



SCUOLA INTERNAZIONALE SUPERIORE DI STUDI AVANZATI

SISSA Digital Library

Weighted Reduced Order Methods for Parametrized Partial Differential Equations with Random Inputs

*Original*

Weighted Reduced Order Methods for Parametrized Partial Differential Equations with Random Inputs / Venturi, Luca; Torlo, Davide; Ballarin, Francesco; Rozza, Gianluigi. - (2019), pp. 27-40. [10.1007/978-3-030-04870-9\_2]

*Availability:*

This version is available at: 20.500.11767/87854 since: 2022-11-16T09:19:16Z

*Publisher:*

Springer

*Published*

DOI:10.1007/978-3-030-04870-9\_2

*Terms of use:*

Testo definito dall'ateneo relativo alle clausole di concessione d'uso

*Publisher copyright*

Springer

This version is available for education and non-commercial purposes.

note finali coverpage

(Article begins on next page)

# Weighted reduced order methods for parametrized partial differential equations with random inputs

Luca Venturi, Davide Torlo, Francesco Ballarin, and Gianluigi Rozza

**Abstract** In this manuscript we discuss weighted reduced order methods for stochastic partial differential equations. Random inputs (such as forcing terms, equation coefficients, boundary conditions) are considered as parameters of the equations. We take advantage of the resulting parametrized formulation to propose an efficient reduced order model; we also profit by the underlying stochastic assumption in the definition of suitable weights to drive to reduction process. Two viable strategies are discussed, namely the weighted reduced basis method and the weighted proper orthogonal decomposition method. A numerical example on a parametrized elasticity problem is shown.

## 1 Introduction

Several problems in applied sciences and engineering can be modeled by partial differential equations (PDEs). Complex models may be characterized by several coefficients, which could possibly be calibrated from experiments or measurements; as measurements of such quantities are often affected, in practice, by noise and uncertainty, one is usually interested in solving *parametrized* PDEs, in which the coefficients are allowed to vary under a supposed probability distribution. As no analytical solution is usually available, numerical approximation by discretization methods (e.g. finite element) are customarily employed. However,

---

Luca Venturi  
mathLab, Mathematics Area, SISSA, Trieste, Italy  
Current address: Courant Institute of Mathematical Sciences, New York University, U.S.  
e-mail: venturi@cims.nyu.edu

Davide Torlo  
mathLab, Mathematics Area, SISSA, Trieste, Italy  
Current address: Institut für Mathematik. Universität Zürich, Switzerland  
e-mail: davide.torlo@math.uzh.ch

Francesco Ballarin  
mathLab, Mathematics Area, SISSA, Trieste, Italy  
e-mail: francesco.ballarin@sissa.it

Gianluigi Rozza  
mathLab, Mathematics Area, SISSA, Trieste, Italy  
e-mail: gianluigi.rozza@sissa.it

it is often unaffordable (in terms of CPU times) to run a new simulation for each parametric scenario, especially when interested in the evaluation of some statistics of the solution. For this reason, several groups in the computational science and engineering community advocate the use of reduced order models.

Model reduction comes in different forms, often depending on the specific needs and applications. For instance, a dimensionality reduction of the physical space can be carried out to speed up computations; e.g. in fluid dynamics, engineers may occasionally be willing to replace a detailed model of a three-dimensional network of tubes by a simplified one-dimensional network, if local three-dimensional features are not relevant for the phenomenon at hand. The simplification can be carried out at the level of mathematical model as well, such as a potential flow description in fluid dynamics. This work will focus, instead, on *projection-based* reduction techniques, which are based on the idea that parametrized solutions often belong to a lower dimensional manifold of the solution space. In such a case, few basis functions (obtained from few solutions of the detailed physical problem) should be enough to span the lower dimensional manifold.

Two widely used techniques for basis functions generation are going to be discussed in this work, namely the reduced basis method [8, 17, 18, 16] and the proper orthogonal decomposition approach [8, 11]. While originally proposed for deterministic problems, such methods have been extended as well to a stochastic framework by properly weighting the reduced basis construction phase according to the underlying probability distribution [3, 4, 5]. The resulting reduced order methods (ROMs) have been named *weighted reduced basis* method [3, 4, 5, 21, 20, 19] and *weighted proper orthogonal decomposition* [22]. The main goal of this chapter is to summarize and compare these two weighted approaches by means of a numerical test case. We do mention that further ROM techniques to tackle uncertainty are available, e.g. [24, 7, 10, 13], but a more extensive comparison is beyond the scope of this work.

The outline of the chapter is as follows: the mathematical model, governed by a parametrized PDE, is outlined in Section 2, as well as its finite element discretization. Two weighted ROMs are proposed in Section 3, which extend deterministic reduced basis and proper orthogonal decomposition methods. Numerical results on a elasticity problem test case are provided in Section 4. Conclusions follow in Section 5.

## 2 Parametrized formulation and high fidelity approximation

Let  $(\Omega, \mathcal{F}, P)$  be a complete probability space, where  $\Omega$  denotes the set of possible outcomes,  $\mathcal{F}$  is the  $\sigma$ -algebra of events and  $P$  is the probability measure. Let  $y = y(\omega)$ ,  $y : (\Omega, \mathcal{F}) \rightarrow (\Gamma, \mathcal{B})$ , where  $\Gamma \subset \mathbb{R}^K$  is a compact set,  $\mathcal{B}$  is the Borel measure, and  $y(\omega) = (y_1(\omega), \dots, y_K(\omega))$  is a random vector which components are independent absolutely continuous random variables. Denote by  $\rho : \mathbb{R}^K \rightarrow \mathbb{R}$  the probability density function of  $y$ . In the following, we will identify the realization of  $y$  as the parameter.

