Enhanced goal-oriented error assessment and computational strategies in adaptive reduced basis solver for stochastic problems

Kevin Serafin[1], Benoît Magnain[1], Eric Florentin[1], Núria Parés[2] and Pedro Díez[2]

[1] *INSA-CVL / Université d'Orléans - PRISME, 88, Boulevard Lahitolle, F-18020 Bourges, France*
[2] *Laboratori de Càlcul Numèric (LaCàN) - Universitat Politècnica de Catalunya, Campus Nord UPC, E-08034 Barcelona, Spain*

SUMMARY

This work focuses on providing accurate low-cost approximations of stochastic finite elements simulations in the framework of linear elasticity. In [E. Florentin, P. Diez, Adaptive reduced basis strategy based on goal oriented error assessment for stochastic problems, Comput. Methods Appl. Mech. Engrg. 225-228 (2012) 116-127], an adaptive strategy has been introduced as an improved Monte-Carlo method for multi-dimensional large stochastic problems. We provide here a complete analysis of the method including a new enhanced goal-oriented error estimator and estimates of CPU cost gain. Technical insight of these two topics are presented in details and numerical examples show the interest of these new developments.

KEY WORDS: Reduced basis; Adaptivity; Stochastic modeling; Goal-oriented error assessment.

INTRODUCTION

Stochastic finite element methods (SFEM) are currently an essential tool for the quantitative prediction of the response of mechanical models. Uncertainties can be modelled via random variables and their associated probability distributions. A state of the art of stochastic methods can be found in [1, 2, 3] and the references provided therein.

The numerical approximation of any stochastic model is affected by errors originated from two distinct sources. The space functions (e.g. displacement field) are approximated in a discrete functional space, associated with a Finite Element (FE) discretization, for example. The stochastic dependence of the solution is resolved also discretely, either with a simple non-intrusive approach like the Monte Carlo (MC) methodology of with a more sophisticated (and intrusive) strategy. In the simpler case, the FE error depends on the mesh parameter,h, and the MC error in the stochastic description of the outcome depends of the number of throws, N_{MC}. The computational cost increases when h decreases (the cost of each FE problem is much larger) and when N_{MC} increases (there are more FE problems to solve). In this work, we consider h and N_{MC} fixed by the user and we focus in providing techniques to efficiently compute each of the N_{MC}, by, in most of the throws, avoiding solving the full FE problem and replacing it by a reduced order model.

In that sense, using a Reduced Basis method to solve a given Monte Carlo throw is an alternative to different approaches using response surface based meta-models, see [4, 5, 6, 7]. The main conceptual difference lies in the fact that the Reduced Basis approach does solve the original equation in the new configuration (in anapproximation functional space spanned by the set of available solutions, corresponding to other configurations). This is in contrast with the response surface approach, in which the solution corresponding to a new configuration is *interpolated* using the solutions available (the equation to be solved is actually used to compute the solutions available, not the new one). Here, we complement the methodology with an error assessment strategy based on the residual of the equation solved for the new configuration, which is fully consistent with the Reduced Basis approach. A different strategy that could be also treated within a similar framework is solving the full FE problem of the new configuration with a smart preconditioning based on previous solutions, see [8].

In [9], a reduced basis methodology to reduce the cost of Monte Carlo simulations is presented, offering an attractive framework for solving stochastic problems with a large number of parameters. The idea is simple and effective because the different Monte Carlo shots lead to similar FE problems and therefore the reduced basis approach is highly competitive. Other interesting recent references on this subject are [10, 11, 12, 13, 14, 15]. An adaptative Reduced-Basis (RB) strategy, based on local error estimator was introduced in [16] and a sensibility analysis on a Quantity of Interest (QOI) was performed. Recently, a different solution has been presented in [17] to obtain a reliable and uncostly solution in the same framework.

We provide here an improvement of the method developed in [16] regarding two different aspects: quality and cost. Specifically, the quality is improved by using reduced basis both for the original primal problem and for the adjoint problem associated to a

given quantity of interest (and not only in the original problem). The computational cost is reduced by an optimal implementation of the adaptive strategy. The efficiency of the proposed implementation is tested using a C++ code which in turn uses appropriate open source libraries.

The paper is structured as follows. The first section introduces the mechanical problem to solve in the framework of stochastic modeling. Section 2 presents the improvement of the error estimator presented in [16] by introducing a reduced basis approximation of the adjoint problem, along with some numerical examples that validate the presented technique. In section 3, a precise analysis of the Computational Processing Unit (CPU) gain obtained thanks to the proposed new optimal implementation is given. A parametric analysis of the method is performed on a 2D example, both regarding the computational cost and the quality of the provided approximations. Finally, section 4 shows the results of the presented goal-oriented error estimation techniques on a large 3D mechanical simulation with heterogeneous material.

1. STOCHASTIC PROBLEM

1.1. Continuous model

Let Ω be a bounded domain representing a mechanical structure and let $(\Theta, \mathcal{T}, P)$ be a complete probability space, where Θ is the set of outcomes, $\mathcal{T}$ is the σ-algebra of events and $P : \mathcal{T} \to [0, 1]$ is a probability measure. The stochastic linear elasticity boundary value problem consists in finding the random displacement field $u(\mathbf{x}, \theta)$, where $\mathbf{x} \in \Omega$ stands for the position vector and $\theta \in \Theta$ is used to denote the randomness, such that P-almost surely in Θ the following equations hold

$$-\mathbf{div}\left(\mathbf{K}(\mathbf{x}, \theta)\boldsymbol{\varepsilon}\left[\mathbf{u}(\mathbf{x}, \theta)\right]\right) = \mathbf{f}_d(\mathbf{x}) \qquad \text{in } \Omega \qquad (1a)$$

$$\mathbf{K}(\mathbf{x}, \theta)\boldsymbol{\varepsilon}\left[\mathbf{u}(\mathbf{x}, \theta)\right] \cdot \mathbf{n} = \mathbf{g}_d(\mathbf{x}) \qquad \text{on } \partial_N\Omega \qquad (1b)$$

$$\mathbf{u}(\mathbf{x}, \theta) = 0 \qquad \text{on } \partial_D\Omega. \qquad (1c)$$

The sets $\partial_D\Omega$ and $\partial_N\Omega$ are the two complementary disjoint parts in which the boundary of the domain $\partial\Omega$ is divided, that is, $\overline{\partial_D\Omega} \cup \overline{\partial_N\Omega} = \partial\Omega$ and $\partial_D\Omega \cap \partial_N\Omega = \varnothing$. Furthermore, the Dirichlet boundary $\partial_D\Omega$ is assumed to be a non-empty set.

For simplicity of presentation, the stochastic behavior of the model is only included in the material parameters, assuming that the Hooke tensor is a random field $\mathbf{K}(\mathbf{x}, \theta)$. However, the presented techniques are also readily applicable to problems including a non-homogeneous random displacement field $\mathbf{u}_d(\mathbf{x}, \theta)$ imposed on $\partial_D\Omega$, or including a random prescribed traction field $\mathbf{g}_d(\mathbf{x}, \theta)$ or a random prescribed body force field $\mathbf{f}_d(\mathbf{x}, \theta)$. As a random field, $\mathbf{K}(\mathbf{x}, \theta)$ is a function mapping each position vector $\mathbf{x}$ to a random variable, which, in the following, it is assumed to be a second-order stochastic process with continuous covariance function, depending on the distance between the locations. Ideally, the full law of the field $\mathbf{K}(\mathbf{x}, \theta)$ should be known, or either the infinite set of n-th order marginal densities should be given. For instance, sometimes it is possible to assume that $\mathbf{K}(\mathbf{x}, \theta)$ is a Gaussian process. However, in practice, $\mathbf{K}(\mathbf{x}, \theta)$ may represent a more complex stochastic processes, where fewer information is known. Typically, if no precise information on $\mathbf{K}(\mathbf{x}, \theta)$ is given, one can assume that the first-order marginal Probability Density Functions (PDF) are the same and known for all the position vectors $\mathbf{x}$, assume a fixed form of the cross-correlation, and define the random process $\mathbf{K}(\mathbf{x}, \theta)$ using only this information.

The strong problem (1) is equivalent to its standard weak form: find $\mathbf{u}(\mathbf{x}, \theta) \in \mathcal{U}$ such that

$$a(\mathbf{u}(\mathbf{x}, \theta), \mathbf{w}(\mathbf{x})) = \ell(\mathbf{w}(\mathbf{x})) \qquad \forall \ \mathbf{w}(\mathbf{x}) \in \mathcal{U} \qquad (2)$$

where the bilinear form expressing the energy product $a(\cdot, \cdot)$ and the linear form including the loading $\ell(\cdot)$ are

$$a(\mathbf{u}, \mathbf{w}) = \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{w}) \, d\Omega \quad , \quad \ell(\mathbf{w}) = \int_{\Omega} \mathbf{f}_d \cdot \mathbf{w} \, d\Omega + \int_{\partial_N\Omega} \mathbf{g}_d \cdot \mathbf{w} \, d\Gamma,$$

and $\mathcal{U}$ is the set of admissible displacements (verifying the homogeneous boundary conditions (1c)), namely $\mathcal{U} = [\mathcal{H}_0^1(\Omega)]^d \otimes \mathcal{L}_P^2(\Theta)$, where $\mathcal{H}_0^1(\Omega)$ is the standard Sobolev space of functions defined in Ω such that they verify the homogeneous Dirichlet boundary conditions and that both the functions and their first derivatives are square-integrable, the superscript d denoting the spatial dimension of the problem, and $\mathcal{L}_P^2(\Theta)$ is the space of square-integrable functions with respect to the probability P.

Here, $\boldsymbol{\varepsilon}(\mathbf{v})$ is the second order deformation tensor, which is defined as the symmetric part of the gradient tensor $\nabla\mathbf{v}$, that is, $\boldsymbol{\varepsilon}(\mathbf{v}) = (\nabla\mathbf{v} + \nabla\mathbf{v}^T)/2$, and the stress tensor $\boldsymbol{\sigma}(\mathbf{v})$, is related to the deformation tensor through the linear constitutive relation $\boldsymbol{\sigma}(\mathbf{v}) = \mathbf{K}\boldsymbol{\varepsilon}(\mathbf{v})$.

1.2. Quantity of interest

The purpose of the stochastic analysis is to determine reliable statistical information of a response quantity of interest (QoI). Note that since the solution $\mathbf{u}(\mathbf{x}, \theta)$ is a random field, any output computed from this solution is a random quantity, and therefore the statistics of this output (expected value, variance, ...) are the relevant information to be estimated. Here, it is assumed that the quantity of interest is a scalar quantity expressed as a linear output of $\mathbf{u}(\mathbf{x}, \theta)$, namely

$$Q(\theta) = \ell^Q(\mathbf{u}(\mathbf{x}, \theta)), \qquad (3)$$

where $\ell^Q(\cdot)$ is a linear functional having the same form of $\ell(\cdot)$, namely

$$\ell^Q(\mathbf{w}) = \int_\Omega \mathbf{f}_d^{\mathrm{Ad}} \cdot \mathbf{w} \, d\Omega + \int_{\partial_N \Omega} \mathbf{g}_d^{\mathrm{Ad}} \cdot \mathbf{w} \, d\Gamma,$$

where for simplicity of presentation, the prescribed adjoint traction and body force fields, $\mathbf{g}_d^{\mathrm{Ad}}(\mathbf{x})$ and $\mathbf{f}_d^{\mathrm{Ad}}(\mathbf{x})$ respectively, are assumed to be deterministic fields, not depending on θ.

It is worth noting that here the quantity of interest is restricted to be linear with respect to the displacement field. However, nonlinear quantities of interest may also be handled using linearization techniques like the ones introduced in [18, 19].

Given a numerical approximation $\mathbf{u}_h$ of the problem solution, the QoI is approximated by $Q_h(\theta) = \ell^Q(\mathbf{u}_h(\mathbf{x}, \theta))$ and the error in the quantity of interest is

$$e^Q(\theta) = Q(\theta) - Q_h(\theta).$$

In order to obtain an alternative representation to the error in the QoI, an auxiliary adjoint problem is introduced: find $\mathbf{v}(\mathbf{x}, \theta) \in \mathcal{U}$ such that

$$a(\mathbf{w}(\mathbf{x}), \mathbf{v}(\mathbf{x}, \theta)) = \ell^Q(\mathbf{w}(\mathbf{x})) \qquad \forall \, \mathbf{w}(\mathbf{x}) \in \mathcal{U}. \tag{4}$$

Then, using equations (4) and (2), the error in the QoI can be expressed as an explicit function of the residual in the weak form of the problem

$$e^Q(\theta) = \ell^Q(\mathbf{u}) - \ell^Q(\mathbf{u}_h) = a(\mathbf{u}, \mathbf{v}) - a(\mathbf{u}_h, \mathbf{v}) = \ell(\mathbf{v}) - a(\mathbf{u}_h, \mathbf{v}) =: R(\mathbf{u}_h; \mathbf{v}).$$

1.3. Discrete surrogate model

As previously mentioned, the goal of this work is to approximate the PDF of $Q(\theta)$ with a controlled prescribed accuracy. Since the dependence of the solution $\mathbf{u}(\mathbf{x}, \theta)$ with respect to the variable θ is usually given by a nonlinear intricate transformation, determining the exact distribution of $Q(\theta)$ requires an analytic determination of this intricate functional transformation. Non-intrusive approaches seek to approximate the PDF of $Q(\theta)$ without any analytic determination of this functional transformation while only using standard available computational codes to approximate the underlying deterministic boundary value problem associated to (1), for instance taking a particular realization of θ.

Here the Monte Carlo technique along with the finite element method is used to get an approximation of the statistics of $Q(\theta)$ in a non-intrusive manner. The sketch of the strategy can be outlined as follows:

- *Discretization of the input random field:* First, the stochasticity of the system is simplified to be able to represent the stochastic input $\mathbf{K}(\mathbf{x}, \theta)$ as a finite-dimensional linear combination of independent random variables.

- *Monte Carlo assessment of the QoI:* Then, the Monte Carlo method is used to assess the statistics of $Q(\theta)$ using the three standard steps described below:
  - *Monte Carlo sampling:* Given the discretization of the input random field, it is trivial to generate N_{MC} realizations of the random input, namely $\mathbf{K}(\mathbf{x}, \theta_k), k = 1, \ldots, N_{\mathrm{MC}}$.
  - *Spatial discretization:* For each of these realizations, the deterministic boundary value problem (1) (freezing randomness) is approximated using the finite element method.
  - *Statistics of $Q(\theta)$:* Gathering all the information given by the finite element approximations $\mathbf{u}_h(\mathbf{x}, \theta_k)$, approximated probability density functions of both $\mathbf{u}(\mathbf{x}, \theta)$ and $Q(\theta)$ are obtained.

The basic steps of this approach are described below in order to introduce the necessary notations.

1.3.1. Discretization of the input random field: finite stochastic dimension The uncertainty of the problem at hand is taken into account in the Hooke tensor $\mathbf{K}(\mathbf{x}, \theta)$. In particular, the variability of the Hooke tensor is modeled through the Young modulus $E(\mathbf{x}, \theta)$. $E(\mathbf{x}, \theta)$ represents a random scalar at each point of the domain, and thus consists of an infinite number of usually correlated random variables. For computational purposes, here the random field E is discretized, i.e. expressed as a finite number of uncorrelated random variables, using the truncated Karhunen-Loève decomposition [20, 21].

Assuming that the spatial correlation function $\mathrm{cov}(E(\mathbf{x}, \theta), E(\mathbf{x}', \theta))$ is regular enough (essentially, it reduces to have square-integrable probability distributions and symmetric correlation functions), the Karhunen-Loève decomposition allows representing a random field by a sum of mutually uncorrelated (zero-mean) scalar random variables multiplied by deterministic functions of $\mathbf{x}$, namely

$$E(\mathbf{x}, \theta) = E_0(\mathbf{x}) + \sum_{i=1}^{+\infty} \sqrt{\lambda_i} E_i(\mathbf{x}) \xi_i(\theta), \tag{5}$$

where $E_i(\mathbf{x})$ and λ_i, $i = 1, 2, \ldots$, are the eigenfunctions and eigenvalues of the covariance operator associated with the random field $E(\mathbf{x}, \theta)$ and $E_0(\mathbf{x})$ is the mean value of $E(\mathbf{x}, \theta)$.

