH13OCPErrorDecay

is slightly affected by the weights, while the sampling strategy is of great importance. In this experiment $|\Xi_t| = 100$. In figure 1.8 we present the average relative error for the state and the control variables where the sampling for the test coincides with the one of the offline phase. This assures good performances for all the samplings. However, we would like to underline that adding distribution information can be crucial when dealing with random variables. Indeed, thanks to some previous knowledge about parameter distribution, one can employ a much lower number of bases. This is due to the nature of the algorithm that focuses on a very specific parameter setting both in creating the reduced spaces and in assessing their accuracy. It is the case of $\mu \sim$ Beta(75,75) (rescaled on the respective parameter ranges), where $n = 4$ is sufficient to reach errors values around $10^{-6}$ for both the variables, gaining two orders of magnitude with respect to uniform sampling. Less bases translate in a gain of computational time that can be exploited in UQ analysis even for very complicated problems such as OCPs that are usually characterized by high computational costs due to the minimization framework.

## 1.6 ▪ Conclusions

ch13:conclusions

This chapter focuses on weighted ROMs for a broad class of stochastic PDEs-based models. First we introduce stochastic PDEs with random inputs and their standard discretizations. Thus, we introduced ROM strategies in order to deal with them in a faster way, following the POD and the RB fashion. The related weighted approaches are described too. Furthermore, we highlighted the role of several sampling strategies based on different (possibly sparse) quadrature rules. We validated this setting in many contexts, such as heat transfer equation, Stokes problems, stabilization of advection dominated problems and optimal control for environmental sciences. The results show the better performances of weighted approaches and distributed quadratures with respect to standard algorithm based on a deterministic viewpoint.

# Bibliography

[1] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza, *Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations*, International Journal for Numerical Methods in Engineering, 102 (2014), pp. 1136–1161, https://doi.org/10.1002/nme.4772. (Cited on p. 8)

[2] V. Bathelmann, E. Novak, and K. Ritter, *High dimensional polynomial interpolation on sparse grids*, Advances in Computational Mathematics, 4 (2000), pp. 273–288, https://doi.org/10.1016/j.cma.2007.07.016. (Cited on p. 6)

[3] G. Carere, M. Strazzullo, F. Ballarin, G. Rozza, and R. Stevenson, *A weighted POD-reduction approach for parametrized PDE-constrained optimal control problems with random inputs and applications to environmental sciences*, Computers & Mathematics with Applications, 102 (2021), pp. 261–276, https://doi.org/10.1016/j.camwa.2021.10.020. (Cited on p. 12)

[4] P. Chen, A. Quarteroni, and G. Rozza, *A Weighted Reduced Basis Method for Elliptic Partial Differential Equations with Random Input Data*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 3163–3185, https://doi.org/10.1137/130905253. (Cited on pp. 1, 2, 3, 4)

[5] P. Chen, A. Quarteroni, and G. Rozza, *Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by stokes equations*, Numerische Mathematik, 133 (2016), pp. 67–102, https://doi.org/10.1007/s00211-015-0743-4. (Cited on p. 8)

[6] G. Galdi, *An introduction to the mathematical theory of the Navier-Stokes equations: Steady-state problems*, Springer Science & Business Media, 2011. (Cited on p. 8)

[7] J. Genovese, F. Ballarin, G. Rozza, and C. Canuto, *Weighted reduced order methods for uncertainty quantification in computational fluid dynamics.*, In preparation, (2022). (Cited on p. 8)

[8] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer Briefs in Mathematics, Springer, Switzerland, 1 ed., 2015, https://doi.org/10.1007/978-3-319-22470-1. (Cited on pp. 1, 3, 4)

[9] S. Hijazi, G. Stabile, A. Mola, and G. Rozza, in QUIET Selected Contributions, M. D'Elia, M. Gunzburger, and G. Rozza, eds., vol. 137 of Lecture Notes in Computational Science and Engineering, Springer International Publishing, 2020, ch. Non-intrusive Polynomial Chaos Method Applied to Full-Order and Reduced Problems in Computational Fluid Dynamics: A Comparison and Perspectives, pp. 217–240, https://doi.org/10.1007/978-3-030-48721-8_10. (Cited on p. 2)

[10] M. Holtz, *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*, vol. 77, Springer, 2010. (Cited on p. 6)