Furthermore, let  $D \subset \mathbb{R}^d$ ,  $d = 1, 2, 3$ , be a bounded domain. We assume that the engineering problem at hand is modeled by a parametrized elliptic partial differential over  $D$ , as follows:

$$\begin{aligned} &\text{find } u : \Gamma \rightarrow \mathbb{V} \text{ such that} \\ &a(u(y(\omega)), v; y(\omega)) = F(v; y(\omega)) \quad \forall v \in \mathbb{V}, \end{aligned}$$

for a.e.  $\omega \in \Omega$ .

In the following we will assume  $\mathbb{V}$  to be a subset of  $H^1(D)$ ; we will also assume that the bilinear form  $a(\cdot, \cdot; y) : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$  is coercive and continuous, as well as the linear functional  $F(\cdot; y) \in \mathbb{V}'$ , in order to guarantee the well-posedness of the problem [15]. Specific expressions for  $a(\cdot, \cdot; y)$  and  $F(\cdot; y)$  will be provided in the numerical experiments in Section 4.

In general, due to the lack of analytical solutions, we will resort to a numerical approximation based on a Galerkin approach. Given a finite dimensional approximation space  $\mathbb{V}_\delta \subseteq \mathbb{V}$  (in the following, a finite element space [15]), characterized by  $\dim(\mathbb{V}_\delta) = N_\delta < \infty$ , we consider the approximate problem:

$$\begin{aligned} &\text{find } u_\delta : \Gamma \rightarrow \mathbb{V}_\delta \text{ such that} \\ &a(u_\delta(y(\boldsymbol{\omega})), v; y(\boldsymbol{\omega})) = F(v; y(\boldsymbol{\omega})) \quad \text{for all } v \in \mathbb{V}_\delta, \end{aligned} \quad (1)$$

for a.e.  $\boldsymbol{\omega} \in \Omega$ . We refer to problem (1) as the *high fidelity* (or *truth*) problem and to  $u_\delta$  as the *high fidelity solution*.

We are interested in the computation of statistics of the solution, e.g. its expectation  $\mathbb{E}[u]$ , or statistics of some output of interest  $s : \mathbb{V} \rightarrow \mathbb{R}$ , e.g.  $\mathbb{E}[s(u)]$ , by means of Monte-Carlo method. Since the repeated evaluation of (1) for different realizations  $y \in \Gamma$  is usually computationally expensive for moderately large  $N_\delta$  (i.e., high accuracy of  $u_\delta(y) \simeq u(y)$ ), we aim at replacing the *high fidelity* solution  $u_\delta(y)$  with a cheaper, yet accurate, approximation. This approximation will be the result of a *reduced order method* in which a Galerkin project is carried out over a further low dimensional space  $\mathbb{V}_N$  (rather than  $\mathbb{V}_\delta$ ).

### 3 Weighted reduced order methods

In this Section we introduce the *weighted* reduced order methods (ROMs), which extends deterministic ROMs by means of proper weighting associated to the underlying probability space. As usual in the ROM literature, we ensure an offline-online computational strategy; during the offline stage the reduced basis space  $\mathbb{V}_N$  is constructed once and for all, while during the online phase (to be repeated for any possible realization of  $y$ ) a Galerkin projection over  $\mathbb{V}_N$  is carried out to find the *reduced order solution*. For the sake of brevity, in this Section we will only discuss in detail the offline phase, as it is the one in which a proper weighting is needed in order to account for stochastic formulation of the problem. The online phase, instead, can be carried out as in the deterministic setting; we refer to [8, 18, 16] for further details.

#### 3.1 Weighted Reduced Basis method

Let us introduce first the weighted Reduced Basis method, originally introduced in [3, 4, 5] as an extension of the (deterministic) Reduced Basis (RB) method [8].

In the deterministic setting, the construction of the reduced space  $\mathbb{V}_N$  is based on the so-called Greedy algorithm, summarized in Algorithm 1(a). The aim of the Greedy algorithm is to build a reduced basis space  $\mathbb{V}_N$  spanned by some of the *truth* snapshots  $\{u_\delta(y^i)\}_{i=1}^{n_t}$ , where  $\mathcal{E}_t = \{y^i\}_{i=1}^{n_t} \subset \Gamma$  is a training set of parameters (line 1). After an initialization to setup the empty reduced space (line 2) and the first value  $y^1$  (line 3), the algorithm proceeds iteratively (lines 4-11). At each iteration  $N$ , the solution corresponding to the current parameter  $y^N$  is computed (line 5) and used to enrich the reduced space (line 6). The parameter  $y^{N+1}$  to be used at the next iteration is then automatically chosen (in a greedy fashion) as the worst approximated solution on the whole set of parameters  $\mathcal{E}_t$  (line 7). The simplest implementation of such selection criterion would require the computation of the error  $e_N(y) = \|u_\delta(y) - u_N(y)\|_{\mathbb{V}}$  for each  $y \in \mathcal{E}_t$ , and the corresponding computation of

$$\arg \max_{y \in \mathcal{E}_t} \|u_\delta(y) - u_N(y)\|_{\mathbb{V}}. \quad (2)$$

**Algorithm 1** Greedy algorithms**Algorithm 1(a)** (Deterministic) Greedy Algorithm

**Input:** parameter domain  $\Gamma$ , tolerance  $\varepsilon^{tol}$  and  $N_{max}$ .  
**Output:** reduced space  $\mathbb{V}_N$ .