Then, thanks to the linearity of the Hooke tensor with respect to the Young modulus, the truncated Karhunen-Loève approximation of the Hooke tensor $\mathbf{K}(\mathbf{x}, \theta)$ is

$$\mathbf{K}(\mathbf{x}, \theta) \approx \mathbf{K}_0(\mathbf{x}) + \sum_{i=1}^{N_{\mathrm{KL}}} \sqrt{\lambda_i} \mathbf{K}_i(\mathbf{x}) \xi_i(\theta) \tag{6}$$

where $\mathbf{K}_0(\mathbf{x})$ and $\mathbf{K}_i(\mathbf{x})$ are the Hooke tensors associated to $E_0(\mathbf{x})$ and $E_i(\mathbf{x})$ respectively.

Thus, the stochasticity of the system is approximated by N_{KL} uncorrelated (zero-mean) random variables $\xi_i(\theta)$, $i = 1, 2, \ldots N_{\mathrm{KL}}$. If $E(\mathbf{x}, \theta)$ can be assumed to be a Gaussian process, uncorrelation and independence are equivalent, and furthermore, linear combinations of Gaussian random variables remain Gaussian-distributed. Therefore, if $E(\mathbf{x}, \theta)$ is a Gaussian process, the random variables $\xi_i(\theta)$ are independent Gaussian random variables and (6) provides a way to parameterize a Gaussian process by a finite number of independent Gaussian random variables. When the input random process $E(\mathbf{x}, \theta)$ can not be assumed to be Gaussian, its parametrization in terms of $\xi_i(\theta)$ is more challenging. If the full stochastic law of $E(\mathbf{x}, \theta)$ is known, the uncorrelated random variables can be computed explicitly as

$$\xi_i(\theta) = \frac{1}{\lambda_i} \int_\Omega (E(\mathbf{x}, \theta) - E_0(\mathbf{x})) E_i(\mathbf{x}) d\Omega. \tag{7}$$

However usually no precise information on $\mathbf{K}(\mathbf{x}, \theta)$ is given, and equation (7) can not be used to compute the uncorrelated random variables $\xi_i(\theta)$ describing the process. The computation of appropriate uncorrelated random variables $\xi_i(\theta)$ parametrizing the problem is still and open research topic (see for instance [22, 23]) and it is not discussed in the present work. Here a simplified approach frequent in the literature is used (see [24, 25, 26, 27]). To simplify the stochasticity of the system, two additional assumptions are introduced: first, the random variables $\{\xi_i(\theta), 1 \le i \le N_{\mathrm{KL}}\}$ are assumed to be independent (and thus mutually uncorrelated) and second, they are assumed to be identically distributed random variables (for instance, uniformly distributed random variables in the range $[-\sqrt{3}, \sqrt{3}]$). The second assumption is only made to uniquely determine the Hooke tensor and other distributions may be considered, as will be seen in the numerical examples, while the first assumption is useful in order to be able to easily generate realizations of $\mathbf{K}(\mathbf{x}, \theta)$. The independence assumption reduces the problem of generating a realization of $\mathbf{K}(\mathbf{x}, \theta)$ to generate the N_{KL} independent realizations of the random variables $\xi_i(\theta)$. It is worth noting that the assumptions introduced here are only made to simplify the problem of generating the realizations of $E(\mathbf{x}, \theta)$ (Monte-Carlo sampling step), and thus the present work is valid also for any more general Karhunen-Loeve decomposition given in the form of (5) or also to any other linear or nonlinear approximation of the input random field allowing to explicitly generate proper realizations of $E(\mathbf{x}, \theta)$, including for instance modelling $E(\mathbf{x}, \theta)$ using a polynomial chaos expansion [3, 28].

1.3.2. Spatial discretisation For each Monte Carlo realization $\mathbf{K}(\mathbf{x}, \theta_k)$, the corresponding solution $\mathbf{u}(\mathbf{x}, \theta_k)$ of problem (1) has to be properly approximated. Note that for a fixed given value of θ, namely θ_k, problem (1) is just a deterministic linear elasticity problem associated with the spatially-dependent Hooke tensor $\mathbf{K}(\mathbf{x}, \theta_k)$. Here, the infinite-dimensional elasticity problem is approximated using the standard Galerkin finite element method.

Let $N_i(\mathbf{x})$, $i = 1, 2, \ldots, N_{\mathrm{FE}}$ be the finite element shape functions associated to the finite dimensional mesh of characteristic size h which defines the finite-dimensional space $\mathcal{U}_h \subset \mathcal{U}$

$$\mathcal{U}_h = \mathrm{span}\left\{ N_1, N_2, \ldots, N_{N_{\mathrm{FE}}} \right\}. \tag{8}$$

Then, the numerical approximation of $\mathbf{u}(\mathbf{x}, \theta_k)$ in $\mathcal{U}_h$ is

$$\mathbf{u}(\mathbf{x}, \theta_k) \approx \mathbf{u}_h(\mathbf{x}, \theta_k) = \sum_{i=1}^{N_{\mathrm{FE}}} \mathbf{u}_i(\theta_k) N_i(\mathbf{x}), \tag{9}$$

where $\mathbf{u}_i(\theta_k)$ are the vector nodal values of the approximation at the mesh nodes $\mathbf{x}_i$. Introducing the global vector of unknowns $\mathbf{U}(\theta_k) = [\mathbf{u}_1(\theta_k)^T \ \mathbf{u}_2(\theta_k)^T \ \ldots \ \mathbf{u}_{N_{\mathrm{FE}}}(\theta_k)^T]^T$ the corresponding discretized form of (2) can be written as a linear system of equations

$$\mathbb{K}(\theta_k) \mathbf{U}(\theta_k) = \mathbf{F}, \tag{10}$$

where $\mathbb{K}$ is the finite element stiffness matrix (discretized form of $a(\cdot, \cdot)$) and $\mathbf{F}$ is the nodal force vector (discretized form of $\ell(\cdot)$). The size of this system of equations is $(d \cdot N_{\mathrm{FE}}) \times (d \cdot N_{\mathrm{FE}})$ where d is the spatial dimension of the problem.

It is worth noting that, since the Hooke tensor associated with the realization θ_k is computed using equation (6) and since the bilinear form $a(\cdot, \cdot)$ is a linear expression on the Hooke tensor, the stiffness matrix corresponding to the realization θ_k reads

$$\mathbb{K}(\theta_k) \approx \mathbb{K}_0 + \sum_{i=1}^{N_{\mathrm{KL}}} \sqrt{\lambda_i} \mathbb{K}_i \xi_i(\theta_k) \tag{11}$$

where $\mathbb{K}_0$ and $\mathbb{K}_i$ are the (deterministic) stiffness matrices corresponding to the Hooke tensor distributions described by $\mathbf{K}_0(\mathbf{x})$ and $\mathbf{K}_i(\mathbf{x})$ respectively.

1.3.3. Statistics of $Q(\theta)$ The Monte Carlo technique approximates the statistics of a given random quantity of interest $Q(\theta)$ by generating a large number of realizations of θ, namely $\{\theta_k\}_{k=1,...,N_{\mathrm{MC}}}$, and properly aggregating the computed approximations $\{\mathbf{u}_h(\mathbf{x}, \theta_k)\}_{k=1,...,N_{\mathrm{MC}}}$.

That is, N_{MC} realizations of $\mathbb{K}(\theta)$ are generated using independent realizations of the random variables $\{\xi_i(\theta)\}_{i=1,...,N_{\mathrm{KL}}}$ appearing in (11), and then, the realizations $\{\mathbf{u}_h(\mathbf{x}, \theta_k)\}_{k=1,...,N_{\mathrm{MC}}}$ (or equivalently $\{\mathbf{U}(\theta_k)\}_{k=1,...,N_{\mathrm{MC}}}$) are obtained by solving N_{MC} linear systems of equations (10). The PDF of $Q(\theta)$ is then characterized by its expectation and variance as:

$$\mathbb{E}[Q(\theta)] \approx \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} Q_h(\theta_k) = \mathbb{E}_{\mathrm{MC}} \quad , \quad \mathbb{V}[Q(\theta)] \approx \frac{1}{N_{\mathrm{MC}} - 1} \sum_{i=1}^{N_{\mathrm{MC}}} \left(Q_h(\theta_k)^2 - \mathbb{E}_{\mathrm{MC}}^2 \right) \tag{12}$$

Also, higher order moments of $Q(\theta)$ or its different percentiles can be analogously obtained.

In order to have an accurate approximation of the PDF of $Q(\theta)$, the number of realizations N_{MC} must be very large. Note that the Monte Carlo simulation process converges very slowly to the answer associated with the quantity of interest $Q_h(\theta)$ (approximated in space, exact in the stochastic dimension), which is ideally obtained after an infinite number of realizations. The convergence rate is estimated to be of order $\mathcal{O}(1/\sqrt{N_{\mathrm{MC}}})$, see for example [29]. Therefore, a fast and competitive method to obtain the realizations of $\mathbf{u}_h(\mathbf{x}, \theta_k)$ is necessary to be able to get accurate approximations of the QoI by including a large number of Monte Carlo runs in (12).

1.4. 2D test case

To fix the ideas, a 2D simple test case similar to the one proposed in [16] is introduced here. As figure 1 shows, a square domain of side $L = 100$ is compressed by a uniform pressure ($P = 0.03$) to a portion of its upper face $\partial_N \Omega = \{(x, L/2) \in \mathbb{R}^2, x \in (-2L/5, 2L/5)\}$. Vertical displacements are set to zero on the bottom of the square (with free horizontal displacements) and horizontal displacements are set to zero on the lateral sides (with free vertical displacements).
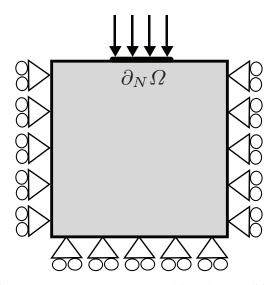

Figure 1. 2D test case: geometry and boundary conditions.

The spatial meshes consist of linear 4-node squares where the number of elements is $n_{\mathrm{el}} = n_{\mathrm{L}}^2$ for different values of the number of 1D elements n_{L} of the sides of the square. Table I provides the precise relation between n_{L} and the number of degrees of freedom of the problem n_{dof} for the considered meshes, where the total number of degrees of freedom is computed as $2(n_{\mathrm{L}} + 1)^2$ minus the number of prescribed zero displacements.

n_{L}	10	20	40	80	160
n_{dof}	209	819	3 239	12 879	51 359

Table I. 2D test case: number of degrees of freedom of the spatial finite element meshes

The mechanical behavior of the material is modeled by the linear elastic Hooke's law, where the uncertainty of the problem is taken into account in the Hooke tensor $\mathbf{K}(\mathbf{x}, \theta)$ through the Young modulus $E(\mathbf{x}, \theta)$. This random field $E(\mathbf{x}, \theta)$ is characterized by its uniform average $\mathbb{E}[E(\mathbf{x}, \theta)] = E_0(\mathbf{x}) = 1$ and its spatial correlation function $\mathcal{C}(\mathbf{x}, \mathbf{x}')$ given by the exponential kernel.

$$\mathcal{C}(\mathbf{x}, \mathbf{x}') = \mathrm{cov}(E(\mathbf{x}, \theta), E(\mathbf{x}', \theta)) = \alpha^2 e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|}{L}}, \tag{13}$$

where $\|\mathbf{x} - \mathbf{x}'\|$ is the standard euclidean distance between $\mathbf{x}$ and $\mathbf{x}'$. It is worth noting that since the variance of $E(\mathbf{x}, \theta)$ is $\mathbb{V}[E(\mathbf{x}, \theta)] = \alpha^2$, α is the standard deviation of the random variable $E(\mathbf{x}, \theta)$. This parameter is used to modulate the deviation of the randomness of the problem from the mean field, which allows testing the sensitivity of the method with respect to the magnitude of the randomness. In this case, the truncated Karhunen-Loeve decomposition of $E(\mathbf{x}, \theta)$ is

$$E(\mathbf{x}, \theta) = E_0(\mathbf{x}) + \alpha \sum_{i=1}^{N_{\mathrm{KL}}} \sqrt{\lambda_i} E_i(\mathbf{x}) \xi_i(\theta), \tag{14}$$

where now the deviation α is shown explicitly in the equation, and λ_i and $E_i(\mathbf{x})$ are the eigenvalues and eigenvectors of the unit-variance covariance operator $\mathcal{C}_1(\mathbf{x}, \mathbf{x}') = e^{-\|\mathbf{x}-\mathbf{x}'\|/L}$.

The eigenvalues and eigenvectors, λ_i and E_i, solution of the Fredholm equation for the unit-variance covariance operator $\mathcal{C}_1(\mathbf{x}, \mathbf{x}')$

$$\int_\Omega \mathcal{C}_1(\mathbf{x}, \mathbf{x}') E_i(\mathbf{x}') d\mathbf{x}' = \lambda_i E_i(\mathbf{x})$$

are approximated by sampling in N_{FE} points ($\mathbf{x}_i$, for $i = 1, \ldots, N_{\mathrm{FE}}$, possibly the nodes of the spatial discretization used in the FEM, see table I) and using a numerical integration scheme with the same points and uniform weights (thus, equal to $|\Omega|/N_{\mathrm{FE}}$), see [30] for a detailed explanation. Moreover, the orthonormality property of the eigenvectors is also enforced with respect to the scalar product corresponding to the same numerical integration scheme.

In this case, the eigenvalues λ_i and eigenvectors E_i are computed using a singular value decomposition (SVD) of the discrete covariance matrix $\mathbb{C}$ where $\mathbb{C}_{ij} = \mathcal{C}_1(\mathbf{x}_i, \mathbf{x}_j)$ for $i, j = 1, \ldots, N_{\mathrm{FE}}$. The eigenvalues and normalized eigenvectors of the covariance function coincide with the singular values and singular vectors of $\mathbb{C}$ after a scaling factor is applied: the eigenvalues λ_i are obtained from the singular values by multiplying them by $|\Omega|/N_{\mathrm{FE}}$ while the normalized eigenvectors are obtained by multiplying the singular vectors (which are orthonormal with respect to the Euclidean norm in $\mathbb{R}^{N_{\mathrm{FE}}}$) by $\sqrt{N_{\mathrm{FE}}/|\Omega|}$. Note that, if the values of λ_i and E_i are computed only to be used in (14), there is no need to apply the scaling factor, since the product $\sqrt{\lambda_i} E_i(\mathbf{x})$ has the same value before and after the scaling factor is applied.

Among the $n_{\mathrm{dof}}/2$ eigenvalues of $\mathbb{C}$, the N_{KL} greatest values are kept. Figure 2 shows the 20 largest eigenvalues obtained in this example.
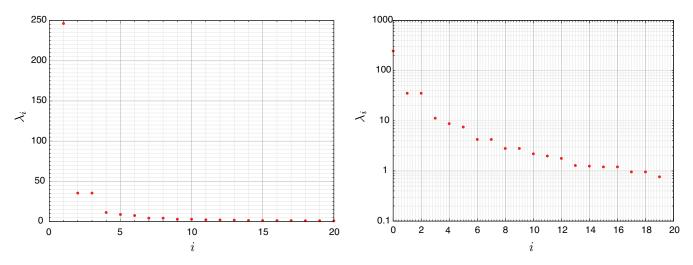

Figure 2. 2D test case: 20 largest eigenvalues λ_i of the Fredholm equation obtained using the SVD technique, normal (left) and semilog (right) scales.

