SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. xx, pp. x

An empirical interpolation and model-variance reduction method for computing statistical outputs of parametrized stochastic partial differential equations

F. Vidal-Codina[†], N. C. Nguyen[†], M. B. Giles[‡], J. Peraire[†]

Abstract. We present an empirical interpolation and model-variance reduction method for the fast and reliable computation of statistical outputs of parametrized stochastic elliptic partial differential equations. Our method consists of three main ingredients: (1) the real-time computation of reduced basis (RB) outputs approximating high-fidelity outputs computed with the hybridizable discontinuous Galerkin (HDG) discretization; (2) the empirical interpolation (EI) for an efficient offline-online decoupling of the parametric and stochastic influence; and (3) a multilevel variance reduction method that exploits the statistical correlation between the low-fidelity approximations and the high-fidelity HDG discretization to accelerate the convergence of the Monte Carlo simulations. The multilevel variance reduction method provides efficient computation of the statistical outputs by shifting most of the computational burden from the high-fidelity HDG approximation to the RB approximations. Furthermore, we develop *a posteriori* error estimates for our approximations of the statistical outputs. Based on these error estimates, we propose an algorithm for optimally choosing both the dimensions of the RB approximations and the size of Monte Carlo samples to achieve a given error tolerance. In addition, we extend the method to compute estimates for the gradients of the statistical outputs. The proposed method is particularly useful for stochastic optimization problems where many evaluations of the objective function and its gradient are required.

Key words. Model reduction, variance reduction, empirical interpolation method, reduced basis method, *a posteriori* error estimation, hybridizable discontinuous Galerkin method, multilevel Monte Carlo method, stochastic elliptic parametrized PDEs, stochastic optimization

AMS subject classifications.

1. Introduction.

In this paper we propose a methodology to accelerate the computation of statistical outputs from parametrized stochastic partial differential equations (PSPDEs). Let us define the quantity of interest $s(\boldsymbol{\theta}, \boldsymbol{y})$ as a functional of a high-fidelity solution $u(\boldsymbol{\theta}, \boldsymbol{y})$ of an underlying PSPDE, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_Q) \in \Theta$ is a vector of deterministic design parameters in the parameter space Θ and $\boldsymbol{y}$ is a vector of random parameters with probability density function $\rho(\boldsymbol{y})$. We focus on the evaluation of both the expectation $E\left[s(\boldsymbol{\theta}, \boldsymbol{y})\right]$ and variance $V\left[s(\boldsymbol{\theta}, \boldsymbol{y})\right]$ as well as their gradients for many different values of $\boldsymbol{\theta} \in \Theta$. This task can be extremely expensive if classic techniques such as Monte Carlo (MC) are employed.

The method developed herein is intended to tackle scenarios that require multiple evaluations of the statistical outputs. One of such scenarios is the stochastic optimization problem:

$$\text{(SP)} \qquad \min_{\boldsymbol{\theta} \in \Theta} F_\gamma(\boldsymbol{\theta}) = E\left[s(\boldsymbol{\theta}, \boldsymbol{y})\right] + \gamma \sqrt{V\left[s(\boldsymbol{\theta}, \boldsymbol{y})\right]},$$

[†]Aerospace Computational Design Lab (ACDL). Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 (fvidal@mit.edu, cuongng@mit.edu, peraire@mit.edu). This work was supported by AFOSR Grant No. FA9550-11-1-0141, AFOSR Grant No. FA9550-12-0357, and the Singapore-MIT Alliance. The first author was also supported by Obra Social la Caixa.

[‡]Mathematical Institute, University of Oxford, Oxford, UK (mike.giles@maths.ox.ac.uk).

where $\gamma \geq 0$ is a given constant. The problem (SP) arises when optimizing an observable quantity of a physical system governed by a PDE in the presence of randomness. Solving (SP) often requires multiple queries of the objective function $F_\gamma(\boldsymbol{\theta})$ and its gradient.