- 1: Sample  $\Xi_t \subset \Gamma$ ;
- 2: Define  $\mathbb{V}_0 = \emptyset$ ;
- 3: Pick arbitrary  $y^1 \in \Xi_t$ ;
- 4: **for**  $N = 1, \dots, N_{max}$  **do**
- 5:   Solve (1) for  $y = y^N$  to compute  $u(y^N)$ ;
- 6:   Update  $\mathbb{V}_N = \mathbb{V}_{N-1} \oplus \text{span}\{u_\delta(y^N)\}$ ;
- 7:   Compute  $y^{N+1} = \arg \max_{y \in \Xi_t} \eta_N(y)$ ;
- 8:   **if**  $\eta_N(y^{N+1}) \leq \varepsilon^{tol}$  **then**
- 9:     **break**
- 10:   **end if**
- 11: **end for**

**Algorithm 1(b)** Weighted Greedy Algorithm

**Input:** parameter domain  $\Gamma$ , tolerance  $\varepsilon^{tol}$  and  $N_{max}$ .  
**Output:** reduced space  $\mathbb{V}_N$ .

- 1: *Properly* sample  $\Xi_t \subset \Gamma$ ;
- 2: Define  $\mathbb{V}_0 = \emptyset$ ;
- 3: Pick arbitrary  $y^1 \in \Xi_t$ ;
- 4: **for**  $N = 1, \dots, N_{max}$  **do**
- 5:   Solve (1) for  $y = y^N$  to compute  $u(y^N)$ ;
- 6:   Update  $\mathbb{V}_N = \mathbb{V}_{N-1} \oplus \text{span}\{u_\delta(y^N)\}$ ;
- 7:   Compute  $y^{N+1} = \arg \max_{y \in \Xi_t} \eta_N^w(y)$ ;
- 8:   **if**  $\eta_N^w(y^{N+1}) \leq \varepsilon^{tol}$  **then**
- 9:     **break**
- 10:   **end if**
- 11: **end for**

A more efficient alternative is to use an error estimator  $\eta_N(y)$  that bounds from above the error, i.e. such that

$$e_N(y) \leq \eta_N(y),$$

and that does not require the evaluation of the *truth* solution  $u_\delta(y)$ . This is readily possible for parametrized elliptic problems as in Section 2, under mild assumptions [8]. Thus, the computation of

$$\arg \max_{y \in \Xi_t} \eta_N(y)$$

is used in place of (2).

In a stochastic context, we want to give different importance to different parameters (i.e., different realizations), reflecting the probability distribution. This can be easily pursued by using a different norm  $\|\cdot\|_w$  in (2), i.e. by optimizing

$$\arg \max_{y \in \Xi_t} \|u_\delta(y) - u_N(y)\|_w, \quad (3)$$

where

$$\|u(y)\|_w = w(y) \|u(y)\|_{\mathbb{V}} \quad \forall u \in \mathbb{V}_\delta, \forall y \in \Gamma, \quad (4)$$

and  $w : \Gamma \rightarrow \mathbb{R}^+$  is a suitable weight function. The choice of the weight function  $w(y)$  may be driven by the desire of minimizing the expectation of the square norm error of the RB approximation, i.e.

$$\begin{aligned} \mathbb{E}[\|u_\delta - u_N\|_{\mathbb{V}}^2] &= \int_{\Omega} \|u_\delta(y(\omega)) - u_N(y(\omega))\|_{\mathbb{V}}^2 dP(\omega) = \\ &= \int_{\Gamma} \|u_\delta(y) - u_N(y)\|_{\mathbb{V}}^2 \rho(y) dy, \end{aligned} \quad (5)$$

that we can bound with

$$\mathbb{E}[\|u_\delta - u_N\|_{\mathbb{V}}^2] \leq \int_{\Gamma} \eta_N(y)^2 \rho(y) dy. \quad (6)$$

This suggests to define the weight function as  $w(y) = \sqrt{\rho(y)}$  as proposed in [4].

Other choices of the weight function  $w$  are possible. For instance, let us suppose that we are interested in the accurate approximation (on average) of a quantity  $s : \mathbb{V} \rightarrow \mathbb{R}$ , which depends on the solution of the

parametric PDE  $u$ , namely in minimizing

$$\mathbb{E}[|s(u_\delta) - s(u_N)|] = \int_{\Gamma} |s(u_\delta(y)) - s(u_N(y))| \rho(y) dy. \quad (7)$$

In this case, it is natural to choose as weight function  $w(y) = \rho(y)$ .

Regardless of the choice of  $w$ , the first modification of the weighted approach lies in using the weighted error estimator

$$\eta_N^w(y) = w(y) \eta_N(y) \quad (8)$$

on line 7 of Algorithm 1(b).

Furthermore, the choice of training set  $\Xi_t$  is usually crucial in the stochastic context (line 1 of Algorithm 1(b)). Two choices that we will compare in the numerical examples are (1) sampling from the uniform distribution  $\mathcal{U}(\Gamma)$ , as it is usually done in the deterministic case, and (2) sampling from the distribution  $\rho$ .