In this test case, the number of terms in the Karhunen-Loeve decomposition is set to $N_{\mathrm{KL}} = 20$ and the uncorrelated zero-mean random variables $\{\xi_i(\theta)\}_{i=1\ldots,N_{\mathrm{KL}}}$ appearing in the Karhunen-Loeve decomposition are assumed to be identically distributed with a non-standard nonlinear distribution obtained as

$$\xi_i = \frac{2}{\sqrt{\pi^2 - 8}} \arcsin\left(\mathrm{Erf}\left(\frac{\mathcal{N}(0,1)}{\sqrt{2}}\right)\right) \tag{15}$$

where $\mathcal{N}(0, 1)$ stands for the standard normal distribution and Erf is the error function. This specific distribution allows dealing with zero-mean and unit-variance random variables with a finite support $\xi_i(\theta) \in \left[-\pi/\sqrt{\pi^2 - 8}, \pi/\sqrt{\pi^2 - 8}\right]$. With this information at hand, the value of α is chosen carefully so that $E(\mathbf{x}, \theta)$ is guaranteed to be strictly positive for all N_{KL}.

The quantity of interest considered here is the vertical displacement at the middle point of the upper surface, namely:

$$Q(\theta) = \ell^Q(\mathbf{u}(\mathbf{x}, \theta)) = u_y(\mathbf{x}^Q, \theta) \quad \text{with} \quad \mathbf{x}^Q = [0, L/2]^T, \tag{16}$$

where $u_y(\mathbf{x}^Q, \theta)$ denotes the vertical displacement of $\mathbf{u}(\mathbf{x}, \theta)$ at point $\mathbf{x}^Q$. Due to the symmetry of the elasticity operator, the adjoint problem takes the same form as (1) where the external loads are $\mathbf{f}_d^{\mathrm{Ad}}(\mathbf{x}) = 0$ and

$$\mathbf{g}_d^{\mathrm{Ad}}(\mathbf{x}) = \begin{cases} [0, -1]^T & \text{for } \mathbf{x} = \mathbf{x}^Q \\ 0 & \text{elsewhere in } \partial_N \Omega. \end{cases}$$

To obtain good approximations of the statistics of the quantity of interest $\mathbb{E}[Q(\theta)]$ and $\mathbb{V}[Q(\theta)]$ a Monte Carlo simulation with $N_{\mathrm{MC}} = 10^5$ is performed. The resulting statistics for different values of the deviation α and for the mesh with $n_{\mathrm{dof}} = 51359$ are shown in table II.

α	5%	10%	20%	25%
$\mathbb{E}[Q(\theta)]$	-1.6901	-1.7016	-1.7512	-1.7942
$\mathbb{V}[Q(\theta)]$	0.0048	0.02	0.0952	0.1742

Table II. 2D test cast: statistics of $Q(\theta)$ obtained for the mesh with $n_{\mathrm{dof}} = 51359$.

In the forthcoming sections, the robustness of the presented techniques is analyzed by keeping N_{MC} constant and increasing the difficulty of the problem by either decreasing the size of the finite element mesh (or equivalently increasing n_{dof}) or by increasing the value of the deviation α. Note that the parameter α allows varying the spatial variability of the random field $E(\mathbf{x}, \theta)$. A larger deviation yields a tougher problem since the Young modulus $E(\mathbf{x}, \theta)$ will strongly differ from its mean value $E_0(\mathbf{x})$, and therefore the difference between the solutions $u(\mathbf{x}, \theta)$ of two different realizations will also be large.

Table III shows the CPU time of the *full* Monte-Carlo analysis, denoted by t_{MC}, as the computational complexity of the Finite Element Method is increased. Hereafter, the phrasing *full* is used in contraposition to *reduced-basis*. Thus, a full Monte-Carlo analysis uses standard non-reduced finite element approximations for all the Monte-Carlo realizations. It can be appreciated that

n_{dof}	209	819	3 239	12 879	51 359
t_{MC}	32 s	3 min	17 min	2h30	21h20

Table III. 2D test case: evolution of the CPU time as n_{dof} is increased.

if the computation of all the different Monte-Carlo throws is performed independently (without benefitting from the information obtained as the process advances to reduce the computational cost), the CPU time associated to the finer finite element meshes becomes very large.

2. ADAPTIVE REDUCED BASED METHOD WITH IMPROVED ERROR ESTIMATOR

The proposed non-intrusive strategy to obtain the statistics of $Q(\theta)$ is extremely simple because it decouples the approximation of the stochastic behavior and the solution of the deterministic mechanical model. The deterministic numerical solver can be used as a black box, without any modification in the code associated with the stochasticity. The random character is accounted for by generating a large number of realizations and solving the corresponding deterministic problems. However, as shown in the previous section, the numerical cost is very large unless the cost of the deterministic numerical solver is drastically reduced. The reduced basis method is particularly interesting in this case, since we need to solve a large number of problems with similar characteristics. By creating a data base of N_{RB} solutions the approximation of the deterministic solution for a new realization can be computed by solving a small $N_{\mathrm{RB}} \times N_{\mathrm{RB}}$ linear system of equations instead of (10). The key point is that in general N_{RB} is much smaller than $d \cdot N_{\mathrm{FE}}$.

2.1. Reduced basis approximation

Let $\{\mathbf{u}_{h(1)}, \mathbf{u}_{h(2)}, \ldots, \mathbf{u}_{h(N_{\mathrm{RB}})}\}$ be a collection of N_{RB} linearly independent solutions (corresponding to different realizations of θ) described by the corresponding vectors of nodal values $\{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_{N_{\mathrm{RB}}}\}$ and let $\mathbb{U}_{\mathbb{RB}} = [\mathbf{U}_1 \, \mathbf{U}_2 \, \ldots \, \mathbf{U}_{N_{\mathrm{RB}}}]$ be the $(d \cdot N_{\mathrm{FE}}) \times N_{\mathrm{RB}}$ matrix storing all the snapshot solutions. Then, for a given new realization θ_k, the reduced basis approximation of $\mathbf{u}_h(\mathbf{x}, \theta_k)$ is

$$\mathbf{u}_{h,\mathrm{RB}}(\mathbf{x}, \theta_k) = \sum_{i=1}^{N_{\mathrm{RB}}} a_i(\theta_k) \mathbf{u}_{h(i)}(\mathbf{x})$$

which can also be written in vector form as

$$\mathbf{U}_{\mathrm{RB}}(\theta_k) = \sum_{i=1}^{N_{\mathrm{RB}}} a_i(\theta_k) \mathbf{U}_i = \mathbb{U}_{\mathbb{RB}} \, \mathbf{a}(\theta_k)$$

where the reduced base vector $\mathbf{a}(\theta_k) = [a_1(\theta_k)\, a_2(\theta_k)\, \ldots\, a_{\mathrm{RB}}(\theta_k)]^T$ is the solution of

$$(\mathbb{U}_{\mathrm{RB}}^T \mathbb{K}(\theta_k)\mathbb{U}_{\mathrm{RB}})\,\mathbf{a}(\theta_k) = \mathbb{U}_{\mathrm{RB}}^T \mathbf{F}. \tag{17}$$

2.2. Error assessment

For each realization θ_k, the error in the approximation of $\mathbf{u}(\mathbf{x}, \theta_k)$ given by the reduced basis solution $\mathbf{u}_{h,\mathrm{RB}}(\mathbf{x}, \theta_k)$ can be decomposed into

$$\mathbf{e}_{\mathrm{RB}}(\mathbf{x}, \theta_k) = \mathbf{e}_h(\mathbf{x}, \theta_k) + \mathbf{e}_{h,\mathrm{RB}}(\mathbf{x}, \theta_k)$$

where $\mathbf{e}_h(\mathbf{x}, \theta_k) = \mathbf{u}(\mathbf{x}, \theta_k) - \mathbf{u}_h(\mathbf{x}, \theta_k)$ accounts for the error introduced by the finite element approximation of the problem and $\mathbf{e}_{h,\mathrm{RB}}(\mathbf{x}, \theta_k) = \mathbf{u}_h(\mathbf{x}, \theta_k) - \mathbf{u}_{h,\mathrm{RB}}(\mathbf{x}, \theta_k)$ accounts for the error introduced by approximating the finite element problem using a reduced basis. The error e_h is reduced by refining the finite element mesh, while reducing the error $\mathbf{e}_{h,\mathrm{RB}}$ involves properly enriching the reduced basis so that it is representative enough. Here, it is assumed that a sufficiently refined finite element mesh is given, so that only the reduced basis error has to be controlled. Therefore, no error estimates are made to enrich the finite element mesh. The error estimates given here concentrate only on deciding whether the reduced basis has to be enriched so that $\mathbf{e}_{h,\mathrm{RB}}$ is kept under a prescribed given tolerance. Specifically, the goal is to control the error in the quantity of interest $e_{\mathrm{RB}}^Q = \ell^Q(\mathbf{e}_{h,\mathrm{RB}})$. Alternatively, one may also include a posterior error estimates controlling the spatial error, see references [31, 32, 33, 34, 35, 36, 37].

In order to assess the reduced basis error $\mathbf{e}_{h,\mathrm{RB}}(\mathbf{x}, \theta_k)$ which in vector form is

$$\mathbf{E}_{\mathrm{RB}}(\theta_k) = \mathbf{U}(\theta_k) - \mathbf{U}_{\mathrm{RB}}(\theta_k) \tag{18}$$

the discretized form of the adjoint problem (4) has to be introduced. Let

$$\mathbf{v}(\mathbf{x}, \theta_k) \approx \mathbf{v}_h(\mathbf{x}, \theta_k) = \sum_{i=1}^{N_{\mathrm{FE}}} \mathbf{v}_i(\theta_k) N_i(\mathbf{x}), \tag{19}$$

and introduce the global vector of unknowns $\mathbf{V}(\theta_k) = [\mathbf{v}_1(\theta_k)^T\, \mathbf{v}_2(\theta_k)^T\, \ldots\, \mathbf{v}_{N_{\mathrm{FE}}}(\theta_k)^T]^T$. Then, the corresponding discretized form of (4) can be written as a linear system of equations.

$$\mathbb{K}(\theta_k)\mathbf{V}(\theta_k) = \mathbf{G}, \tag{20}$$

where $\mathbf{G}$ is the nodal vector associated to the QoI functional $\ell^Q(\cdot)$. Note that the vector $\mathbf{G}$ can also be used to evaluate the value of the QoI for any function $\mathbf{w} \in \mathcal{U}_h$. In particular, the finite element approximations of $Q(\theta_k)$ can be evaluated as

$$Q_h(\theta_k) = \mathbf{G}^T \mathbf{U}(\theta_k) \quad , \quad Q_{\mathrm{RB}}(\theta_k) = \mathbf{G}^T \mathbf{U}_{\mathrm{RB}}(\theta_k),$$

and the error in the quantity of interest due to the introduction of a reduced basis is

$$e_{\mathrm{RB}}^Q(\theta_k) = Q_h(\theta_k) - Q_{\mathrm{RB}}(\theta_k) = \ell^Q(\mathbf{e}_{h,\mathrm{RB}}) = \mathbf{G}^T \mathbf{E}_{\mathrm{RB}}(\theta_k). \tag{21}$$

From this expression, an alternative representation for the error in the quantity of interest $e_{\mathrm{RB}}^Q(\theta_k)$ is obtained using equations (21) and (20), the symmetry of the stiffness matrix $\mathbb{K}$ and equations (18) and (10)

$$\begin{aligned} e_{\mathrm{RB}}^Q(\theta_k) &= \mathbf{G}^T \mathbf{E}_{\mathrm{RB}}(\theta_k) = \mathbf{V}(\theta_k)^T \mathbb{K}(\theta_k)\mathbf{E}_{\mathrm{RB}}(\theta_k) \\ &= \mathbf{V}(\theta_k)^T (\mathbf{F} - \mathbb{K}(\theta_k)\mathbf{U}_{\mathrm{RB}}(\theta_k)) = \mathbf{V}(\theta_k)^T \mathbf{R}_{\mathrm{RB}}(\theta_k), \end{aligned} \tag{22}$$

where $\mathbf{R}_{\mathrm{RB}}(\theta_k)$ is identified as the residual of (10) associated with the reduced basis approximation.

This error expression cannot be used in this form because it requires computing $\mathbf{V}(\theta_k)$ for each realization θ_k, thus involving the solution of a large number of linear system of equations of the same size of the original finite element problem. The idea introduced in [16] to overcome this problem consists in approximating $e_{\mathrm{RB}}^Q(\theta_k)$ by replacing the exact finite element solution of the adjoint problem in (22) by its mean value $\mathbf{V}_0^T$. Namely

$$e_{\mathrm{RB}}^Q(\theta_k) \approx \mathbf{V}_0^T \mathbf{R}_{\mathrm{RB}}(\theta_k) = \eta_{\mathrm{MEAN}}^Q(\theta_k). \tag{23}$$

Using this approach, the cost of evaluating an error estimate for $e_{\mathrm{RB}}^Q(\theta_k)$ is reduced to compute the residual of its corresponding reduced basis approximation. In [16], it is shown that this approximation is precise for problems where the variability in the adjoint problem is small. This error estimate is going to be represented in the following by η_{MEAN}^Q, where the subscript MEAN clearly indicates that the estimate is obtained using the mean value of the adjoint problem.

In this work, a more elaborate approach is used. Instead of approximating the adjoint solution $\mathbf{V}(\theta_k)$ by its mean value $\mathbf{V}_0^T$, a reduced basis approach is used. That is, a collection of $N_{\mathrm{RB}}^{\mathrm{Ad}}$ linearly independent solutions of the adjoint problem $\{\mathbf{v}_{h(1)}, \mathbf{v}_{h(2)}, \ldots, \mathbf{v}_{h(N_{\mathrm{RB}}^{\mathrm{Ad}})}\}$ described by the corresponding matrix of nodal values $\mathbb{V}_{\mathrm{RB}} = [\mathbf{V}_1\, \mathbf{V}_2\, \ldots\, \mathbf{V}_{N_{\mathrm{RB}}^{\mathrm{Ad}}}]$, and the adjoint

solution $\mathbf{V}(\theta_k)$ is approximated by

$$\mathbf{V}_{\mathrm{RB}}(\theta_k) = \mathbb{V}_{\mathrm{RB}}\,\mathbf{d}(\theta_k)$$

where the reduced basis coefficients $\mathbf{d}(\theta_k)$ are computed solving the adjoint reduced basis system of equations

$$(\mathbb{V}_{\mathrm{RB}}^T\mathbb{K}(\theta_k)\mathbb{V}_{\mathrm{RB}})\,\mathbf{d}(\theta_k) = \mathbb{V}_{\mathrm{RB}}^T\mathbf{G}. \tag{24}$$

Using this reduced basis approach also for the adjoint problem yields the new estimate for the reduced basis error in the quantity of interest

$$e_{\mathrm{RB}}^Q(\theta_k) \approx (\mathbf{V}_{\mathrm{RB}}(\theta_k))^T\mathbf{R}_{\mathrm{RB}}(\theta_k) = \eta_{\mathrm{DB}}^Q(\theta_k). \tag{25}$$

Note that this approach does not substantially increase the computational cost, since the cost of computing the reduced basis approximation of the adjoint problem is comparable to the cost of computing the reduced basis finite element approximation of the direct problem. This error estimate, represented in the following by η_{DB}^Q is going to be named after improved or *double base* error estimate, the DB in the subscript representing the double base denomination.

2.3. Adaptative Algorithm

The goal of the adaptive algorithm presented here is, given a large number of realizations of θ, namely $\{\theta_k\}_{k=1,\ldots,N_{\mathrm{MC}}}$, return the corresponding series of approximated quantities of interest $\{Q_{\mathrm{RB}}(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$, where the error (with respect to the full finite element computation) in these approximations is below a certain given prescribed accuracy ϵ_0, that is

$$|e_{\mathrm{RB}}^Q(\theta_k)| = |Q_h(\theta_k) - Q_{\mathrm{RB}}(\theta_k)| < \epsilon_0 \quad \forall k = 1, \ldots, N_{\mathrm{MC}}. \tag{26}$$

Note that low values of ϵ_0 will require more snapshots in the reduced basis approximation of the direct problem, and therefore the cost of the adaptive algorithm is strongly related to the prescribed accuracy.