The most natural approach is to approximate the objective function using a MC simulation

$$\text{(SP-MC)} \qquad \min_{\boldsymbol{\theta}\in\Theta} F_{M,\gamma}(\boldsymbol{\theta}) = E_M\left[s(\boldsymbol{\theta},\boldsymbol{y})\right] + \gamma\sqrt{V_M\left[s(\boldsymbol{\theta},\boldsymbol{y})\right]}$$

with M i.i.d. samples drawn from the corresponding probability distribution. A common technique to solve the problem (SP-MC) is to use sample average approximation (SAA) [41, 38, 42] in a deterministic optimization method, where the samples are fixed beforehand and reused at each optimization step. Extensive research has been conducted for problems with special structure [23, 39] developing convergence rates and stochastic bounds [27, 3]. Robbins-Monro stochastic approximation (SA) [36] is a different approach involving an iterative procedure that resembles steepest descent with stochastic gradient information. The Robbins-Monro algorithm is the most widely used stochastic approximation technique, and it has motivated substantial research regarding convergence conditions and step size policies [34, 29, 4].

In any event, in order to solve (SP-MC), we may need many evaluations of the high-fidelity output due to the slow rate of convergence of MC. This situation negatively impacts the computational cost, since the high-fidelity output can be very expensive. A possible strategy to alleviate the computational cost is to approximate the high-fidelity output with an inexpensive surrogate. A popular technique is the reduced basis (RB) method [32, 35, 31], which enables the computation of real-time solutions and error bounds of parametrized PDEs. The RB method has been used for various uncertainty quantification problems: evaluation of statistical moments using MC sampling methods [6, 18, 44]; PDE-constrained stochastic optimal control problems, combining the RB method with sparse grid stochastic collocation for the approximation of the stochastic space [9, 8]; Bayesian estimation of parameters with control variates [5]; and Bayesian inversion problems [10]. In these problems, the RB method effects a significant reduction of the computational cost, since the expensive truth approximation is replaced by the inexpensive RB model.

In our previous work [44] we introduced a model and variance reduction method (MVR) to address evaluation of statistical moments for stochastic PDEs. In this article we extend the MVR method by combining it with the empirical interpolation (EI) method [1] to tackle scenarios where evaluating output statistics for many queries is desired. Instead of computing the statistical output for each parameter value independently, we exploit the fact that the solutions lie on a low-dimensional manifold induced by the $(\boldsymbol{\theta},\boldsymbol{y})$-dependence. The main idea is to construct an empirical interpolant that decouples the parametric and stochastic influence of the problem, enabling an offline-online strategy that allows us to reuse the precomputed RB outputs. The proposed method can be advantageous, for instance, in solving stochastic optimization problems such as (SP). This approach is built upon variance reduction techniques such as the control variates method [5, 7, 19], the multilevel Monte Carlo method [14, 2, 12, 43], the multi-fidelity Monte Carlo method [30], and parametric multilevel Monte Carlo [21, 20]. The common strategy for these techniques is to achieve a reduction in the variance of the output by making use of the correlation between the output and a surrogate.

This article is organized as follows. In Section 2 we review the computation of the output and its gradient, using the hybridizable discontinuous Galerkin (HDG) method for a high-fidelity approximation and the reduced basis (RB) method for a real-time, albeit coarser, approximation of the output. In Section 3 we pursue a decomposition between design variables $\boldsymbol{\theta}$ and stochastic variables $\boldsymbol{y}$ through the EIM [1] that allows for an efficient offline-online objective function evaluation. In Section 4 we devise a multilevel control variates method that exploits the statistical correlation between different output approximations to accelerate the convergence of the Monte Carlo simulations, extending the framework in [44] to include EI approximations. In addition, we also propose estimators for the gradient of statistical outputs by combining the gradients of the different output approximations in a similar multilevel structure. In Section 5, we present numerical results to demonstrate the performance of the proposed method. Finally, in Section 6, we discuss some directions for future research.