### 3.2 Weighted Proper Orthogonal Decomposition

Let us introduce next the weighted POD method, originally introduced in [22] as an extension of the deterministic POD–Galerkin approach [8]. For instance, the weighted POD may be used when an error estimator  $\eta_N(y)$  is not readily available for the weighted RB approach.

---

#### Algorithm 2 POD algorithms

---

##### Algorithm 2(a) (Deterministic) POD algorithm

- Input:** parameter domain  $\Gamma$ , tolerance  $\varepsilon^{tol}$  and  $N_{max}$ .  
**Output:** reduced space  $\mathbb{V}_N$ .
- 1: Sample  $\Xi_t \subset \Gamma$ ;
  - 2: **for each**  $y \in \Xi_t$  **do**
  - 3:   Solve (1) to compute  $u(y)$ ;
  - 4: **end for**
  - 5: Assemble the matrix  $\hat{C}_{ij} = \langle \varphi_i, \varphi_j \rangle_{\mathbb{V}}$ ;
  - 6: Compute its eigenvalues  $\lambda_k$ , and corresponding eigenvectors  $\psi_k$ ,  $k = 1, \dots, n_t$ ;
  - 7: Find the minimum  $N \in \{1, \dots, N_{max}\}$  such that  $E_N > 1 - \varepsilon^{tol}$ ;
  - 8: Define  $\mathbb{V}_N = \text{span}\{\xi^1, \dots, \xi^N\}$ , being  $\xi^i = \sum_{j=1}^{n_t} \psi_j^i \varphi_j$ .

##### Algorithm 2(b) Weighted POD algorithm

- Input:** parameter domain  $\Gamma$ , tolerance  $\varepsilon^{tol}$  and  $N_{max}$ .  
**Output:** reduced space  $\mathbb{V}_N$ .
- 1: *Properly* sample  $\Xi_t \subset \Gamma$ ;
  - 2: **for each**  $y \in \Xi_t$  **do**
  - 3:   Solve (1) to compute  $u(y)$ ;
  - 4: **end for**
  - 5: Assemble the *weighted* matrix  $\hat{C}_{ij}^w = w_i \langle \varphi_i, \varphi_j \rangle_{\mathbb{V}}$ ;
  - 6: Compute its eigenvalues  $\lambda_k$ , and corresponding eigenvectors  $\psi_k$ ,  $k = 1, \dots, n_t$ ;
  - 7: Find the minimum  $N \in \{1, \dots, N_{max}\}$  such that  $E_N > 1 - \varepsilon^{tol}$ ;
  - 8: Define  $\mathbb{V}_N = \text{span}\{\xi^1, \dots, \xi^N\}$ , being  $\xi^i = \sum_{j=1}^{n_t} \psi_j^i \varphi_j$ .
- 

In its deterministic version, the POD method aims to minimize the mean square error:

$$\int_{\Gamma} \|u(y) - u_N(y)\|_{\mathbb{V}}^2 dy. \quad (9)$$

over all possible reduced spaces of dimension  $N$ . From a practical point of view, a training set  $\Xi_t$  of cardinality  $n_t$  is introduced (line 1 of Algorithm 2(a)), and high fidelity solutions are computed for each  $y \in \Xi_t$  (lines 2-4). This enables to consider a discretized version of (9):

$$\frac{1}{n_t} \sum_{y \in \mathcal{E}_t} \|u(y) - u_N(y)\|_{\mathbb{V}}^2 \quad (10)$$

where  $\varphi_i = u(y^i)$ ,  $i = 1, \dots, n_t = |\mathcal{E}_t|$ . The reduced order subspace  $\mathbb{V}_N$  is therefore defined as the space spanned by the  $N$  leading eigenvectors of the operator [8]

$$C : v \in \mathbb{V} \rightarrow \sum_{i=1}^{n_t} \langle v, \varphi_i \rangle_{\mathbb{V}} \cdot \varphi_i.$$

In practice, a basis of  $\mathbb{V}_N$  can be found by computing the leading eigenvectors of the matrix  $\hat{C} = (\langle \varphi_i, \varphi_j \rangle_{\mathbb{V}})_{1 \leq i, j \leq n_t}$  (lines 5-6). Denoting by  $(\lambda_k, \psi_k)$  the eigenvalue-eigenvector pairs, the optimal subspace  $\mathbb{V}_N$  is obtained as

$$\mathbb{V}_N = \text{span}\{\xi^1, \dots, \xi^N\},$$

being  $\xi^i = \sum_{j=1}^{n_t} \psi_j^i \varphi_j$  (line 8). Practical stopping criterion involves either a maximum reduced space dimension  $N_{max}$ , or the selection of the minimum dimension  $N$  such that the so called retained energy

$$E_N = \frac{\sum_{k=1}^N \lambda_k}{\sum_{k=1}^{n_t} \lambda_k}$$

is greater than a prescribed tolerance (line 7).