Guaranteeing the quality of the approximations of the QoI using the full finite element error $e_{\mathrm{RB}}^Q(\theta_k)$ implies computing the full finite element adjoint problem (20) for each realization of θ_k which is computationally unaffordable, therefore, as mentioned in the previous section, the previous accuracy requirements are in practice substituted by ensuring that

$$|\eta_{\mathrm{DB}}^Q(\theta_k)| = |(\mathbf{V}_{\mathrm{RB}}(\theta_k))^T\mathbf{R}_{\mathrm{RB}}(\theta_k)| < \epsilon_0 \quad \forall k = 1, \ldots, N_{\mathrm{MC}}. \tag{27}$$

However, this modified criteria only ensures a good quality for the approximation $Q_{\mathrm{RB}}(\theta_k)$ if $\mathbf{V}_{\mathrm{RB}}(\theta_k)$ is a good approximation of $\mathbf{V}(\theta_k)$. Therefore, to use (27) as an acceptance criteria for $Q_{\mathrm{RB}}(\theta_k)$, the adaptive algorithm has to enforce a good quality of the adjoint reduced basis approximation. The quality of the reduced basis approximation of the adjoint problem can be measured using the residual

$$\mathbf{R}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k) = \mathbf{G} - \mathbb{K}(\theta_k)\mathbf{V}_{\mathrm{RB}}(\theta_k),$$

since equation (20) ensures that the residual is zero for $\mathbf{V}_{\mathrm{RB}}(\theta_k) = \mathbf{V}(\theta_k)$. Here the quality of the adjoint approximation is measured using the scalar value.

$$\eta_{\mathrm{DB}}^{\mathrm{Ad}} = (\mathbf{U}_{\mathrm{RB}}(\theta_k))^T\mathbf{R}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k). \tag{28}$$

Remark 1
Equation (22) is indeed a representation of the error introduced by the reduced basis approximation involving the dual solution and the direct residual. In the case of homogeneous Dirichlet boundary conditions this can also be written in terms of the direct solution and the dual residual, namely

$$e_{\mathrm{RB}}^Q(\theta_k) \quad = \mathbf{V}(\theta_k)^T\mathbf{R}_{\mathrm{RB}}(\theta_k) \quad = \mathbf{U}(\theta_k)^T\mathbf{R}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k). \tag{29}$$

Actually, estimate (28) is obtained replacing the exact finite element solution of the direct problem $\mathbf{U}(\theta_k)$ by its reduced basis approximation $\mathbf{U}_{\mathrm{RB}}(\theta_k)$. It is worth noting that the estimate would provide good approximations of the error in the quantity of interest if: 1) $\mathbf{U}_{\mathrm{RB}}(\theta_k)$ is a good approximation of $\mathbf{U}(\theta_k)$ and 2) if $\mathbf{U}_{\mathrm{RB}}(\theta_k)$ cannot be exactly described using the reduced basis for the adjoint problem, since in this case the orthogonality condition of the adjoint residual yields $\eta_{\mathrm{DB}}^{\mathrm{Ad}} = 0$ even though the error $e_{\mathrm{RB}}^Q(\theta_k)$ may not be zero, thus producing an underestimation of the error. As shown in the numerical examples presented in the forthcoming sections, the direct and adjoint basis are not too different and, consequently, this non-desired behavior is not appreciated in the results,. The estimate $\eta_{\mathrm{DB}}^{\mathrm{Ad}}$ is therefore performing well in these examples. The possibility of using more accurate and robust error estimates for the adjoint problem may bring some advantages in some cases and is to be addressed in forthcoming works.

Therefore, the problem of computing $\{Q_h(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$ which involves calling N_{MC} times the direct problem solver associated to the full finite element mesh, is replaced to compute the approximations $\{Q_{\mathrm{RB}}(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$. This approximation only involves solving N_{MC} small versions of the direct problem (its reduced basis projection) along with the solution of N_{MC} small versions of the adjoint problem to ensure the quality of the approximations for the quantity of interest. The direct and adjoint reduced bases are initially set to one snapshot and are enhanced as the adaptive algorithm advance using the two error estimates η_{DB}^Q and $\eta_{\mathrm{DB}}^{\mathrm{Ad}}$ for the direct and adjoint problem respectively. The adaptation of the reduced basis as the adaptive algorithm is performed allows ensuring the quality of the approximated QoI, $Q_{\mathrm{RB}}(\theta_k)$, while ensuring a minimal size of the snapshot basis for the direct and adjoint problems.

The proposed adaptive algorithm is detailed hereafter in Algorithm 1.

Data: ϵ_0, the realizations $\{\theta_k\}_{k=1,\ldots,N_{\mathrm{MC}}}$, and the routines generating $\mathbb{K}(\theta_k)$, $\mathbf{F}$ and $\mathbf{G}$
Result: A number of N_{MC} realizations of the QoI, $\{Q_{\mathrm{RB}}(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$, with error lower than ϵ_0
First Monte Carlo throw ($k = 1$)
generate $\mathbb{K}(\theta_1)$, $\mathbf{F}$, $\mathbf{G}$;
compute $\mathbf{U}_1$ solving $\mathbb{K}(\theta_1)\mathbf{U}_1 = \mathbf{F}$ and $\mathbf{V}_1$ solving $\mathbb{K}(\theta_1)\mathbf{V}_1 = \mathbf{G}$;
compute and store $Q_1 = \mathbf{G}^{\mathbf{T}}\mathbf{U_1}$;
initialize $\mathbb{U}_{\mathrm{RB}} = [\mathbf{U}_1]$ and $N_{\mathrm{RB}} = 1$;
initialize $\mathbb{V}_{\mathrm{RB}} = [\mathbf{V}_1]$ and $N_{\mathrm{RB}}^{\mathrm{Ad}} = 1$;
Subsequent Monte Carlo throws
for $k = 2 \ldots N_{\mathrm{MC}}$ **do**
    generate $\mathbb{K}(\theta_k)$;
    compute $\mathbb{K}_{\mathrm{RB}}(\theta_k) := \mathbb{U}_{\mathrm{RB}}^T\mathbb{K}(\theta_k)\mathbb{U}_{\mathrm{RB}}$ and $\mathbf{F}_{\mathrm{RB}} := \mathbb{U}_{\mathrm{RB}}^T\mathbf{F}$;
    compute $\mathbb{K}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k) := \mathbb{V}_{\mathrm{RB}}^T\mathbb{K}(\theta_k)\mathbb{V}_{\mathrm{RB}}$ and $\mathbf{G}_{\mathrm{RB}} := \mathbb{V}_{\mathrm{RB}}^T\mathbf{G}$;
    solve the direct and adjoint reduced basis problems
    compute $\mathbf{a}$ solving $\mathbb{K}_{\mathrm{RB}}(\theta_k)\mathbf{a} = \mathbf{F}_{\mathrm{RB}}$;
    compute $\mathbf{d}$ solving $\mathbb{K}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k)\mathbf{d} = \mathbf{G}_{\mathrm{RB}}$;
    compute $\mathbf{U}_{\mathrm{RB}}^k = \mathbb{U}_{\mathrm{RB}}\mathbf{a}$ and $\mathbf{V}_{\mathrm{RB}}^k = \mathbb{V}_{\mathrm{RB}}\mathbf{d}$;
    assess the error of the reduced basis solutions
    compute the residuals $\mathbf{R}_{\mathrm{RB}} = \mathbf{F} - \mathbb{K}(\theta_k)\mathbf{U}_{\mathrm{RB}}^k$ and $\mathbf{R}_{\mathrm{RB}}^{\mathrm{Ad}} = \mathbf{G} - \mathbb{K}(\theta_k)\mathbf{V}_{\mathrm{RB}}^k$;
    compute the error estimates $\eta_{\mathrm{DB}}^Q = (\mathbf{V}_{\mathrm{RB}}^k)^T\mathbf{R}_{\mathrm{RB}}$ and $\eta_{\mathrm{DB}}^{\mathrm{Ad}} = (\mathbf{U}_{\mathrm{RB}}^k)^T\mathbf{R}_{\mathrm{RB}}^{\mathrm{Ad}}$;
    if $|\eta_{\mathrm{DB}}^Q| > \epsilon_0$ *and* $|\eta_{\mathrm{DB}}^{\mathrm{Ad}}| > \epsilon_0$ **then**
        solution not admissible: re-computing the full solutions and enriching the two basis
        compute $\mathbf{U}_k$ solving $\mathbb{K}(\theta_k)\mathbf{U}_k = \mathbf{F}$;
        set $\mathbf{U}_{\mathrm{RB}}^k = \mathbf{U}_k$;
        update $\mathbb{U}_{\mathrm{RB}} = \begin{bmatrix}\mathbf{U}_1 \cdots \mathbf{U}_{N_{\mathrm{RB}}} \mathbf{U}_k\end{bmatrix}$ and $N_{\mathrm{RB}} \leftarrow N_{\mathrm{RB}} + 1$;
        compute $\mathbf{V}_k$ solving $\mathbb{K}(\theta_k)\mathbf{V}_k = \mathbf{G}$;
        update $\mathbb{V}_{\mathrm{RB}} = \begin{bmatrix}\mathbf{V}_1 \cdots \mathbf{V}_{N_{\mathrm{RB}}} \mathbf{V}_k\end{bmatrix}$ and $N_{\mathrm{RB}}^{\mathrm{Ad}} \leftarrow N_{\mathrm{RB}}^{\mathrm{Ad}} + 1$;
    else if $|\eta_{\mathrm{DB}}^Q| > \epsilon_0$ *and* $|\eta_{\mathrm{DB}}^{\mathrm{Ad}}| < \epsilon_0$ **then**
        solution not admissible: re-computing the direct full solution and enriching $\mathbb{U}_{\mathrm{RB}}$
        compute $\mathbf{U}_k$ solving $\mathbb{K}(\theta_k)\mathbf{U}_k = \mathbf{F}$;
        set $\mathbf{U}_{\mathrm{RB}}^k = \mathbf{U}_k$;
        update $\mathbb{U}_{\mathrm{RB}} = \begin{bmatrix}\mathbf{U}_1 \cdots \mathbf{U}_{N_{\mathrm{RB}}} \mathbf{U}_k\end{bmatrix}$ and $N_{\mathrm{RB}} \leftarrow N_{\mathrm{RB}} + 1$;
    else if $|\eta_{\mathrm{DB}}^Q| < \epsilon_0$ *and* $|\eta_{\mathrm{DB}}^{\mathrm{Ad}}| > \epsilon_0$ **then**
        solution not admissible: re-computing the adjoint full solution and enriching $\mathbb{V}_{\mathrm{RB}}$
        compute $\mathbf{V}_k$ solving $\mathbb{K}(\theta_k)\mathbf{V}_k = \mathbf{G}$;
        update $\mathbb{V}_{\mathrm{RB}} = \begin{bmatrix}\mathbf{V}_1 \cdots \mathbf{V}_{N_{\mathrm{RB}}} \mathbf{V}_k\end{bmatrix}$ and $N_{\mathrm{RB}}^{\mathrm{Ad}} \leftarrow N_{\mathrm{RB}}^{\mathrm{Ad}} + 1$;
    else
        admissible solution: no extra computations required;
    end
    compute and store $Q_{\mathrm{RB}}(\theta_k) = \mathbf{G}^T\mathbf{U}_{\mathrm{RB}}^k$;
end

Algorithm 1: Adaptive reduced basis solver with double base update.

2.4. Numerical results

The performance of the goal oriented adaptive procedure given in Algorithm 1 is compared to the approach developed in [16], for the 2D test case presented in section 1.4. The difference between these two approaches is that, in the new approach, the acceptance criterion for an approximation of the quantity of interest $Q(\theta_k)$ is driven by the *double base* error estimate, i.e. $|\eta_{\mathrm{DB}}^Q(\theta_k)| < \epsilon_0$, while in [16] the mean error estimate (also named after *classic* error estimate) is used, i.e. $|\eta_{\mathrm{MEAN}}^Q(\theta_k)| < \epsilon_0$. In other words, the error estimate $e_{\mathrm{RB}}^Q(\theta_k)$ given in equation (22) is approximated replacing the exact finite element approximation of the adjoint problem $\mathbf{V}(\theta_k)$ by its mean $\mathbf{V}_0$ in the classic approach, while in the double base approach, $\mathbf{V}(\theta_k)$ is replaced by a sufficiently accurate reduced basis approximation $\mathbf{V}_{\mathrm{RB}}(\theta)$.

At the end of the goal-oriented Monte-Carlo simulation, both strategies provide approximations of the quantities of interest $\{Q_{\mathrm{RB}}(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$ approximating the exact finite element values $\{Q_h(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$, with associated errors $\{e_{\mathrm{RB}}^Q(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$ which are expected to be below the prescribed error threshold ϵ_0. To verify the accuracy of the proposed error estimates when used in the acceptance criterion, i.e. to verify that the estimates correctly enforce the prescribed tolerance ϵ_0,

the index

$$E_{\max}^Q = \frac{\max_{k \in 1..N_{\mathrm{MC}}} |e_{\mathrm{RB}}^Q(\theta_k)|}{\varepsilon_0}$$

is introduced. Note that $E_{\max}^Q$ is the ratio between the maximum error committed in all the Monte-Carlo simulations and the prescribed maximum error. Ideally, one would wish for $E_{\max}^Q$ to be less than or equal to one, i.e. $E_{\max}^Q \leq 1$, since in this case all the approximations $\{Q_h(\theta_k)\}_{k=1,...,N_{\mathrm{MC}}}$ would have a certified maximum error below the threshold ϵ_0. However, since the proposed estimates $|\eta_{\mathrm{DB}}^Q(\theta_k)|$ and $|\eta_{\mathrm{MEAN}}^Q(\theta_k)|$ are not a guaranteed error estimates for $|e_{\mathrm{RB}}^Q(\theta_k)|$, $E_{\max}^Q$ may exceed one and the accuracy of the goal-oriented adaptive algorithms will be measured by the surplus of $E_{\max}^Q$ from 1.

Figure 3 shows the obtained $E_{\max}^Q$ values, both for the classical and double base techniques, when varying the prescribed error ϵ_0 and for different values of the standard deviation α. In this case, the underlying finite element computations are performed with a mesh with $n_{\mathrm{dof}} = 12879$ (or equivalently $n_{\mathrm{L}} = 80$). The results obtained using underlying finite element meshes of $n_{\mathrm{dof}} = 819$ ($n_{\mathrm{L}} = 20$) and $n_{\mathrm{dof}} = 3239$ ($n_{\mathrm{L}} = 40$) are not detailed here since the obtained results are similar to the ones reported in figure 3.
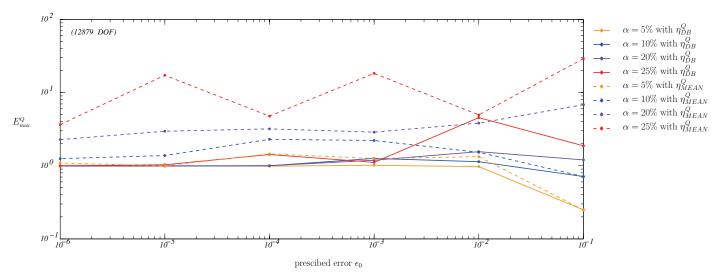

Figure 3. Accuracy of the error estimates when varying both ϵ_0 and α. Computations performed using the underlying finite element mesh of $n_{\mathrm{dof}} = 12879$ ($n_{\mathrm{L}} = 80$).

It can be seen that for low values of the standard deviation, α ranging from 5% to 10%, both methods properly impose the prescribed error. For all the values $\epsilon_0 \in [10^{-6}; 10^{-1}]$ and for the three different computational meshes, the value of $E_{\max}^Q$ remains close to one. However, for larger values of the standard deviation, the index $E_{\max}^Q$ only remains close to one for the double basis approach. This is due to the fact that the hypothesis introduced in equation (23) that the adjoint problem can be approximated by its mean value is only reasonable for low values of the standard deviation α. Changing the mean value of the adjoint problem by a reduced basis approximation to estimate the error as in equation (25) allows using this methodology also for high values of α, by just adjusting the number of snapshots used in the adjoint reduced basis approximation. It will be seen in section 2.4 that, this gain in the accuracy is in detriment of the computational cost of the method, but that this increase in the cost remains still very competitive.