[11] M. Kärcher and M. A. Grepl, *A certified reduced basis method for parametrized elliptic optimal control problems*, ESAIM: Control, Optimisation and Calculus of Variations, 20 (2014), pp. 416–441, https://doi.org/10.1051/cocv/2013069. (Cited on p. 12)

[12] M. Kärcher, Z. Tokoutsi, M. A. Grepl, and K. Veroy, *Certified reduced basis methods for parametrized elliptic optimal control problems with distributed controls*, Journal of Scientific Computing, 75 (2018), pp. 276–307, https://doi.org/10.1007/s10915-017-0539-z. (Cited on p. 12)

[13] U. E. Morelli, P. Barral, P. Quintela, G. Rozza, and G. Stabile, *A numerical approach for heat flux estimation in thin slabs continuous casting molds using data assimilation*, International Journal for Numerical Methods in Engineering, 122 (2021), pp. 4541–4574, https://doi.org/10.1002/nme.6713. (Cited on p. 2)

[14] F. Negri, G. Rozza, A. Manzoni, and A. Quarteroni, *Reduced basis method for parametrized elliptic optimal control problems*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2316–A2340, https://doi.org/10.1137/120894737. (Cited on p. 12)

[15] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, vol. 23, Springer Berlin Heidelberg, 1994, https://doi.org/10.1007/978-3-540-85268-1. (Cited on p. 9)

[16] G. Rozza, D. Huynh, N. Nguyen, and A. Patera, *Real-time reliable simulation of heat transfer phenomena*, in ASME -American Society of Mechanical Engineers - Heat Transfer Summer Conference Proceedings, S. Francisco, CA, USA, 2009, https://doi.org/10.1115/HT2009-88212. (Cited on p. 3)

[17] G. Rozza, D. Huynh, and A. T. Patera, *Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations*, Archives of Computational Methods in Engineering, 15 (2008), pp. 229–275, https://doi.org/10.1007/s11831-008-9019-9. (Cited on pp. 1, 3, 8)

[18] G. Rozza, N. Nguyen, A. Patera, and S. Deparis, *Reduced basis methods and a posteriori error estimators for heat transfer problems*, in ASME -American Society of Mechanical Engineers - Heat Transfer Summer Conference Proceedings, S. Francisco, CA, USA, 2009, https://doi.org/10.1115/HT2009-88211. (Cited on p. 3)

[19] G. Rozza and K. Veroy, *On the stability of the reduced basis method for Stokes equations in parametrized domains*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 1244–1260, https://doi.org/10.1016/j.cma.2006.09.005. (Cited on p. 8)

[20] M. Strazzullo, F. Ballarin, R. Mosetti, and G. Rozza, *Model Reduction for Parametrized Optimal Control Problems in Environmental Marine Sciences and Engineering*, SIAM Journal on Scientific Computing, 40 (2018), pp. B1055–B1079, https://doi.org/10.1137/17M1150591. (Cited on p. 12)

[21] T. Sullivan, *Introduction to Uncertainty Quantification*, Springer, 2015, https://doi.org/10.1007/978-3-319-23395-6. (Cited on p. 1)

[22] D. Torlo, F. Ballarin, and G. Rozza, *Stabilized Weighted Reduced Basis Methods for Parametrized Advection Dominated Problems with Random Inputs*, SIAM/ASA Journal on Uncertainty Quantification, 6 (2018), pp. 1475–1502, https://doi.org/10.1137/17M1163517. (Cited on pp. 4, 10)

[23] F. Tröltzsch, *Optimal control of partial differential equations*, Graduate studies in mathematics, 112 (2010), https://doi.org/10.1090/gsm/112. (Cited on p. 12)

[24] L. Venturi, F. Ballarin, and G. Rozza, *A Weighted POD Method for Elliptic PDEs with Random Inputs*, Journal of Scientific Computing, 81 (2019), pp. 136–153, https://doi.org/10.1007/s10915-018-0830-7. (Cited on pp. 1, 2, 3, 4)

[25] L. VENTURI, D. TORLO, F. BALLARIN, AND G. ROZZA, *Weighted Reduced Order Methods for Parametrized Partial Differential Equations with Random Inputs*, Springer International Publishing, Cham, 2019, pp. 27–40, `https://doi.org/10.1007/978-3-030-04870-9_2`. (Cited on p. 4)

[26] G. W. WASILKOWSKI, *Explicit cost bounds of algorithms for multivariate tensor product problems*, Journal of Complexity, 1 (1995), pp. 1–56, `https://doi.org/10.1006/jcom.1995.1001`. (Cited on p. 6)