To accommodate a stochastic setting, we observe that (10) involves a uniform Monte-Carlo approximation of (9). In the stochastic case, one would like to find the reduced order sub-space  $\mathbb{V}_N$  which minimizes the mean square error

$$\mathbb{E}[\|u - u_N\|_{\mathbb{V}}^2] = \int_{\Gamma} \|u(y(\omega)) - u_N(y(\omega))\|_{\mathbb{V}}^2 dP(\omega). \quad (11)$$

Nonetheless, any quadrature rule can be used for that purpose. In this framework, let  $\mathcal{E}_t \subset \Gamma$  and  $w_i$ ,  $i = 1, \dots, n_t$  be, respectively, the set of nodes and weights of a chosen quadrature rule:

$$\int_{\Gamma} \|u(y(\omega)) - u_N(y(\omega))\|_{\mathbb{V}}^2 dP(\omega) \simeq \sum_{i=1}^{n_t} w_i \|u(y_i) - u_N(y_i)\|_{\mathbb{V}}^2. \quad (12)$$

Computationally this turns out to require the evaluation of the eigenvalues of a weighted (preconditioned) matrix

$$C^w = (\omega_i \langle \varphi_i, \varphi_j \rangle_{\mathbb{V}})_{1 \leq i, j \leq n_t} = WC \quad \text{for} \quad W = \text{diag}(w_1, \dots, w_{n_t})$$

We observe that the matrix  $C^w$  is not diagonalizable in the usual sense, but with respect to the scalar product induced by  $C$ ; therefore this allows to recover  $N$  orthogonal leading eigenvectors.

We note that, once a quadrature rule has been selected, the weighted POD method prescribes both the weights (employed in line 5 of Algorithm 2(b)) and the training set (line 1). This is a difference to the weighted RB method, which only prescribes weights without an associated training set. Concerning the choice of the quadrature rule, the most straightforward way to do this is to use a Monte-Carlo approximation of (11), i.e. to choose  $\mathcal{E}_t$  sampled from the distribution  $\rho$  and  $w_i \equiv 1$ . For higher dimensional parameter spaces, a simple option is to perform a tensor product of a one-dimensional quadrature rules; however, in order to not be affected by the curse of dimensionality (which would entail very large training sets), we can instead adopt a Smolyak-type sparse quadrature rule [2, 9, 23, 6, 14] to significantly reduce the computational load (see e.g. Fig. 1)

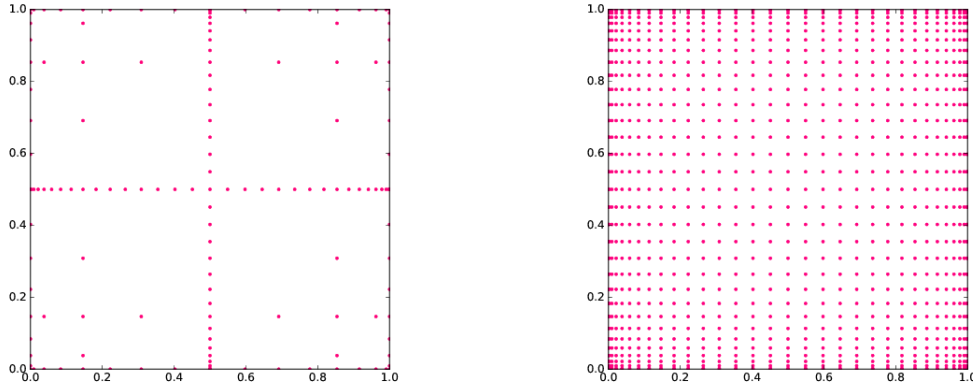


Fig. 1: Two dimensional grids based on nested Clenshaw-Curtis nodes of order  $q = 6$ . The left one is based on a Smolyak rule (145 nodes), while the right one on a full tensor product rule (1089 nodes).

## 4 Numerical results

In this Section we apply weighted reduced order methods to a linear elasticity problem.

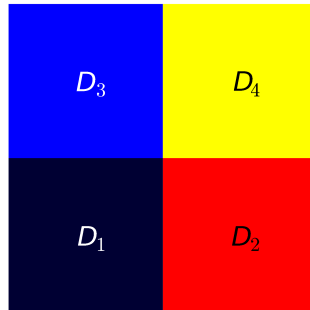


Fig. 2: Geometrical set-up of linear elasticity test case.

Let  $D_1, \dots, D_4$  be a partition of the domain  $D = [0, 1]^2$ , as in Fig. 2. We consider a linear elasticity problems characterized by six parameters, as follows:

$$a(\mathbf{u}, \mathbf{v}; y^1, \dots, y^4) = \sum_{i=1}^4 y^i \int_{D_i} \{ \lambda (\nabla \cdot \mathbf{u})(\nabla \cdot \mathbf{v}) + 2\mu \mathbf{e}(\mathbf{u}) : \mathbf{e}(\mathbf{v}) \} dx,$$

$$f(\mathbf{v}; y^5, y^6) = \sum_{i=1}^2 y^{i+4} \int_{(i-1)/2}^{i/2} v_2(1, x_2) dx_2,$$

for every  $\mathbf{u}, \mathbf{v} \in \mathbb{V}$ , where  $\mathbb{V} = \{ \mathbf{v} \in H^1(D; \mathbb{R}^2) : \mathbf{v}|_{[0,1] \times \{0,1\}} = 0 \}$ ,  $\nabla \cdot \mathbf{u}$  denotes the divergence of the displacement  $\mathbf{u}$ , and  $\mathbf{e}(\mathbf{u})$  is the symmetric part of the gradient of  $\mathbf{u}$ . This corresponds to consider the equation



of a linear elastic block, split in four parts, with Lamé constants rescaled by a parameter  $y^i$ ,  $i = 1, \dots, 4$ . The unscaled Lamé constants  $\lambda$  and  $\mu$  are obtained for Young modulus equal to 1 and Poisson ratio set to 0.3. Boundary conditions correspond to later tractions equal to  $y^5$  ( $y^6$ ) on the bottom (top) half of the right side, and clamped on bottom and top sides.