Figure 4 shows the distribution of errors e_{RB}^Q for $\epsilon_0 = 10^{-3}$, $\alpha = 5\%$ and $n_{dof} = 3239$. The distribution obtained using both the classical and double base approaches, again show that better estimates are obtained when the adjoint problem is approximated using the reduced basis approach. Indeed, the double base error distribution is more concentrated around zero. It can also be seen that, for this particular case, the double base approach guarantees that $|e_{\mathrm{RB}}^Q| < \epsilon_0$ for all the occurrences ($E_{\max}^Q = 0.6 \leq 1$). On the other hand, even though for the classical approach $E_{\max}^Q = 1.1 > 1$, the histogram shows that only very few Monte-Carlo throws are not meeting the error requirements. In fact, less that 1% of the occurrences do not meet this criteria.

Therefore, even though figures 3 show that, for some values of ϵ_0 and n_{dof}, the error is not strictly enforced in all the Monte-Carlo throws ($E_{\max}^Q > 1$), the number of occurrences not meeting the error requirements are a very small fraction of the occurrences.

3. OPTIMAL IMPLEMENTATION AND COMPUTATIONAL ASPECTS

This section provides some computational details of the presented techniques. A poor implementation of the adaptive reduced basis techniques presented above may paradoxically yield to a method with higher cost than a classic Monte-Carlo simulation, or at most achieve a suboptimal CPU gain with respect to it. Therefore, an efficient implementation is crucial to yield the desired CPU gains.

The core ingredient of reduced basis techniques is to solve small systems of equations (17) rather than the original larger ones (10), which a-priori are cheaper. However, several aspects have to be carefully dealt with in order to yield an optimal CPU time:
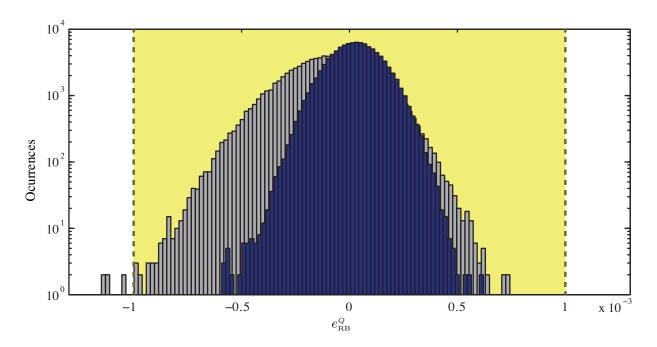
Figure 4. Histogram showing the occurrences of e_{RB}^Q (in semi-log) scale for the classic approach (in grey) and the developed double basis method (in blue) for $\epsilon_0 = 10^{-3}$, $\alpha = 5\%$ and $n_{dof} = 3239$. The yellow region shows the occurrences ensuring $|e_{\mathrm{RB}}^Q| < \epsilon_0$.

- the matrices $\mathbb{K}(\theta_k)$ associated to the original systems of equations are sparse matrices while the associated reduced ones $\mathbb{K}_{\mathbb{RB}}(\theta_k)$ are full matrices,
- the time to generate matrices $\mathbb{K}$ and $\mathbb{K}_{RB}$ used is Algorithm 1 can be large compared with the time that takes to solve the reduced systems (17),
- the need to introduce error estimates (and the computations they require like the residual computation) can yield a non-neglegible computational increase.

These aspects are discussed in this section and specific techniques are provided to efficiently implement the reduced basis technique. It is worth noting that these computational aspects are a major improvement, not only for the adaptive reduced basis solver with double base update described in algorithm 1 but also for the same algorithm when only the *classical* error estimate based on the mean adjoint solution is used. Thus, this strategy is also bringing an important improvement to the results presented in [16], in terms of computational its cost.

3.1. Efficient computation of the stiffness matrices (offline precomputation)

As mentioned before, even though the solution of the reduced basis direct (resp. adjoint) problem given in equation (17) (resp. (24)) is really fast since it involves solving a $N_{\mathrm{RB}} \times N_{\mathrm{RB}}$ (resp. $N_{\mathrm{RB}}^{\mathrm{Ad}} \times N_{\mathrm{RB}}^{\mathrm{Ad}}$) linear system of equations. However, the assembling of the reduced stiffness matrix $\mathbb{K}_{\mathbb{RB}}(\theta_k)$ (resp. $\mathbb{K}_{\mathbb{RB}}^{\mathrm{Ad}}(\theta_k)$ requires assembling the global matrix $\mathbb{K}(\theta_k)$ and then projecting it to the reduced basis space $\mathbb{U}_{\mathbb{RB}}$ (resp. $\mathbb{V}_{\mathbb{RB}}$). When the cost in CPU time of the algorithm is carefully examined, this assembly procedure, that has to be done for each throw and both for the direct and adjoint problem, has a significant cost and therefore it has to be efficiently implemented.

3.1.1. Computation of $\mathbb{K}(\theta_k)$ The non-intrusive approach summarized in Algorithm 1 requires generating $\mathbb{K}(\theta_k)$ for all the Monte-Carlo throws $\{\theta_k\}_{k=1,...,N_{\mathrm{MC}}}$. Each realization θ_k is associated to the Hooke operator $\mathbf{K}(\mathbf{x}, \theta_k)$ given by the truncated Karhunen-Loeve decomposition (6) and therefore it is possible to generate the stiffness matrix $\mathbb{K}(\theta_k)$ from $\mathbf{K}(\mathbf{x}, \theta_k)$ using a standard finite element assembly procedure. This approach requires to make a global assembly operation for each throw θ_k. In order to optimize the CPU cost, it is crucial to use the linearity of the bilinear form of the problem with respect to the Hooke tensor and use formula (11) to compute $\mathbb{K}(\theta_k)$.

In the offline stage where the Young's modulus random field is discretized using a truncated Karhunen-Loeve decomposition, the eigenvectors $\{E_i(\mathbf{x})\}_{i=1,...,N_{\mathrm{KL}}}$ of the covariance function are obtained. These functions defining the different Young modules in turn define the Hooke tensors $\{\mathbf{K}_i(\mathbf{x})\}_{i=1,...,N_{\mathrm{KL}}}$ and its finite element global assembled stiffness matrices $\{\mathbb{K}_i\}_{i=1,...,N_{\mathrm{KL}}}$. These global stiffness matrices are calculated and stored in the offline stage. Note that in general N_{KL} is a small number, and therefore the cost of storing the matrices $\mathbb{K}_i$ remains reasonable as far as N_{KL} remains reasonable. Then, during the online Monte-Carlo stage, the stiffness matrix $\mathbb{K}(\theta_k)$ associated with each throw is generated using (11) as a simple linear combination of the pre-computed global stiffness matrices.

3.1.2. Computation of $\mathbb{K}_{\mathrm{RB}}(\theta_k)$ and $\mathbb{K}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k)$ Using a reduced basis approach to compute the direct and adjoint computation requires also computing the reduced basis matrices $\mathbb{K}_{\mathrm{RB}}(\theta_k)$ and $\mathbb{K}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k)$ for each throw $\{\theta_k\}_{k=1,\ldots,N_{\mathrm{MC}}}$. These matrices can be obtained from the global stiffness matrix $\mathbb{K}(\theta_k)$ as

$$\mathbb{K}_{\mathrm{RB}}(\theta_k) = \mathbb{U}_{\mathrm{RB}}^T \mathbb{K}(\theta_k)\mathbb{U}_{\mathrm{RB}} \quad \text{and} \quad \mathbb{K}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k) = \mathbb{V}_{\mathrm{RB}}^T \mathbb{K}(\theta_k)\mathbb{V}_{\mathrm{RB}}.$$

However, from a computational point of view it is preferable to take advantage of the Karhunen-Loeve expansion (11) to compute the reduced basis matrix. Namely, precomputing the reduced stiffness matrices $\mathbb{K}_{\mathrm{RB},i}$ and $\mathbb{K}_{\mathrm{RB},i}^{\mathrm{Ad}}$ associated with each of the Karhunen-Loeve modes (eigenvectors)

$$\mathbb{K}_{\mathrm{RB},i} = \mathbb{U}_{\mathrm{RB}}^T \mathbb{K}_i \mathbb{U}_{\mathrm{RB}} \quad \text{and} \quad \mathbb{K}_{\mathrm{RB},i}^{\mathrm{Ad}} = \mathbb{V}_{\mathrm{RB}}^T \mathbb{K}_i \mathbb{V}_{\mathrm{RB}} \tag{30}$$

in the off-line stage, allows readily computing the reduced basis stiffness matrices for each throw by just using a linear combination of the pre-computed matrices as

$$\mathbb{K}_{\mathrm{RB}}(\theta_k) = \sum_{i=0}^{N_{\mathrm{KL}}} \sqrt{\lambda_i}\mathbb{K}_{\mathrm{RB},i}\xi_i(\theta) \quad \text{and} \quad \mathbb{K}_{\mathrm{RB}}^{\mathrm{Ad}}(\theta_k) = \sum_{i=0}^{N_{\mathrm{KL}}} \sqrt{\lambda_i}\mathbb{K}_{\mathrm{RB},i}^{\mathrm{Ad}}\xi_i(\theta). \tag{31}$$

where we have set $\lambda_0 = 1$ and $\xi_0(\theta) = 1$ to simplify the previous expressions. The computation of the reduced basis stiffness matrices using (31) is computationally inexpensive, and it only requires the storage of the pre-computed reduced stiffness direct and adjoint for each Karhunen-Loeve modes. As mentioned before, since usually N_{KL} is not large, this cost in memory remains reasonable in general.

Remark 2
For problems including random terms in the source and/or tractions terms $\mathbf{f}_d(\mathbf{x},\theta)$ and $\mathbf{g}_d(\mathbf{x},\theta)$ respectively, the load vector $\mathbf{F}$ also depends on θ, namely $\mathbf{F}(\theta)$. An optimal computation of the reduced basis load vector for each throw $\mathbf{F}_{\mathrm{RB}}(\theta_k) = \mathbb{U}_{\mathrm{RB}}^T \mathbf{F}(\theta_k)$ also requires preforming a Karhunen-Loeve decomposition of the loads $\mathbf{f}_d(\mathbf{x},\theta)$ and $\mathbf{g}_d(\mathbf{x},\theta)$ which allow computing $\mathbf{F}_{\mathrm{RB}}(\theta_k)$ as a linear combination of the pre-computed vectors $\mathbf{F}_{\mathrm{RB},i}$ associated to the eigenvectors of the loads of the problem respectively.

3.1.3. Efficient Reduced Basis growing When the quality of the reduced basis approximations is not deemed sufficient, the size of the reduced basis is increased ($N_{\mathrm{RB}} \leftarrow N_{\mathrm{RB}} + 1$ in Algorithm 1). The enhancement of the basis is done both for the direct and adjoint problem. Here the details of this growing procedure is given only for the direct problem but the same applies to the adjoint problem.

Increasing the basis for the direct problem reduces to add a vector $\widetilde{\mathbf{U}}$ in the last column of matrix $\mathbb{U}_{\mathrm{RB}}$ to obtain the new basis $\widetilde{\mathbb{U}}_{\mathrm{RB}}$,

$$\widetilde{\mathbb{U}}_{\mathrm{RB}} = [\mathbb{U}_{\mathrm{RB}}, \widetilde{\mathbf{U}}]. \tag{32}$$

Since the matrices $\{\mathbb{K}_{\mathrm{RB},i}\}_{i=1,\ldots,N_{\mathrm{KL}}}$ are a projection of $\mathbb{K}_i$ onto $\mathbb{U}_{\mathrm{RB}}$ these matrices have to be recomputed each time a new snapshot is introduced in $\mathbb{U}_{\mathrm{RB}}$.

A simple, but computationally expensive, procedure consists in generating the new set of N_{KL} matrices from scratch by replacing $\mathbb{U}_{\mathrm{RB}}$ by $\widetilde{\mathbb{U}}_{\mathrm{RB}}$ in (30). However, a simple alternative procedure can be used to optimize the CPU cost of computing the reduced basis matrices $\{\widetilde{\mathbb{K}}_{\mathrm{RB},i}\}_{i=1,\ldots,N_{\mathrm{KL}}}$ by using the already available information stored in $\{\mathbb{K}_{\mathrm{RB},i}\}_{i=1,\ldots,N_{\mathrm{KL}}}$. Indeed, taking advantage of the block form of matrix $\widetilde{\mathbb{K}}_{\mathrm{RB},i}$ allows rewriting the $(N_{\mathrm{RB}}+1) \times (N_{\mathrm{RB}}+1)$ matrix $\widetilde{\mathbb{K}}_{\mathrm{RB},i}$ as

$$\widetilde{\mathbb{K}}_{\mathrm{RB},i} = \widetilde{\mathbb{U}}_{\mathrm{RB}}^T \mathbb{K}_i \widetilde{\mathbb{U}}_{\mathrm{RB}} = \left[\begin{array}{c} \mathbb{U}_{\mathrm{RB}}^T \\ \widetilde{\mathbf{U}}^T \end{array} \right] \mathbb{K}_i [\mathbb{U}_{\mathrm{RB}}, \widetilde{\mathbf{U}}] = \left[\begin{array}{cc} \mathbb{U}_{\mathrm{RB}}^T \mathbb{K}_i \mathbb{U}_{\mathrm{RB}} & \mathbb{U}_{\mathrm{RB}}^T \mathbb{K}_i \widetilde{\mathbf{U}} \\ \widetilde{\mathbf{U}}^T \mathbb{K}_i \mathbb{U}_{\mathrm{RB}} & \widetilde{\mathbf{U}}^T \mathbb{K}_i \widetilde{\mathbf{U}} \end{array} \right] = \left[\begin{array}{cc} \mathbb{K}_{\mathrm{RB},i} & \mathbb{X}_i^T \widetilde{\mathbf{U}} \\ \widetilde{\mathbf{U}}^T \mathbb{X}_i & c_i \end{array} \right].$$

Therefore the computation of the enhanced matrices $\widetilde{\mathbb{K}}_{\mathrm{RB},i}$ requires only computing the matrices $\mathbb{X}_i = \mathbb{K}_i \mathbb{U}_{\mathrm{RB}}$ and then do the products involved in $\widetilde{\mathbf{U}}^T \mathbb{X}_i$ and $c_i = \widetilde{\mathbf{U}}^T \mathbb{K}_i \widetilde{\mathbf{U}}$. In principle, the computation of the matrices $\mathbb{X}_i$ require a matrix times matrix operations. However, if no memory restrictions apply, these matrices can be stored and then, for each enhancement, the computation of the new matrix $\widetilde{\mathbb{X}}_i$ is done just performing a matrix times vector operation (of size $d \cdot N_{\mathrm{FE}}$)

$$\widetilde{\mathbb{X}}_i = \mathbb{K}_i \widetilde{\mathbb{U}}_{\mathrm{RB}} = [\mathbb{X}_i, \mathbb{K}_i \widetilde{\mathbf{U}}]. \tag{33}$$

The computation of the scalar value c_i is more demanding since it involves two matrix times vector operations of size $d \cdot N_{\mathrm{FE}}$ which cannot be simplified by storing previously computed information.

3.2. Efficient evaluating of the goal oriented error estimates η_{MEAN}^{Q} and η_{DB}^{Q}

3.2.1. Mean estimation: η_{MEAN}^{Q} The Karhunen-Loeve decomposition of the stiffness matrix $\mathbb{K}(\theta_k)$ can also be used in this context to alleviate the computational cost of computing $\eta_{\text{MEAN}}^{Q}(\theta_k)$ for each throw. Indeed, expanding equation (23) and using (11) yields

$$
\begin{aligned}
\eta_{\text{MEAN}}^{Q}(\theta_k) \quad &= \mathbf{V}_0^T \mathbf{R}_{\text{RB}}(\theta_k) = \mathbf{V}_0^T \mathbf{F} - \mathbf{V}_0^T \mathbb{K}(\theta_k)\mathbb{U}_{\text{RB}}\mathbf{a} = \mathbf{V}_0^T \mathbf{F} - \left(\sum_{i=0}^{N_{\text{KL}}} \sqrt{\lambda_i}\xi_i(\theta) \, \mathbf{V}_0^T \mathbb{K}_i \mathbb{U}_{\text{RB}} \right) \mathbf{a} \\
&= \mathbf{V}_0^T \mathbf{F} - \left(\sum_{i=0}^{N_{\text{KL}}} \sqrt{\lambda_i}\xi_i(\theta) \, \mathbf{V}_0^T \mathbb{X}_i \right) \mathbf{a},
\end{aligned}
$$

where again we have set $\lambda_0 = 1$ and $\xi_0(\theta) = 1$. Then, the error estimate η_{MEAN}^{Q} can be efficiently computed by storing the scalar value $\mathbf{V}_0^T \mathbf{F}$ and the vectors $\{\mathbf{V}_0^T \mathbb{X}_i\}_{i=1,\dots,N_{\text{KL}}}$. Each evaluation of the mean estimate requires then to compute a proper linear combination of the stored vectors and a vector times vector computation of size $d \cdot N_{\text{FE}}$.