2. HDG Discretization and Model Reduction.

2.1. Problem statement. Let $\mathcal{D} \in \mathbb{R}^d$ be a regular domain with Lipschitz boundary $\partial\mathcal{D}$. We consider the following stochastic boundary value problem: find a function u such that,

$$-\nabla \cdot (\kappa \boldsymbol{\nabla} u) + \varrho u = f, \qquad \forall \boldsymbol{x} \in \mathcal{D}\,, \tag{2.1a}$$

$$\kappa \boldsymbol{\nabla} u \cdot \boldsymbol{n} + \nu u \;= g, \qquad \forall \boldsymbol{x} \in \partial\mathcal{D}\,, \tag{2.1b}$$

where f is the source term, κ is the diffusion coefficient, ϱ is the Helmholtz parameter, ν is the Robin coefficient, and g is the boundary data. For simplicity of exposition we shall assume that ϱ is a stochastic function and that f, κ, ν, g are deterministic. Note that since we allow $\varrho, f, \kappa, \nu, g$ to be complex-valued functions, the solution u is in general a complex stochastic function.

We next introduce a probability space $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is the σ-algebra of the subsets of Ω, and P is the probability measure. If Z is a real random variable in $(\Omega, \mathcal{F}, P)$ and ω an element of the probability space, we denote its expectation by $E[Z] = \int_\Omega Z(\omega) dP(\omega)$. We will consider random functions v in $L^2(\mathcal{D} \times \Omega)$ equipped with the following norm

$$\|v\|^2 = E\left[\int_{\mathcal{D}} |v(\boldsymbol{x}, \cdot)|^2 d\boldsymbol{x}\right] = \int_\Omega \int_{\mathcal{D}} |v(\boldsymbol{x}, \omega)|^2 d\boldsymbol{x}\, dP(\omega).$$

We will assume that $\varrho \in L^2(\mathcal{D} \times \Theta \times \Omega)$, where $\varrho \equiv \varrho(\boldsymbol{\theta}, \boldsymbol{y})$ depends affinely on some parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^Q$ and random variables $\boldsymbol{y} = \big(y_1(\omega), \ldots, y_{Q'}(\omega)\big)$ for $\omega \in \Omega$. It thus follows that we can write ϱ in the form

$$\varrho(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y}) = \varrho_0(\boldsymbol{x}) + \sum_{r=1}^{R} \sigma_r(\boldsymbol{\theta}, \boldsymbol{y}) \varrho_r(\boldsymbol{x}). \tag{2.2}$$

The amount of terms R in the expansion depends on the interaction between the design and stochastic parameters, but we can assume it is in the order of QQ'. We shall assume that the random variables are mutually independent and defined in the interval $\Lambda_{q'} \subset \mathbb{R}$ with a probability density function $\rho_{q'} : \Lambda_{q'} \to \mathbb{R}^+$, hence $\boldsymbol{y} \in \Lambda$ with $\Lambda = \prod_{q'=1}^{Q'} \Lambda_{q'}$.

Therefore, the solution u of (2.1) can be written as a function of $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{y} \in \Lambda$, namely, $u(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})$. Now let ℓ be a bounded linear functional. We introduce a random output s defined as

$$s(\boldsymbol{\theta}, \boldsymbol{y}) = \ell(u(\cdot, \boldsymbol{\theta}, \boldsymbol{y})). \tag{2.3}$$

We are interested in evaluating the expectation and variance of s for any value of the design parameters $\boldsymbol{\theta}$ as

$$E[s(\boldsymbol{\theta}, \boldsymbol{y})] = \int_\Lambda s(\boldsymbol{\theta}, \boldsymbol{y})\rho(\boldsymbol{y})d\boldsymbol{y}, \qquad V[s(\boldsymbol{\theta}, \boldsymbol{y})] = \int_\Lambda \left(E[s(\boldsymbol{\theta}, \boldsymbol{y})] - s(\boldsymbol{\theta}, \boldsymbol{y})\right)^2 \rho(\boldsymbol{y})d\boldsymbol{y},$$

where $\rho(\boldsymbol{y}) = \prod_{q'=1}^{Q'} \rho_{q'}(y_{q'})$.