The random variables  $y^i$ ,  $i = 1, \dots, 6$ , are such that

$$\begin{aligned} \frac{y^i - 1}{2} &\sim \text{Beta}(\alpha, \beta), & \text{for } i = 1, \dots, 4, \\ \frac{y^i - 2}{4} &\sim \text{Beta}(\alpha, \beta), & \text{for } i = 5, 6. \end{aligned}$$

We will consider two cases, namely  $(\alpha, \beta) = (10, 10)$  and  $(\alpha, \beta) = (75, 75)$ , the latter resulting in a more peaked probability distribution.

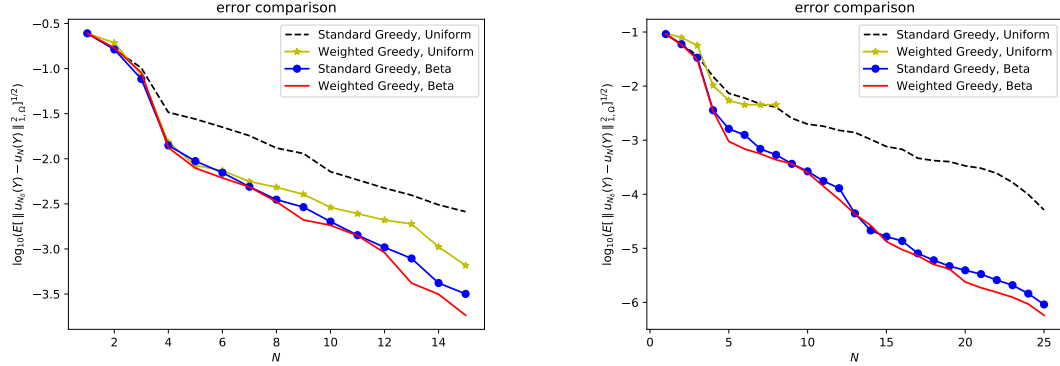
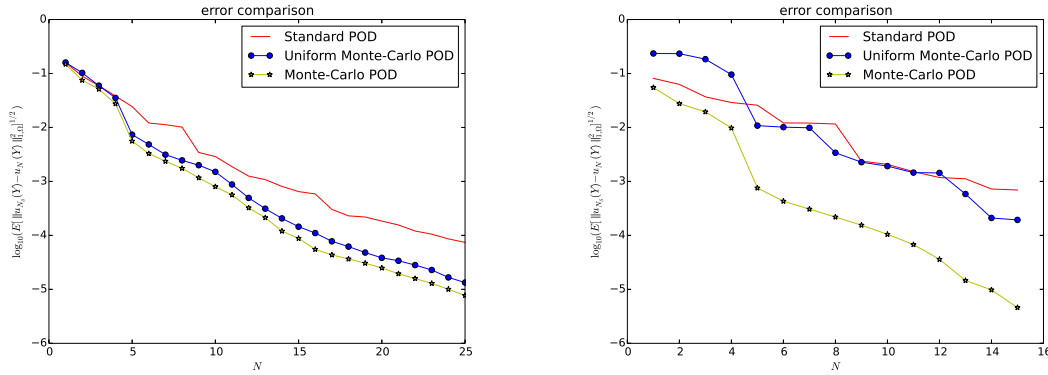


Fig. 3: Error analysis for standard and weighted RB algorithms, employing either a Uniform or Beta sampling for  $\mathcal{E}_T$ , for the cases  $(\alpha, \beta) = (10, 10)$  (left) and  $(\alpha, \beta) = (75, 75)$  (right).

Let us first comment the results of the weighted RB approach, summarized in Fig. 3, in which we plot the error  $\mathbb{E}[\|u - u_N\|_{\mathbb{V}}^2]$  for increasing reduced basis dimension  $N$ . We consider four different cases, which concern both the weighting (standard Greedy vs weighted Greedy) and the sampling (Uniform sampling vs sampling from the underlying Beta distribution) used during the Offline stage. Fig. 3(left), obtained for  $(\alpha, \beta) = (10, 10)$ , shows that, with an Uniform sampling, a proper weighting allows to improve the accuracy of the reduced order model (compare the dashed black line to the solid gold one). Further improvement can be obtained by sampling from the underlying Beta distribution (solid blue and solid red lines). The best accuracy is obtained if both the correct weighting and the appropriate sampling are employed (solid red line), resulting in an improvement of more than one order of magnitude compared to the standard Greedy algorithm employed in the deterministic setting. Such an improvement is even more prominent in Fig. 3(right), which corresponds to  $(\alpha, \beta) = (75, 75)$ ; in this case, the best weighted approach outperforms the deterministic algorithm by two orders of magnitude. The importance of a proper sampling is even more remarkable in this case, as the weighted Greedy with Uniform sampling suffered a breakdown after very few iterations due to the improper exploration of the parameter space, which led to a singular reduced order system.

	Standard	Uniform Monte-Carlo	Monte-Carlo	Gauss-Legendre	Sparse Gauss-Jacobi
$ \mathcal{E}_T $	500	500	500	729	389

Table 1: Cardinality of the training set  $\mathcal{E}_T$  employed in the various POD algorithms.Fig. 4: Error analysis for standard, Uniform Monte-Carlo and Monte-Carlo POD algorithms, for the cases  $(\alpha, \beta) = (10, 10)$  (left) and  $(\alpha, \beta) = (75, 75)$  (right).