Of course, each time a new snapshot is introduced in $\mathbb{U}_{\text{RB}}$, the vectors $\mathbf{V}_0^T \mathbb{X}_i$ have to be recomputed. However, once the low cost enrichment strategy proposed in the previous section is used to update $\widetilde{\mathbb{X}}_i$, the new vectors are updated as

$$
\mathbf{V}_0^T \widetilde{\mathbb{X}}_i = [\mathbf{V}_0^T \mathbb{X}_i \, , \, \mathbf{V}_0^T \mathbb{K}_i \widetilde{\mathbf{U}}],
$$

where the computation of the new (last) scalar position of the vectors $\mathbf{V}_0^T \mathbb{K}_i \widetilde{\mathbf{U}}$ only require a vector times vector operation since the products $\mathbb{K}_i \widetilde{\mathbf{U}}$ have already been computed in (33).

3.2.2. Double basis estimation: η_{DB}^{Q} The double base estimate $\eta_{\text{DB}}^{Q}(\theta_k)$ can also be computed efficiently by storing the small $N_{\text{KL}} \times N_{\text{KL}}$ matrices

$$
\mathbb{K}_{\text{DB},i} = \mathbb{V}_{\text{RB}}^T \mathbb{K}_i \mathbb{U}_{\text{RB}} \quad i = 1, \dots, N_{\text{KL}}. \tag{34}
$$

Indeed expanding (25) and using (11) yields

$$
\begin{aligned}
\eta_{\text{DB}}^{Q}(\theta_k) \quad &= (\mathbf{V}_{\text{RB}}(\theta_k))^T \mathbf{R}_{\text{RB}}(\theta_k) = \mathbf{d}^T \mathbb{V}_{\text{RB}}^T \mathbf{F} - \mathbf{d}^T \mathbb{V}_{\text{RB}}^T \mathbb{K}(\theta_k)\mathbb{U}_{\text{RB}}\mathbf{a} = \mathbf{d}^T \mathbb{V}_{\text{RB}}^T \mathbf{F} - \left(\sum_{i=0}^{N_{\text{KL}}} \sqrt{\lambda_i}\xi_i(\theta) \, \mathbf{d}^T \mathbb{V}_{\text{RB}}^T \mathbb{K}_i \mathbb{U}_{\text{RB}} \right) \mathbf{a} \\
&= \mathbf{d}^T \mathbb{V}_{\text{RB}}^T \mathbf{F} - \left(\sum_{i=0}^{N_{\text{KL}}} \sqrt{\lambda_i}\xi_i(\theta) \, \mathbf{d}^T \mathbb{K}_{\text{DB},i} \right) \mathbf{a}.
\end{aligned}
$$

Obviously, these matrices have to be recomputed each time that either the direct or adjoint reduced basis is enhanced. However, an enrichment strategy similar to those used in the previous section can be used to drastically decrease the computational complexity. Indeed,

$$
\widetilde{\mathbb{K}}_{\text{DB},i} = \widetilde{\mathbb{V}}_{\text{RB}}^T \mathbb{K}_i \widetilde{\mathbb{U}}_{\text{RB}} = \left[\begin{array}{c} \mathbb{V}_{\text{RB}}^T \\ \widetilde{\mathbf{V}}^T \end{array} \right] \mathbb{K}_i [\mathbb{U}_{\text{RB}} \, , \, \widetilde{\mathbf{U}}] = \left[\begin{array}{cc} \mathbb{V}_{\text{RB}}^T \mathbb{K}_i \mathbb{U}_{\text{RB}} & \mathbb{V}_{\text{RB}}^T \mathbb{K}_i \widetilde{\mathbf{U}} \\ \widetilde{\mathbf{V}}^T \mathbb{K}_i \mathbb{U}_{\text{RB}} & \widetilde{\mathbf{V}}^T \mathbb{K}_i \widetilde{\mathbf{U}} \end{array} \right] = \left[\begin{array}{cc} \mathbb{K}_{\text{DB},i} & (\mathbb{X}_i^{\text{Ad}})^T \widetilde{\mathbf{U}} \\ \widetilde{\mathbf{V}}^T \mathbb{X}_i & \hat{c}_i \end{array} \right],
$$

where the matrices $\mathbb{X}_i = \mathbb{K}_i \mathbb{U}_{\text{RB}}$ and $\mathbb{X}_i^{\text{Ad}} = \mathbb{K}_i \mathbb{V}_{\text{RB}}$ are stored in the process of enlarging the stiffness matrices $\widetilde{\mathbb{K}}_{\text{RB},i}$ and $\widetilde{\mathbb{K}}_{\text{RB},i}^{\text{Ad}}$ respectively.

It goes without saying that if just one of the basis is enhanced, the equivalent simplified formulas are used (for an enhancement of the direct and adjoint basis respectively)

$$
\widetilde{\mathbb{K}}_{\text{DB},i} = \left[\mathbb{K}_{\text{DB},i} \, , \, (\mathbb{X}_i^{\text{Ad}})^T \widetilde{\mathbf{U}} \right] \quad , \quad \widetilde{\mathbb{K}}_{\text{DB},i} = \left[\begin{array}{c} \mathbb{K}_{\text{DB},i} \\ \widetilde{\mathbf{V}}^T \mathbb{X}_i \end{array} \right].
$$

3.3. Global browsing with minimal snapshot basis

In [16] an adaptive algorithm for reduced basis enrichment is proposed. The basic idea is to successively generate the reduced basis along the Monte Carlo simulation process, increasing the number of elements of the direct and adjoint reduced basis when it is required. The major inconvenience of this approach is that, once the reduced basis is enriched, the cost of computing every subsequent throw is increased despite the fact that maybe the new solutions may be precisely described using a smaller basis. Therefore, the cost of this approach clearly depends on the ordering of the realizations $\{\theta_k\}_{k=1,\dots,N_{\text{MC}}}$. Optimally, one should order the realizations starting from those whose accuracy requirements can be met with a lower number of terms on the reduced basis expansion and ending with the more complex realizations that need all the snapshots to be properly described.

To overcome this drawback a new strategy is proposed relying on Algorithm 1. The main idea is that given a reduced basis, all the realizations which meet the accuracy requirements with this basis are handled and removed from the list of remaining realizations to be dealt with. After this step is finished, the information provided by the current reduced basis is not sufficient to properly deal with the remaining realizations and the reduced basis is enriched. The algorithm continues dealing with all the realizations that can be precisely computed with this new basis. The procedure ends when there are no more realizations to run.

The key point of this new approach is that it optimizes the use of each intermediate reduced basis (from the starting basis with only one snapshot to the full basis with N_{RB} snapshot), since at each step it solves the maximum number of realizations which verity the given error criteria.

The algorithm associated with this procedure is detailed hereafter in Algorithm 2. The acceptance error criteria for the quantities of interest $Q_{\text{RB}}(\theta_k)$ is assessed here using the classical mean approach, that is, the estimate $\eta_{\text{MEAN}}^{Q}(\theta_k)$ is used to approximate $e_{\text{RB}}^{Q}(\theta_k)$ for each throw. Therefore the reduced basis method is only used in the direct problem, since the exact finite element solution of the adjoint problem $\mathbf{V}(\theta_k)$ is replaced by its mean value $\mathbf{V}_0$ which is computed in the full finite element mesh before starting the Monte-Carlo simulation. Future works, will include also a minimal snapshot approach including the improved error estimate $\eta_{\text{DB}}^{Q}(\theta_k)$ which requires to optimally enhance both the direct and adjoint reduced basis.

Data: ϵ_0, the realizations $\{\theta_k\}_{k=1,\ldots,N_{\text{MC}}}$, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{V_0}$, $\mathbb{U}_{\text{RB}} = [\mathbf{U}_1]$, $N_{\text{RB}} = 1$,
  $\{\mathbb{K}_i\,,\ \mathbb{K}_{\text{RB},i} = \mathbb{U}_{\text{RB}}^T \mathbb{K}_i \mathbb{U}_{\text{RB}}\,,\ \mathbb{X}_i = \mathbb{K}_i \mathbb{U}_{\text{RB}}\,,\ \mathbf{V}_0^T \mathbb{X}_i\}_{i=1,\ldots,N_{\text{KL}}}$ and the routine generating $\mathbb{K}(\theta_k)$
Result: A number of N_{MC} realizations of the QoI, $\{Q_{\text{RB}}(\theta_k)\}_{k=1,\ldots,N_{\text{MC}}}$, with approximated error lower than ϵ_0
while *all the throws are not solved* **do**
    for *all the not-solved throws* **do**
        compute $\mathbb{K}_{\text{RB}}(\theta_k) = \sum_{i=0}^{N_{\text{KL}}} \sqrt{\lambda_i} \mathbb{K}_{\text{RB},i} \xi_i(\theta)$;
        compute $\mathbf{F}_{\text{RB}} = \mathbb{U}_{\text{RB}}^T \mathbf{F}$;
        compute $\mathbf{a}$ solving $\mathbb{K}_{\text{RB}}(\theta_k)\mathbf{a} = \mathbf{F}_{\text{RB}}$;
        assess the error of the reduced basis solution
        compute the error estimate $\eta_{\text{MEAN}}^{Q}(\theta_k) = \mathbf{V}_0^T \mathbf{F} - \left(\sum_{i=0}^{N_{\text{KL}}} \sqrt{\lambda_i} \xi_i(\theta)\, \mathbf{V}_0^T \mathbb{X}_i \right) \mathbf{a}$
        if $|\eta_{\text{MEAN}}^{Q}(\theta_k)| > \epsilon_0$ **then**
            solution not admissible;
            tag the throw as not solved
        else
            solution admissible;
            tag the throw as solved;
            compute $\mathbf{U}_{\text{RB}}^k = \mathbb{U}_{\text{RB}} \mathbf{a}$;
            compute and store $Q_{\text{RB}}(\theta_k) = \mathbf{G}^T \mathbf{U}_{\text{RB}}^k$;
        end
    end
    select one off the not-solved throw and compute the full solution associated to this throw compute $\mathbb{K}(\theta_k)$;
    compute $\mathbf{U}_k$ solving $\mathbb{K}(\theta_k)\mathbf{U}_k = \mathbf{F}$;
    compute and store $Q_{\text{RB}}(\theta_k) = \mathbf{G}^T \mathbf{U}_{\text{RB}}^k$;
    tag the throw as solved;
    enrich the basis
    update $\mathbb{U}_{\text{RB}} = [\mathbf{U}_1 \cdots \mathbf{U}_{N_{\text{RB}}} \mathbf{U}_k]$ and $N_{\text{RB}} \leftarrow N_{\text{RB}} + 1$;
    for $i = 1, \ldots, N_{\text{KL}}$ **do**
        update $\mathbf{V}_0^T \mathbb{X}_i \leftarrow [\mathbf{V}_0^T \mathbb{X}_i\,,\ \mathbf{V}_0^T \mathbb{K}_i \mathbf{U}_k]$;
        update $\mathbb{X}_i \leftarrow [\mathbb{X}_i\,,\ \mathbb{K}_i \mathbf{U}_k]$;
        update $\mathbb{K}_{\text{RB},i} \leftarrow \begin{bmatrix} \mathbb{K}_{\text{RB},i} & \mathbb{X}_i^T \mathbf{U}_k \\ \mathbf{U}_k^T \mathbb{X}_i & \mathbf{U}_k^T \mathbb{K}_i \mathbf{U}_k \end{bmatrix}$;
    end
end

Algorithm 2: Adaptive reduced basis solver (efficient browsing of the throws) with reduced basis update only for the direct problem.

It may seem that this approach is more computationally demanding than the approach with successively enrich reduced basis, since for each basis all the unresolved throws are computed, and therefore, the direct problem associated to a given realization θ_k is solved as many times as necessary to meet the precision requirements. However, note that the problems are really small (for instance, for the initial step $\mathbb{U}_{\text{RB}} = [\mathbf{U}_1]$ where all the throws are computed, the direct problem consists of a simple division). Moreover, if no memory restrictions apply, the optimization techniques introduced for the basis enrichment allows performing the full Monte-Carlo simulation in very reasonable computational times. Additionally, the main advantage of this new algorithm is that it can be completely parallelized.

4. NUMERICAL ILLUSTRATION OF THE COMPUTATIONAL COST OF THE METHOD

4.1. Improvement in the computational cost

The proposed algorithms have been implemented in C++, using the Eigen library (FER numerical modelling software [38]). The Eigen library allows using a sparse storage for problems with a large number of degrees of freedom while using linear solvers involving a Cholesky decomposition of the form $\mathbb{L}\mathbb{D}\mathbb{L}^T$. The linear solver is the same for dense matrices (matrices involved in the reduced basis problems) and for sparse matrices (matrices involved in the resolution of the full finite element direct and adjoint problems). It is therefore well suited to use in the proposed algorithms since it is applicable for both dense and sparse linear systems, allowing to easily perform comparisons between the different CPU computational times.

A study of the computational time gain, achieved using the proposed method, is given in the following. A similar study is presented in [16] where the study of the evolution of the size of the reduced basis for the direct problem as the Monte-Carlo simulation advances shows the potential of the therein proposed method (classical/mean approach). The size of the reduced basis remains small compared to the size of the full finite element problem, and therefore a substantial gain in the computational cost is observed. However, in [16] it is already observed that this gain in cost is only achieved if special attention is paid in using an efficient implementation of all the computations. That is, it is critical that the reduced basis strategy is used along with techniques like the ones presented in section 3 to optimize the computational cost. Specifically, the goal-oriented reduced-basis Monte-Carlo simulation requires constructing, solving and verifying a large number of small dense linear systems of equations. A non-optimal implementation of these three steps can greatly compromise the efficiency of the proposed methods.

As in section 2.4, a comparison of the performance of the goal oriented adaptive procedure given in Algorithm 1 is compared to the approach developed in [16], for the 2D numerical example described in section 1.4. The comparison presented here, however, focuses on its computational efficiency.

A Monte-Carlo simulation with $N_{\mathrm{MC}} = 10^5$ is performed for three different values of the deviation parameter $\alpha = 5\%, 10\%$ and 20%, the five different fixed finite element meshes (table I) and for six different values of the prescribed error ϵ_0 (from $\epsilon_0 = 10^{-1}$ to $\epsilon_0 = 10^{-6}$). The computational cost of the two methods is measured with the aid of the ratio R_{CPU}

$$R_{\mathrm{CPU}} = \frac{t_{\mathrm{RB}}}{t_{\mathrm{MC}}}, \tag{35}$$

which divides the computational time of the Monte-Carlo analysis using the reduced basis approach to solve the finite element problems (t_{RB}) by the computational time of the Monte-Carlo analysis using the full finite element computations (t_{MC}) given in table III. Note that, to simplify the notation, the time t_{RB} both refers to the computational time of the reduced basis approach where the adjoint problem is approximated by either its mean or by using the reduced basis approach. Obviously, one expects $R_{\mathrm{CPU}} << 1$ since it means that the gain in cost when using a reduced basis approach is considerable. Also, it is expected that $R_{\mathrm{CPU}}^{\mathrm{MEAN}} < R_{\mathrm{CPU}}^{\mathrm{DB}}$ since the first approach (classic/mean) does not require dealing with a reduced base for the adjoint problem (double base approach).