2.2. Problem discretization. We now state the results for computing the output (2.3) using the HDG method to discretize (2.1), referring the reader to [44] for a more thorough derivation. The physical domain $\mathcal{D}$ is triangulated into elements T forming a mesh $\mathcal{T}_h$ satisfying the standard finite element conditions [11]. Then, letting $\partial\mathcal{T}_h := \{\partial T : T \in \mathcal{T}_h\}$ and denoting by $\mathcal{F}_h$ the set of the faces F of the elements $T \in \mathcal{T}_h$, we seek a scalar approximation $u_h \in W_h^p$ to u, and a scalar approximation $\widehat{u}_h \in M_h^p$ to the *trace* of u on element boundaries, where

$$\begin{aligned} W_h^p &= \{w \in L^2(\mathcal{D}) : w|_T \in P_p(T) \ \ \forall T \in \mathcal{T}_h\}, \\ M_h^p &= \{\mu \in L^2(\mathcal{F}_h) : \mu|_F \in P_p(F) \ \forall F \in \mathcal{F}_h\}, \end{aligned}$$

and $P_p(D)$ is a space of complex-valued polynomials of degree at most p on D. Note that $\widehat{u}_h$ are defined only on the faces of the elements, hence they are single valued. Following the discussion in [44], we set the $\mathcal{N}$-dimensional approximation space to be $\boldsymbol{W}_h^p := W_h^p \times M_h^p$, for a solution $\boldsymbol{u}_h := (u_h, \widehat{u}_h)$, whose degrees of freedom are represented in the vector $\mathbf{u}$. Since we never have access to the true solution, we approximate the output using our HDG solution as

$$s_h(\boldsymbol{\theta}, \boldsymbol{y}) = \boldsymbol{\ell}^\dagger \mathbf{u}(\boldsymbol{\theta}, \boldsymbol{y}),$$

where $\boldsymbol{\ell}$ results from the HDG approximation of the linear functional $\ell(\cdot)$ and the superscript $\dagger$ denotes the conjugate transpose. The degrees of freedom for the field $\boldsymbol{u}_h$ are obtained as the solution of the linear system

$$\mathbf{A}(\boldsymbol{\theta}, \boldsymbol{y})\mathbf{u} = \mathbf{b}, \tag{2.4}$$

where the matrix $\mathbf{A}(\boldsymbol{\theta}, \boldsymbol{y})$ and the vector $\mathbf{b}$ result from the weak formulation of the HDG method described in [44]. The HDG discretization and the expression (2.2) allows for an affine representation of the matrix as

$$\mathbf{A}(\boldsymbol{\theta}, \boldsymbol{y}) = \mathbf{A}_0 + \sum_{r=1}^{R} \sigma_r(\boldsymbol{\theta}, \boldsymbol{y})\mathbf{A}_r,$$

where the matrices $\mathbf{A}_r, r = 0, \dots R$, are independent of $(\boldsymbol{\theta}, \boldsymbol{y})$. Finally, even though the parameter independent matrices are used for the reduced basis approximation, the solution $\boldsymbol{u}_h$

is never computed as the solution of the full system (2.4). Instead, we can invoke discontinuity of the approximation spaces to write u_h in terms of $\widehat{u}_h$. This common strategy in HDG methods enables us to solve for the global degrees of freedom of $\widehat{u}_h$ only and then recover u_h efficiently.

2.3. Reduced Basis formulation. We now briefly review the RB method [32, 35, 31] for approximating the output of a parametric PDE. Let us assume that we are given $N_{\max}$ basis functions $\boldsymbol{\zeta}_n \in \boldsymbol{W}_h^p$, $1 \le n \le N_{\max}$. We define the associated hierarchical RB space as

$$\boldsymbol{W}_N = \operatorname{span}\{\boldsymbol{\zeta}_n, 1 \le n \le N\}, \quad N = 1, \dots, N_{\max} \ .$$

The space $\boldsymbol{W}_N$ consists of orthonormalized solutions at parameter values selected by a Greedy sampling procedure [37, 16]. We then form the matrix $\boldsymbol{\Psi}_N$ whose columns are the vectors of degrees of freedom of the first N basis functions.