We next carry out a similar task for different POD algorithms. Fig. 4 reports a comparison between standard POD (i.e. sampling from Uniform distribution, no weighting), Uniform Monte-Carlo POD (i.e. sampling from a Uniform distribution, and weighting according to the probability density function  $\rho$ ) and Monte-Carlo POD (i.e. sampling from the underlying Beta distribution, without any weighting). In contrast to the previous comparison for RB methods, we note that the fourth case (i.e. sampling from Beta and weighting according to  $\rho$ ) is omitted, as it does not admit an interpretation in terms of the quadrature formula (12). The cardinality of the training set (i.e., the number of quadrature points) is set to 500 for all three cases (see Table 1). The results in Fig. 4 show that Monte-Carlo POD guarantees the best results, with improvements of one (left) and two (right) orders of magnitude, respectively, with respect to the standard (deterministic) POD in terms of accuracy. Results for  $(\alpha, \beta) = (75, 75)$  emphasize the importance of a correct sampling (and thus, correct weighting as induced by (12)) as well: indeed, the Uniform Monte-Carlo POD provides marginal improvements over the standard POD in the case of highly concentrated distribution.

Owing to the interpretation of (12) as a quadrature rule, we provide in Fig. 5-6 a comparison to two further possible POD choices, based on tensor Gauss-Legendre and sparse Gauss-Jacobi quadratures, respectively. The cardinality of the training set in the Gauss-Legendre case is 729, while the sparse Gauss-Jacobi entails 389 training points (resulting from the sparsification of a tensor Gauss-Jacobi rule of 729 nodes), as summarized in Table 1. The former value was chosen as the minimum cardinality larger than 500 associated to a tensor Gauss-Legendre quadrature formula, in order to provide a fair comparison to the Monte-Carlo cases discussed previously. Fig. 5 shows indeed similar improvements as in Fig. 4 for what concerns the comparison between the standard POD and the weighted approaches. As in the weighted RB method, an inaccurate exploration of the parameter space caused the Gauss-Legendre POD to generate singular reduced order systems for  $(\alpha, \beta) = (75, 75)$  for  $N \geq 10$ . Finally, Fig. 6 depicts a similar comparison between Monte-Carlo and sparse Gauss-Jacobi POD. The latter should be preferred because, even though the two weighted

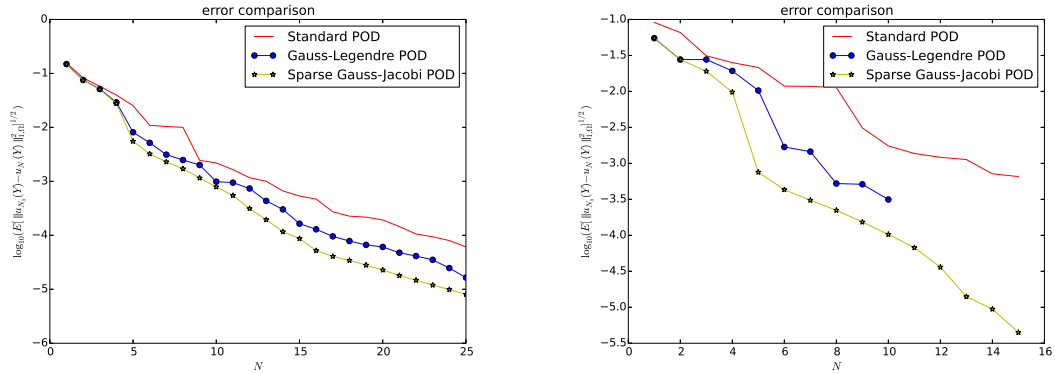


Fig. 5: Error analysis for standard, Gauss-Legendre and sparse Gauss-Jacobi POD algorithms, for the cases  $(\alpha, \beta) = (10, 10)$  (left) and  $(\alpha, \beta) = (75, 75)$  (right).

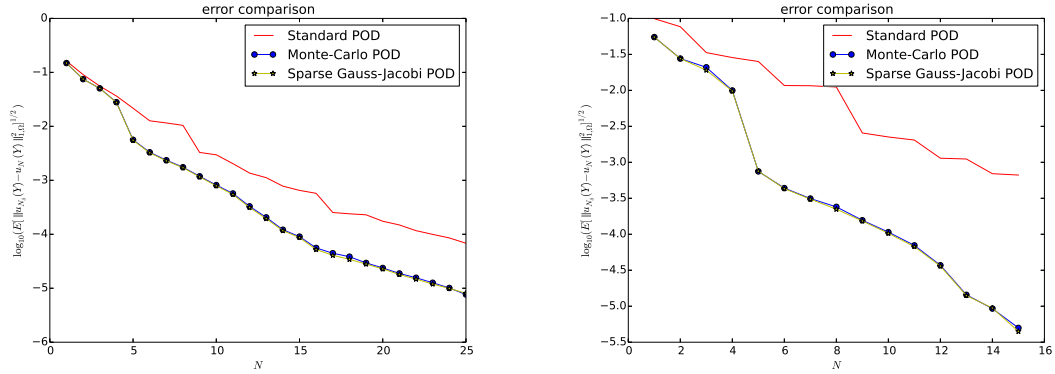


Fig. 6: Error analysis for standard, Monte-Carlo and sparse Gauss-Jacobi POD algorithms, for the cases  $(\alpha, \beta) = (10, 10)$  (left) and  $(\alpha, \beta) = (75, 75)$  (right).

quadrature rules perform similarly in terms of error, the sparse Gauss-Jacobi is obtained through a less dense training set  $\mathcal{E}_t$ , resulting in a less expensive Offline phase (see lines 2-4 of Algorithm 2(b)).