Figure 5 shows the computational times of the classical and double base approaches, highlighting the benefits of both approaches in terms of computational cost. From the figures, the following major conclusions can be extracted:

- For almost all computations, the computational time of the classical/mean approach is smaller than the computational time of the double base approach. However, this difference is not very significative and decreases as the reference finite element mesh is refined. Recalling that the double base approach provides much better accuracy than the classical approach, the extra computational effort is worth it.

- For a fixed value of the desired precision ϵ_0, the ratios R_{CPU} decrease as the number of degrees of freedom increase. This is due to the fact that the number of snapshots needed in the reduced basis approximations is hardly sensitive to the mesh refinements. As the mesh is refined, the computational cost of the full finite element approximations required in the full Monte-Carlo approach is greatly increased whereas the size of the small linear systems that have to be solved to obtain the reduced basis approximations remain nearly constant. Therefore, since the ratio between the size of the full finite element system of equations and the size of the reduced basis systems substantially increase, the ration R_{CPU} rapidly decreases.

- For a fixed reference finite element problem, a demanding prescribed error ϵ_0 yields less gain in the computational cost of the method. Of course this is due to the fact that in order to ensure small prescribed errors, the number of terms in the reduced basis approximations needs to be larger. Obviously, as the number of the reduced basis snapshots increases, the reduced basis approach becomes less appealing and more computationally demanding.

- Finally, a larger deviation parameter α also yields less computational gains. As in the case where the prescribed error ϵ_0 is small, in general a large deviation parameter α imply a larger variability between the solutions associated with different random realizations θ_k and therefore more number of snapshots are needed to properly describe both the direct and adjoint approximations using a reduced basis approach.

Consequently, the results show that the proposed double-base approach perfectly meets the expectations in terms of CPU time gain. Only one somewhat unrealistic configuration, combining both a coarse reference finite element mesh and a very small prescribed error, yields larger computational times than the full Monte-Carlo simulation.
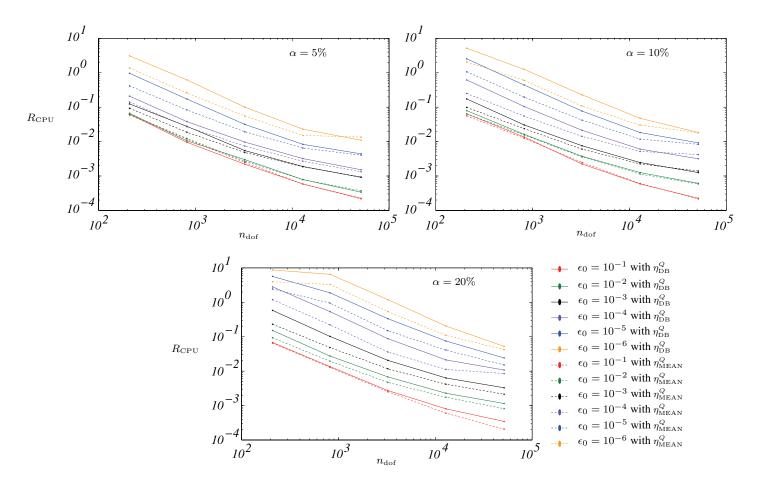
Figure 5. Computational gain of the classical and double base approaches versus the full Monte-Carlo simulation (R_{CPU}) for $\alpha = 5\%, 10\%$ and 20%, the finite element meshes referred in table I and various prescribed errors ϵ_0.

4.2. Computational gain when using a global browsing with minimal snapshot basis

This section presents a comparison between the computational cost of the global browsing reduced basis adaptive strategy presented in Algorithm 2, and its sequentially computed equivalent version. That is, the global browsing technique is compared to the *improved* adaptive strategy presented in [16] based on classical estimate (η_{MEAN}^Q), where the term *improved* refers to the fact that the technique introduced in [16] is now optimally implemented using the previously discussed computational aspects.

The approach presented in algorithm 2 guarantees that, for each constructed intermediate reduced basis, all the realizations which meet the accuracy requirements with this basis are handled and therefore removed from the list of remaining realizations to be dealt with. Consequently, each realization is computed using a minimum size reduced basis. Although the final size of the reduced basis remains unchanged, the computational cost of the intermediate realization is greatly reduced.

The computational gain introduced by the global browsing is measured introducing the CPU time ratio

$$R_{\mathrm{algo}} = \frac{t_{\mathrm{RB}}^0}{t_{\mathrm{RB}}^2}, \tag{36}$$

where t_{RB}^2 is the CPU time associated with algorithm 2 and t_{RB}^0 is the CPU time associated with its sequential version. Note that the desired scenario is that the global browsing alleviates the computational cost and therefore $t_{\mathrm{RB}}^2 \leq t_{\mathrm{RB}}^0$ which implies $R_{\mathrm{algo}} \geq 1$. Thus, a larger computational gain is associated with large values of R_{algo}.

Figure 6 shows the obtained computational quotients R_{algo} for $\alpha = 5\%$ and $\alpha = 10\%$. As can be seen, the global browsing strategy is advantageous for low values of the prescribed errors ϵ_0 and low values of the deviation parameter. As the error requirements become more restrictive, the finite element approximations have to be solved more precisely and therefore the number of needed snapshots in the reduced basis becomes more problem dependent. In this case, the cost of global browsing becomes larger since using less snapshots in some of the occurrences does not compensate the fact that most of the occurrences are solved more than once. It can also be observed that the gain when using the global browsing is also increased as the reference finite element mesh is refined.

Finally, even though a direct gain in the computational time is not obtained in all the cases, a crucial feature of the global browsing method described in algorithm 2 is that the method is a perfect candidate for parallelization, just like in a classic Monte Carlo simulation. Indeed, for a given fixed reduced basis, all the realizations that have to be computed and tested for accuracy are
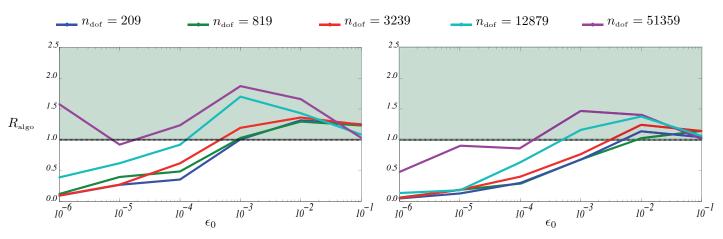
Figure 6. Comparison of the computational cost of the global browsing strategy versus the standard incremental adaptive procedure (R_{algo}) for $\alpha = 5\%$ (left) and $\alpha = 10\%$ (right), the finite element meshes referred in table I and various prescribed errors ϵ_0. The green region shows the computations for which the global browsing is advantageous.

completely independent and should be run in parallel. Only after all the realizations have been computed, the results have to be regrouped to enrich the reduced basis and then throw all the remaining occurrences again in a parallel form.

Therefore, the global browsing, although in some cases slightly degrades the computational cost of the standard reduced basis approach, has the advantage of being completely parallelized.

4.3. Robustness of the reduced basis techniques with respect to ordering

In this section, the influence of the ordering of the realizations that have to be computed is studied. Note that while in a full Monte Carlo simulation the order of the realizations that have to be computed has no effect on the computational time of the method since all the computations are mutually independent, when using a reduced basis technique with no global browsing, the ordering of the realizations can *a priori* be important since the reduced basis is enriched as the finite element computations associated to each realization cannot be properly solved using the previous computations. Once the basis is enriched, the subsequent computations become costlier regardless of its complexity.

To study this influence, a Monte-Carlo simulation with $N_{\mathrm{MC}} = 100.000$ is performed for a the deviation parameter $\alpha = 5\%$, the finite element mesh with $n_{\mathrm{dof}} = 819$ and for two prescribed errors $\epsilon_0 = 10^{-4}$ and $\epsilon_0 = 10^{-6}$. As mentioned above, the classical mean approach and the double basis approach present very close results for low values of the standard deviation (e.g. $\alpha = 5\%$ as in this case, where further enriching the adjoint base does not bring a significant improvement). Thus, the robustness of the method is only checked here for the improved version of the algorithm presented in [16].

The independence assumption on the random variables $\xi_i(\theta)$ appearing in the Karhunen-Loeve decomposition of the Young's Modulus $E(\mathbf{x}, \theta)$, see (14), allows obtaining the N_{MC} realizations defined by $\{E(\mathbf{x}, \theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$, by just generating a normally distributed $N_{\mathrm{MC}} \times N_{\mathrm{KL}}$ random matrix, denoted in the following by $\mathbf{M}_{N(0,1)}$, and compute the N_{MC} values of $E(\mathbf{x}, \theta_k)$ using the values of each row of $\mathbf{M}_{N(0,1)}$ in equation (15) along with (14). Recall that in this test case N_{KL} is set to 20.

Four different sorting strategies are tested before using the values $\{E(\mathbf{x}, \theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$ (resp. $\{\mathbb{K}(\theta_k)\}_{k=1,\ldots,N_{\mathrm{MC}}}$) in the Monte Carlo simulation. In order to set the different sorting strategies, first a set of functions

$$\mathbf{E}(\mathbf{x}) = [\mathbf{E_1}(\mathbf{x}), \mathbf{E_2}(\mathbf{x}), \ldots, \mathbf{E_{N_{\mathrm{MC}}}}(\mathbf{x})]$$

is computed as detailed above using the values in the rows of $\mathbf{M}_{N(0,1)}$ in equations (15) and (14). The four different sorting strategies are then:

- Sorting 1: take the values of the Young's modulus as initially obtained $\mathbf{E}(\mathbf{x})$
- Sorting 2: randomly reorder the functions in $\mathbf{E}(\mathbf{x})$
- Sorting 3: sort $\mathbf{E}(\mathbf{x})$ by its associated value $\xi_1(\theta)$ (first column of $\mathbf{M}_{N(0,1)}$) in increasing order
- Sorting 4: sort $\mathbf{E}(\mathbf{x})$ by its associated value $\xi_1(\theta)$ (taking absolute values) in increasing order

Figure 7 shows that the final size of the basis and the profile of its enrichment are only slightly affected by the different sorting strategies, both for the prescribed errors $\epsilon_0 = 10^{-4}$ and $\epsilon_0 = 10^{-6}$. This illustrates the robustness of the method because the size of the basis is mainly depending on the overall complexity of the problem to be treated (prescribed errors, quantity of interest, number of realizations, size of the stochastic space...), and the influence of sorting the sampling is nearly negligible in the presented numerical results. In practice, the difference of the results due to the various sorting strategies of the realizations act as a small noise source.
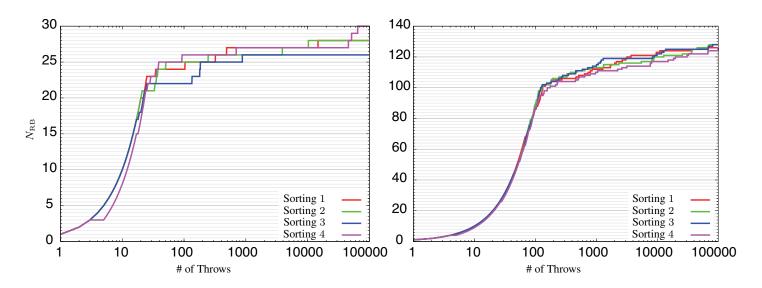
Figure 7. Number of terms N_{RB} included in the reduced basis for the direct problem as the Monte Carlo simulation advances, for the prescribed value of the final error $\epsilon_0 = 10^{-4}$ (left) and $\epsilon_0 = 10^{-6}$ (right).

5. A 3D NUMERICAL EXAMPLE

Here, a 3D numerical example is presented to confirm the good performance of the presented techniques when applied to finite element problems of significant size. As shown in figure 8, the goal is to predict the mechanical behavior of a 3D cubic structure of side $L = 10$ formed by a heterogeneous material. The domain is composed of a 75% matrix while the remaining 25% of the volume corresponds to 40 spherical inclusions, of random sizes and positions. The four lateral faces and the base of the cube are fixed while a uniform pressure $P = 20$ is applied to a small square centered on the top of the cube (with sides parallel to those of the edges of the cube), of size $l = 2$. Finally, the quantity of interest examined here is the vertical displacement of the node placed in the center of the top of the cube (corresponding to a punctual force for the adjoint problem).
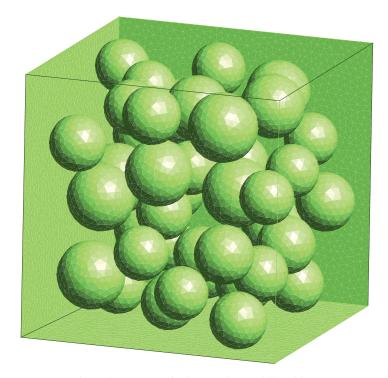

Figure 8. 3D numerical example: model problem.

The randomness of the problem is given by having 41 different materials each varying independently around a prescribed mean value. Specifically, denoting by ω_0 the domain occupied by the matrix and by $\omega_i, i = 1, \dots, 40$ the different subdomains of the

spherical inclusions, the value of the Young's modulus in each domain ω_i is of the form:

$$E^i(\xi_i(\theta)) = E_0^i(1 + \alpha_i\xi_i(\theta)). \tag{37}$$

where $\xi_i(\theta)$ are independent and identically distributed random variables. Note that, in this case, Young's modulus can be rewritten as

$$E(\mathbf{x}, \theta) = \sum_{i=0}^{40} 1_{\omega_i}(\mathbf{x})E^i(\xi_i(\theta)) = \sum_{i=0}^{40} 1_{\omega_i}(\mathbf{x})E_0^i + \sum_{i=0}^{40} \alpha_i 1_{\omega_i}(\mathbf{x})\xi_i(\theta), \tag{38}$$

where $E_0 = \sum_{i=0}^{40} 1_{\omega_i}(\mathbf{x})E_0^i$ is the mean spatial distribution. Therefore, the input random field $E(\mathbf{x}, \theta)$ is already given by a linear combination of 41 independent random variables $\{\xi_i(\theta)\}_{i=0,\dots,40}$, and hence there is no need to use the Karhunen-Loeve decomposition to discretize the input random field.

In particular, the parameters associated to the random field $E(\mathbf{x}, \theta)$ for this problem are: $E_0^0 = 1$ for the matrix, $E_i^0 = 2$ for all the spheres and a standard deviation of $\alpha_i = 5\%$ for all the subdomain including the cube matrix. The independent random variables $\{\xi_i(\theta)\}_{i=0\dots,N_{\mathrm{KL}}}$ are defined using the non-standard nonlinear distribution given in equation (15).

The statistics of the quantity of interest are approximated using a Monte-Carlo simulation with $N_{\mathrm{MC}} = 100.000$ realizations for three different finite element meshes formed of linear tetrahedral elements with $n_{\mathrm{dof}} = 50746$, $n_{\mathrm{dof}} = 86493$ and $n_{\mathrm{dof}} = 276736$. The results presented herein are computed using both the improved version of the algorithm presented in [16] based on a mean approach and the double base adaptive algorithm presented in algorithm 1. Is is worth noting that in this 3D example where the computational requirements are demanding, the new proposed implementation is crucial to obtain the desired results in a competitive computational time.

The approximated PDFs of $Q(\theta)$ for the finer mesh with $n_{\mathrm{dof}} = 276736$ characterized by the mean and variance values

$$\mathbb{E}_{\mathrm{MEAN}}[Q(\theta)] \approx 0.987731 \quad \text{and} \quad \mathbb{V}_{\mathrm{MEAN}}[Q(\theta)] \approx 0.001536$$

for the mean approach and

$$\mathbb{E}_{\mathrm{DB}}[Q(\theta)] \approx 0.98755 \quad \text{and} \quad \mathbb{V}_{\mathrm{DB}}[Q(\theta)] \approx 0.00153$$

for the double base approach are given in figure 9. Actually, figure 9 shows the histogram of the occurrences of $Q(\theta)$. The approximated PDF of $Q(\theta)$ is obtained from this histogram by just scaling the ordinate axis (dividing by $N_{\mathrm{MC}} = 10^5$). As can be
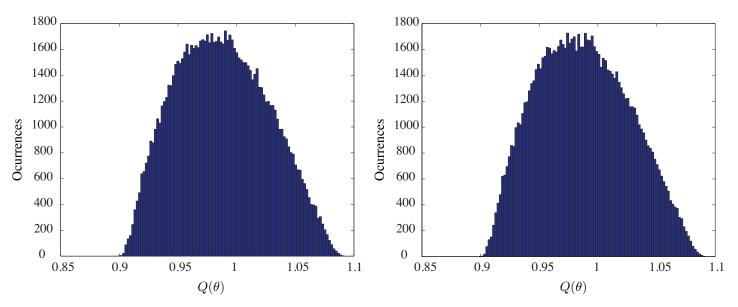

Figure 9. 3D numerical example: histogram showing the occurrences of $Q(\theta)$ for the mesh with $n_{\mathrm{dof}} = 276736$ for the mean approach (left) and the new double base approach (right).

seen, since the considered deviations are small ($\alpha_i = 5\%$) the use of the reduced basis only for the primal problem or for both the primal and adjoint problem yields similar results.