Finally, we apply the Galerkin projection as follows: Given $(\boldsymbol{\theta}, \boldsymbol{y}) \in \Theta \times \Lambda$, we find a RB approximation $\boldsymbol{u}_N(\boldsymbol{\theta}, \boldsymbol{y}) = \boldsymbol{\Psi}_N \boldsymbol{\lambda}_N$ in which the coefficient vector $\boldsymbol{\lambda}_N$ is the solution of

$$\left(\boldsymbol{\Psi}_N^\dagger \mathbf{A}(\boldsymbol{\theta}, \boldsymbol{y}) \boldsymbol{\Psi}_N\right) \boldsymbol{\lambda}_N = \boldsymbol{\Psi}_N^\dagger \mathbf{b} \, . \tag{2.5}$$

We can now evaluate the RB output as

$$s_N(\boldsymbol{\theta}, \boldsymbol{y}) = \boldsymbol{\ell}^\dagger \boldsymbol{u}_N(\boldsymbol{\theta}, \boldsymbol{y}). \tag{2.6}$$

The above RB formulation is implemented using the offline-online decomposition procedure [31, 35]. The online cost of evaluating the RB output $s_N(\boldsymbol{\theta}, \boldsymbol{y})$ for any given $(\boldsymbol{\theta}, \boldsymbol{y}) \in \Theta \times \Lambda$ is $O(N^3 + RN^2)$; see [31, 35] for details.

The above formulation can be further extended by including the dual problem and developing a "primal-dual" formulation, which usually leads to superior RB convergence and error characterization of the output [33, 31]. However, we do not include the primal-dual formulation in this work as we focus only on the multiple output scenario, where it is more efficient to develop only a primal RB.

2.4. Computation of gradients. In the optimization context the usage of first order optimization algorithms usually leads to accelerated convergence to the optimum value. For stochastic problems, where each evaluation of the objective function is expensive, the incorporation of gradient information can lead to a more efficient exploration of the design space, and it should be used whenever available. We review now how to obtain the gradients for both the HDG output s_h and the RB output s_N with respect to the design parameters $\boldsymbol{\theta}$.

For clarity, we drop the $\boldsymbol{y}$-dependence on $\mathbf{A}$ since it is irrelevant for gradient computation, and for completeness we also consider parametric dependence for the source term $\mathbf{b}$. We recall that the HDG output is evaluated as

$$\mathbf{A}(\boldsymbol{\theta})\mathbf{u}(\boldsymbol{\theta}) = \mathbf{b}(\boldsymbol{\theta}), \qquad s_h(\boldsymbol{\theta}) = \boldsymbol{\ell}^\dagger \mathbf{u}(\boldsymbol{\theta}). \tag{2.7}$$

The output derivatives with respect to the design parameters can be computed as

$$\frac{\partial s_h}{\partial \theta_q} = \frac{\partial s_h}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \theta_q} = \boldsymbol{\ell}^\dagger \mathbf{A}^{-1} \left(\frac{\partial \mathbf{b}}{\partial \theta_q} - \frac{\partial \mathbf{A}}{\partial \theta_q} \mathbf{u} \right) = \boldsymbol{\phi}^\dagger \left(\frac{\partial \mathbf{b}}{\partial \theta_q} - \frac{\partial \mathbf{A}}{\partial \theta_q} \mathbf{u} \right),$$

where $\boldsymbol{\phi}$ is the solution of the adjoint system $\mathbf{A}^\dagger \boldsymbol{\phi} = \boldsymbol{\ell}$.

The gradients for the RB output can be computed analogously. We note that the primal RB formulation (2.5) and the RB output (2.6) can be written as

$$\begin{aligned} \mathbf{A}_N(\boldsymbol{\theta})\boldsymbol{\lambda}_N(\boldsymbol{\theta}) &= \mathbf{b}_N(\boldsymbol{\theta}), \\ s_N(\boldsymbol{\theta}) &= \boldsymbol{\ell}_N^\dagger \boldsymbol{\lambda}_N(\boldsymbol{\theta}), \end{aligned} \tag{2.8}$$

where $\mathbf{A}_N(\boldsymbol{\theta}) = \boldsymbol{\Psi}_N^\dagger \mathbf{A}(\boldsymbol{\theta})\boldsymbol{\Psi}_N$, $\mathbf{b}_N(\boldsymbol{\theta}) = \boldsymbol{\Psi}_N^\dagger \mathbf{b}(\boldsymbol{\theta})$ and $\boldsymbol{\ell}_N = \boldsymbol{\Psi}_N^\dagger \boldsymbol{\ell}$. The derivatives of the RB output can be evaluated efficiently through the adjoint equation as follows