## 5 Conclusion

In this chapter we have presented two weighted reduced order methods for stochastic partial differential equations. A parametrized formulation has been introduced in order to profit by reduction methods based on the projection onto a low dimensional manifold, which has been identified by means of either a Greedy algorithm or a proper orthogonal decomposition, customarily employed in a deterministic setting. Corresponding weighted approaches have been introduced to properly account for the stochastic nature of the problem. The

main rationale for the weighting procedure has been emphasized, especially in relation to the minimization of the expectation of the mean square error between the high fidelity and reduced order approximations. Suitable discretizations of such expectation have been discussed, highlighting in particular the role of (possibly sparse) quadrature rules. A numerical test case for a stochastic linear elasticity problem has been presented. The results show that the weighted methods perform significantly better than the standard (deterministic) ones; bigger improvements have been obtained for more concentrated parameter distributions. Moreover, results demonstrate that the choice of a training set  $\mathcal{E}_t$  which is representative of the distribution is also essential. Further developments of this work concern the investigation of the proposed weighted reduced order methods in stochastic (nonlinear) fluid dynamics problems.

**Acknowledgements** We acknowledge the support by European Union Funding for Research and Innovation - Horizon 2020 Program - in the framework of European Research Council Executive Agency: H2020 ERC Consolidator Grant 2015 AROMA-CFD project 681447 “Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics”. We also acknowledge 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”. The computations in this work have been performed with RBniCS [1] library, developed at SISSA mathLab, which is an implementation in FEniCS [12] of several reduced order modelling techniques; we acknowledge developers and contributors to both libraries.

## References

1. Ballarin, F., Sartori, A., Rozza, G.: RBniCS - reduced order modelling in FEniCS. <http://mathlab.sissa.it/rbnics> (2015)
2. Bathelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics* **4**(12), 273–288 (2000)
3. Chen, P.: Model order reduction techniques for uncertainty quantification problems. Ph.D. thesis, École polytechnique fédérale de Lausanne EPFL (2014)
4. Chen, P., Quarteroni, A., Rozza, G.: A weighted reduced basis method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis* **51**(6), 3163–3185 (2013)
5. Chen, P., Quarteroni, A., Rozza, G.: Reduced basis methods for uncertainty quantification. *SIAM/ASA Journal on Uncertainty Quantification* **5**(1), 813–869 (2017)
6. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. *Numerical Algorithms* **3-4**(18), 209–232 (1998)
7. Guan, Q., Gunzburger, M., Webster, C.G., Zhang, G.: Reduced basis methods for nonlocal diffusion problems with random input data. *Computer Methods in Applied Mechanics and Engineering* **317**, 746–770 (2017)
8. Hesthaven, J., Rozza, G., Stamm, B.: *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer (2016)
9. Holtz, M.: *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*. Springer (2010)
10. Iliescu, T., Liu, H., Xie, X.: Regularized reduced order models for a stochastic Burgers equation. Submitted, arXiv preprint 1701.01155 (2017)
11. Ito, K., Ravindran, S.: A reduced-order method for simulation and control of fluid flows. *J. Comput. Phys.* **2**(143), 403–425 (1998)
12. Logg, A., Mardal, K.A., Wells, G.: *Automated solution of differential equations by the finite element method: The FEniCS book*, vol. 84. Springer Science & Business Media (2012)
13. Nobile, F., Tempone, R., Webster, C.G.: A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis* **5**(46), 2309–2345 (2008)
14. Novak, E., Ritter, K.: High dimensional integration of smooth functions over cubes. *Numerische Mathematik* **1**(75), 79–97 (1996)
15. Quarteroni, A., Valli, A.: *Numerical Approximation of Partial Differential Equations*. Springer (1994)
16. Rozza, G., Huynh, D., Nguyen, N., Patera, A.: 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)

17. Rozza, G., Huynh, D., Patera, A.: 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**(15), 229–275 (2008)
18. Rozza, G., Nguyen, N., Patera, A., Deparis, S.: 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)
19. Spannring, C.: Weighted reduced basis methods for parabolic PDEs with random input data. Ph.D. thesis, Graduate School CE, Technische Universität Darmstadt (2018)
20. Spannring, C., Ullmann, S., Lang, J.: A weighted reduced basis method for parabolic PDEs with random data. Submitted, arXiv preprint 1712.07393 (2017)
21. Torlo, D., Ballarin, F., Rozza, G.: Stabilized weighted reduced basis methods for parametrized advection dominated problems with random inputs. Submitted, arXiv preprint 1711.11275 (2017)
22. Venturi, L., Ballarin, F., Rozza, G.: A weighted POD method for elliptic PDEs with random inputs (2018). Submitted, arXiv preprint 1802.08724
23. Wasilkowski, G.W.: Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity* **1**(11), 1–56 (1995)
24. Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing* **3**(27), 1118–1139 (2006)