This example is representative of the crucial advance that the present work provides in the context of goal-oriented stochastic finite element adaptive strategies. Indeed, the resolution of the complete direct problem takes 42 minutes for the finest mesh, and hence a full analysis of the stochastic behavior of the QoI would take 8 years of a single-processor computation.

Table IV shows the computational times associated with strategies presented in this work for the three different finite element meshes and for different values of the prescribed error ϵ_0. It is worth noting that, of course, the Full Monte-Carlo simulations have not been performed, and therefore the CPU times regarding the Full Monte-Carlo simulations detailed in table IV have been estimated by extrapolation (multiplying the CPU time spent in solving 100 complete direct problems by 1000 to obtain the CPU

cost of $N_{\mathrm{MC}} = 10^5$). The final number of snapshots in the reduced basis N_{RB} and $N_{\mathrm{RB}}^{\mathrm{Ad}}$ are also reported in the table for the double base approach given in algorithm 1.

n_{dof}		Double base				Mean	
		$\mathrm{Time}^{\mathrm{DB}}$	$R_{\mathrm{CPU}}^{\mathrm{DB}}$	N_{RB}	$N_{\mathrm{RB}}^{\mathrm{Ad}}$	$\mathrm{Time}^{\mathrm{MEAN}}$	$R_{\mathrm{CPU}}^{\mathrm{MEAN}}$
50746	Full Monte-Carlo	4 100 000					
	RB $\epsilon_0 =10\%$	96	$2.34 10^{-5}$	1	1	55	$1.34 10^{-5}$
	RB $\epsilon_0 =1\%$	141	$3.43 10^{-5}$	1	2	55	$1.34 10^{-5}$
	RB $\epsilon_0 =0.1\%$	1 289	$3.14 10^{-4}$	7	9	277	$6.75 10^{-5}$
	RB $\epsilon_0 =0.01\%$	7932	$1.93 10^{-3}$	43	43	1 896	$4.62 10^{-4}$
	RB $\epsilon_0 =0.001\%$	17 570	$4.29 10^{-3}$	73	65	5 124	$1.25 10^{-3}$
86493	Full Monte-Carlo	$1.098 10^7$					
	RB $\epsilon_0 =10\%$	226	$2.06 10^{-5}$	1	1	124	$1.13 10^{-5}$
	RB $\epsilon_0 =1\%$	340	$3.10 10^{-5}$	1	2	124	$1.13 10^{-5}$
	RB $\epsilon_0 =0.1\%$	1 363	$1.24 10^{-5}$	8	9	346	$3.14 10^{-5}$
	RB $\epsilon_0 =0.01\%$	8 368	$7.62 10^{-4}$	43	43	1 965	$1.79 10^{-4}$
	RB $\epsilon_0 =0.001\%$	19 624	$1.79 10^{-3}$	79	56	5 193	$4.72 10^{-4}$
276736	Full Monte-Carlo	$2.5323 10^8$					
	RB $\epsilon_0 =10\%$	5126	$2.02 10^{-5}$	1	1	2 594	$1.02 10^{-5}$
	RB $\epsilon_0 =1\%$	7 264	$2.87 10^{-5}$	1	2	2 594	$1.02 10^{-5}$
	RB $\epsilon_0 =0.1\%$	55 644	$2.20 10^{-4}$	7	9	14 881	$5.87 10^{-5}$
	RB $\epsilon_0 =0.01\%$	301 120	$1.19 10^{-3}$	43	44	89 152	$3.52 10^{-4}$

Table IV. Time (in seconds) to solve the 3D example for the full Monte-Carlo simulation and the Reduced basis (RB) adaptive procedure for different meshes and values of the prescribed error using algorithm 1 both for the double base and mean approach. The sizes N_{RB} and $N_{\mathrm{RB}}^{\mathrm{Ad}}$ of the double base approach are also reported.

The results confirm that the new procedure allows drastically reducing the computational cost of the Monte-Carlo simulation. Moreover, it is observed that the improvement in the R_{CPU} cost increases as the size of the problem increases (larger n_{dof}), but that this gain is less important for demanding accuracy requirements (small values of ϵ_0). However, in all the cases, the proposed methodology allows obtaining good statistics of the quantity of interest in a reasonable computational time.

6. CONCLUSIONS

In this paper a method based on an adaptive reduced basis strategy has been presented in the framework of linear elasticity. A complete analysis of the method is given including a new enhanced goal-oriented error estimator and estimates of CPU cost gain. The new method is an improvement of the method developed in [16] both regarding the accuracy of the results and the computational cost. Indeed, the accuracy improvements are obtained introducing a reduced basis for the adjoint problem which allows introducing a more accurate error estimate which in terms allows better controlling the adaptive algorithm. This makes it possible to reach good level of accuracy for a quantity of interest even for problems with large variations in the stochastic behaviour. On the other hand, the computational cost is reduced introducing an optimal implementation that allows enhancing the CPU cost of the adaptive method: 1) the reduced basis stiffness matrices are computed using pre-computed reduced stiffness for the direct and adjoint problem for each Karhunen-Loeve mode, 2) the reduced basis growing is efficiently implemented providing a very cheap construction of the stiffness matrices, and 3) an efficient browsing of the Monte-Carlo throws is presented.

A parametric analysis of the method is performed regarding cost and quality on a 2D example. The results illustrate the obtained gain in terms of both quality and cost. Also, different simulations are introduced to show the robustness of the method. Finally, the technique is applied to a large 3D mechanical simulation with heterogeneous material where performing a full Monte Carlo simulation is unaffordable (regarding its CPU time). The proposed methodology allows obtaining very accurate statistics for the quantity of interest in reasonable CPU times.

REFERENCES

1. Schuëller GI. A state-of-the-art report on computational stochastic mechanics. *Probabilistic Engineering Mechanics* 1997; **12**(4):197 – 321. A State-of-the-Art Report on Computational Stochastic Mechanics.
2. Nouy A. Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations. *Archives of Computational Methods in Engineering* 2009; **16**:251–285.
3. Stefanou G. The stochastic finite element method: Past, present and future. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(9-12):1031 – 1051.
4. Giraldi L, Litvinenko A, Liu D, Matthies HG, Nouy A. To be or not to be intrusive? The solution of parametric and stochastic equations—the "plain vanilla" Galerkin case. *SIAM J. Sci. Comput.* 2014; **36**(6):A2720–A2744, doi:10.1137/130942802. URL http://dx.doi.org/10.1137/130942802.

5. Ladevèze P, Florentin E. Verification of stochastic models in uncertain environments using the constitutive relation error method. *Comput. Methods Appl. Mech. Engrg.* 2006; **196**(1-3):225–234, doi:10.1016/j.cma.2006.03.006. URL http://dx.doi.org/10.1016/j.cma.2006.03.006.

6. Davison JJ. *Response surface designs and analysis for bi-randomization error structures.* ProQuest LLC, Ann Arbor, MI, 1995. URL http://gateway.proquest.com/openurl?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&res_da thesis (Ph.D.)–Virginia Polytechnic Institute and State University.

7. Zhu X, Narayan A, Xiu D. Computational aspects of stochastic collocation with multifidelity models. *SIAM/ASA J. Uncertain. Quantif.* 2014; **2**(1):444–463, doi:10.1137/130949154. URL http://dx.doi.org/10.1137/130949154.

8. Stavroulakis G, Giovanis DG, Papadrakakis M, Papadopoulos V. A new perspective on the solution of uncertainty quantification and reliability analysis of large-scale problems. *Comput. Methods Appl. Mech. Engrg.* 2014; **276**:627–658, doi:10.1016/j.cma.2014.03.009. URL http://dx.doi.org/10.1016/j.cma.2014.03.009.

9. Boyaval S, Bris CL, Maday Y, Nguyen NC, Patera AT. A reduced basis approach for variational problems with stochastic parameters: Application to heat conduction with variable robin coefficient. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(41-44):3187 – 3206, doi: DOI:10.1016/j.cma.2009.05.019. URL http://www.sciencedirect.com/science/article/pii/S0045782509002114.

10. Sachdeva SK, Nair PB, Keane AJ. Hybridization of stochastic reduced basis methods with polynomial chaos expansions. *Probabilistic Engineering Mechanics* 2006; **21**(2):182 – 192.

11. Grigoriu M. Reduced order models for random functions. application to stochastic problems. *Applied Mathematical Modelling* 2009; **33**(1):161 – 175.

12. Nguyen N. A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. *Journal of Computational Physics* 2008; **227**(23):9807 – 9822.

13. Huynh D, Knezevic D, Patera A. Certified reduced basis model validation: A frequentistic uncertainty framework. *Computer Methods in Applied Mechanics and Engineering* 2012; **201-204**(0):13 – 24.

14. Mohan PS, Nair PB, Keane AJ. Multi-element stochastic reduced basis methods. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(17-18):1495 – 1506.

15. Boyaval S, Bris CL, Lelièvre T, Maday Y, Nguyen NC, Patera AT. Reduced basis techniques for stochastic problems. *Archives of Computational Methods in Engineering* 2010; **17**(4):435–454.

16. Florentin E, Díez P. Adaptive reduced basis strategy based on goal oriented error assessment for stochastic problems. *Computer Methods in Applied Mechanics and Engineering* 2012; **225-228**(0):116 – 127.

17. Bianchini I, Argiento R, Auricchio F, Lanzarone E. Efficient uncertainty quantification in stochastic finite element analysis based on functional principal components. *Computational Mechanics* 2015; **56**(3):533–549.

18. Oden J, Bass J, Westermann T. Development of a post-processor for a posteriori error estimation of quantities of interest. *Internal Report, Texas Institute for Computational and Applied Mathematics* 2001; .

19. Ladevèze P, Chamoin L. Calculation of strict error bounds for finite element approximations of non-linear pointwise quantities of interest. *Internat. J. Numer. Methods Engrg.* 2010; **84**(13):1638–1664, doi:10.1002/nme.2957. URL http://dx.doi.org/10.1002/nme.2957.

20. Loeve M. Fonctions aleatoires du second ordre. *Compte rendu de l'académie des Sciences - Paris 220* 1945; .

21. Karhunen K. Uber lineare methoden in der wahrscheinlichkeitsrechnung. *Amer. Acad. Sci., Fennicade (Translation: RAND Corporation, Santa Monica, California, Rep. T-131, Aug. 1960).* 1947; **37**:3–79.

22. Phoon KK, Huang SP, Quek ST. Simulation of second-order processes using Karhunen-Loeve expansion. *Comput. & Structures* 2002; **80**(12):1049–1060, doi:10.1016/S0045-7949(02)00064-0. URL http://dx.doi.org/10.1016/S0045-7949(02)00064-0.

23. Phoon KK, Huang HW, Quek ST. Simulation of strongly non-gaussian processes using Karhunen-Loeve expansion. *Probabilistic Engineering Mechanics* 2005; **20**(2):188–198, doi:10.1016/j.probengmech.2005.05.007. URL http://www.sciencedirect.com/science/article/pii/S0266892005000123.

24. Babuška I, Tempone R, Zouraris GE. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Comput. Methods Appl. Mech. Engrg.* 2005; **194**(12-16):1251–1294, doi:10.1016/j.cma.2004.02.026. URL http://dx.doi.org/10.1016/j.cma.2004.02.026.

25. Boyaval S, Le Bris C, Maday Y, Nguyen NC, Patera AT. A reduced basis approach for variational problems with stochastic parameters: application to heat conduction with variable Robin coefficient. *Comput. Methods Appl. Mech. Engrg.* 2009; **198**(41-44):3187–3206, doi:10.1016/j.cma.2009.05.019. URL http://dx.doi.org/10.1016/j.cma.2009.05.019.

26. Almeida RC, Oden JT. Solution verification, goal-oriented adaptive methods for stochastic advection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.* 2010; **199**(37-40):2472–2486, doi:10.1016/j.cma.2010.04.001. URL http://dx.doi.org/10.1016/j.cma.2010.04.001.

27. Chen P, Quarteroni A. Accurate and efficient evaluation of failure probability for partial different equations with random input data. *Comput. Methods Appl. Mech. Engrg.* 2013; **267**:233–260, doi:10.1016/j.cma.2013.08.016. URL http://dx.doi.org/10.1016/j.cma.2013.08.016.

28. O'Hagan A. Polynomial chaos: A tutorial and critique from a statistician's perspective. *Technical Report*, University of Sheffield, UK may 2013.

29. Fishman G. *Monte Carlo: Concepts, Algorithms, and Applications.* Springer NY, 1995.

30. Betz W, Papaioannou I, Straub D. Numerical methods for the discretization of random fields by means of the Karhunen-Loève expansion. *Comput. Methods Appl. Mech. Engrg.* 2014; **271**:109–129, doi:10.1016/j.cma.2013.12.010. URL http://dx.doi.org/10.1016/j.cma.2013.12.010.

31. Chamoin L, Díez P ((eds.)). *Verifying calculations—forty years on.* Springer Briefs in Applied Sciences and Technology, Springer, Cham, 2016, doi: 10.1007/978-3-319-20553-3. URL http://dx.doi.org/10.1007/978-3-319-20553-3, An overview of classical verification techniques for FEM simulations.

32. Paraschivoiu M, Peraire J, Patera AT. A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations. *Comput. Methods Appl. Mech. Engrg.* 1997; **150**(1-4):289–312, doi:10.1016/S0045-7825(97)00086-8. URL http://dx.doi.org/10.1016/S0045-7825(97)00086-8, symposium on Advances in Computational Mechanics, Vol. 2 (Austin, TX, 1997).

33. Cirak F, Ramm E. A posteriori error estimation and adaptivity for linear elasticity using the reciprocal theorem. *Comput. Methods Appl. Mech. Engrg.* 1998; **156**(1-4):351–362, doi:10.1016/S0045-7825(97)00220-X. URL http://dx.doi.org/10.1016/S0045-7825(97)00220-X.

34. Parés N, Bonet J, Huerta A, Peraire J. The computation of bounds for linear-functional outputs of weak solutions to the two-dimensional elasticity equations. *Comput. Methods Appl. Mech. Engrg.* 2006; **195**(4-6):406–429, doi:10.1016/j.cma.2004.10.013. URL http://dx.doi.org/10.1016/j.cma.2004.10.013.

35. Ladevèze P, Chamoin L, Florentin E. A new non-intrusive technique for the construction of admissible stress fields in model verification. *Comput. Methods Appl. Mech. Engrg.* 2010; **199**(9-12):766–777, doi:10.1016/j.cma.2009.11.007. URL http://dx.doi.org/10.1016/j.cma.2009.11.007.

36. Verdugo F, Parés N, Díez P. Error assessment in structural transient dynamics. *Archives of Computational Methods in Engineering* 2014; **21**(1):59–90, doi:10.1007/s11831-014-9096-x. URL http://dx.doi.org/10.1007/s11831-014-9096-x.

37. Nadal E, Díez P, Rodenas JJ, Tur M, Fuenmayor FJ. A recovery-explicit error estimator in energy norm for linear elasticity. *Comput. Methods Appl. Mech. Eng.* 2015; **287**:172–190.

38. Feng ZQ, Magnain B, Cros JM. Fer/impact: logiciel de simulation numérique des problèmes d'impact. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 2006; **15**(1-3):175–186.