$$\frac{\partial s_N}{\partial \theta_q} = \boldsymbol{\ell}_N^\dagger \mathbf{A}_N^{-1}\left[\frac{\partial \mathbf{b}_N}{\partial \theta_q} - \frac{\partial \mathbf{A}_N}{\partial \theta_q}\boldsymbol{\lambda}_N\right] = \boldsymbol{\eta}_N^\dagger\left[\frac{\partial \mathbf{b}_N}{\partial \theta_q} - \frac{\partial \mathbf{A}_N}{\partial \theta_q}\boldsymbol{\lambda}_N\right] = \boldsymbol{\eta}_N^\dagger \frac{\partial \mathbf{b}_N}{\partial \theta_q} - \sum_{r=1}^{R} \frac{\partial \sigma_r}{\partial \theta_q}\boldsymbol{\eta}_N^\dagger \mathbf{A}_{N,r}\boldsymbol{\lambda}_N.$$

where $\boldsymbol{\eta}_N$ is given by

$$\mathbf{A}_N^\dagger \boldsymbol{\eta}_N = \boldsymbol{\ell}_N.$$

The gradient is evaluated with only $\mathcal{O}\left(RN^2\right)$ extra complexity, since we reuse the LU decomposition of $\mathbf{A}_N$ from Equation (2.8). This analysis can be extended in a straightforward manner for the primal-dual formulation, and to include Hessian information as well.

3. Empirical Interpolation Method.

The minimization problem defined in (SP) requires, for any new design parameter $\boldsymbol{\theta}$, the computation of the expected value $E[s_h(\boldsymbol{\theta}, \boldsymbol{y})]$. Approximating this expectation by a MC simulation combined with the RB approximation has been considered in [6]. In particular, the MC-RB approximation of $E[s_h(\boldsymbol{\theta}, \boldsymbol{y})]$ is calculated as

$$E_M[s_N(\boldsymbol{\theta}, \boldsymbol{y})] = \frac{1}{M}\sum_{m=1}^{M} s_N(\boldsymbol{\theta}, \boldsymbol{y}_m), \tag{3.1}$$

where $Y_M = \{\boldsymbol{y}_m \in \Lambda, 1 \leq m \leq M\}$ is a collection of i.i.d. samples drawn from the density function $\rho(\boldsymbol{y})$. This will result in a computational cost of $O(M(N^3 + RN^2))$. This procedure can be very expensive if many expectations need to be evaluated and if M is very large.

An alternative methodology is the empirical interpolation method [1] for separating the parametric variables $\boldsymbol{\theta}$ and stochastic variables $\boldsymbol{y}$. Let $K > 0$ be a positive integer. The idea behind the empirical interpolation is to construct a set of interpolation points $\{\overline{\boldsymbol{y}}_k \in \Lambda\}_{k=1}^K$ and a set of interpolation basis functions $\{v_k(\boldsymbol{y})\}_{k=1}^K$ such that, for any $\boldsymbol{\theta} \in \Theta$, we can approximate $s_N(\boldsymbol{\theta}, \boldsymbol{y})$ by a function $\widetilde{s}_N(\boldsymbol{\theta}, \boldsymbol{y})$, defined as

$$\widetilde{s}_N(\boldsymbol{\theta}, \boldsymbol{y}) = \sum_{k=1}^{K} \alpha_k(\boldsymbol{\theta}) v_k(\boldsymbol{y}). \tag{3.2}$$

Here the coefficient vector $\boldsymbol{\alpha}(\boldsymbol{\theta})$ is given by

$$\boldsymbol{\alpha}(\boldsymbol{\theta}) = \boldsymbol{C}^{-1}\boldsymbol{c}(\boldsymbol{\theta}), \tag{3.3}$$

where $c_l(\boldsymbol{\theta}) = s_N(\boldsymbol{\theta}, \overline{\boldsymbol{y}}_l)$ for $1 \le k, l \le K$ and $\boldsymbol{C}$ is introduced in Algorithm 1. The gradient of $\widetilde{s}_N(\boldsymbol{\theta}, \boldsymbol{y})$ is then calculated as

$$\frac{\partial \widetilde{s}_N(\boldsymbol{\theta}, \boldsymbol{y})}{\partial \theta_q} = \sum_{k=1}^{K} \frac{\partial \alpha_k(\boldsymbol{\theta})}{\partial \theta_q} v_k(\boldsymbol{y}), \tag{3.4}$$

where

$$\boldsymbol{C}\frac{\partial \boldsymbol{\alpha}(\boldsymbol{\theta})}{\partial \theta_q} = \frac{\partial \boldsymbol{c}(\boldsymbol{\theta})}{\partial \theta_q}, \qquad \frac{\partial c_l(\boldsymbol{\theta})}{\partial \theta_q} = \frac{\partial s_N(\boldsymbol{\theta}, \overline{\boldsymbol{y}}_l)}{\partial \theta_q}, \quad l = 1, \ldots, K. \tag{3.5}$$

Here the gradient of the RB output $s_N(\boldsymbol{\theta}, \overline{\boldsymbol{y}}_l)$ is computed as described above.

The naive choice of basis is $v_k(\boldsymbol{y}) = s_N(\overline{\boldsymbol{\theta}}_k, \boldsymbol{y})$ for $1 \le k \le K$, where $\{\overline{\boldsymbol{\theta}}_k \in \Theta\}_{k=1}^K$ are selected sample points; but it should be avoided as the strong linear dependence between basis vectors will adversely affect the numerical stability of the interpolation [1]. We can adopt a more numerically stable basis defined by

$$[v_1(\boldsymbol{y}), \ldots, v_K(\boldsymbol{y})] = [s_N(\overline{\boldsymbol{\theta}}_1, \boldsymbol{y}), \ldots, s_N(\overline{\boldsymbol{\theta}}_K, \boldsymbol{y})]\boldsymbol{D}, \qquad \boldsymbol{y} \in \Lambda, \tag{3.6}$$

where $\boldsymbol{D} = \boldsymbol{E}^{-1}\boldsymbol{C}$ and $\boldsymbol{E}$ is a $K \times K$ matrix with entries $E_{kl} = s_N(\overline{\boldsymbol{\theta}}_k, \overline{\boldsymbol{y}}_l), 1 \le l, k \le K$.

The sets $\{\overline{\boldsymbol{y}}_k \in Y_J\}_{k=1}^K$, $\{\overline{\boldsymbol{\theta}}_k \in \Theta_I\}_{k=1}^K$ and the matrix $\boldsymbol{C}$ are computed by using the empirical interpolation method [1] (summarized in Algorithm 1) on the $\boldsymbol{\theta}$-training set $\Theta_I = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_I]$ and the $\boldsymbol{y}$-training set $Y_J = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_J]$. These training sets are typically very large, generated with uniform samples for $\boldsymbol{\theta}$ and with samples drawn from $\rho(\boldsymbol{y})$ for $\boldsymbol{y}$. Another option is to employ quasi-random sequences for the training sets in order to better explore the spaces, or even adaptive sparse grids to help mitigate the curse of dimensionality. Note that the cost for computing $s_N(\boldsymbol{\theta}_i, \boldsymbol{y}_j)$ is $O(IJ(N^3 + RN^2))$.

After executing Algorithm 1, we need to evaluate the basis functions $\{v_k(\boldsymbol{y})\}_{k=1}^K$ on the desired set of samples Y_M as

$$[v_1(\boldsymbol{y}_m), \ldots, v_K(\boldsymbol{y}_m)] = [s_N(\overline{\boldsymbol{\theta}}_1, \boldsymbol{y}_m), \ldots, s_N(\overline{\boldsymbol{\theta}}_K, \boldsymbol{y}_m)]\boldsymbol{D}, \qquad \forall\, \boldsymbol{y}_m \in Y_M, \tag{3.7}$$

with an additional cost of $O(KM(N^3+RN^2))$. Hence, the total computational cost is $O((IJ+KM)(N^3 + RN^2))$. Note that the empirical interpolation is executed only once before we actually compute the expectation and its gradient for any $\boldsymbol{\theta}$, and that (3.2) will only be evaluated for the samples in Y_M.

Finally, we can evaluate the associated expectation for any $\boldsymbol{\theta} \in \Theta$ as

$$E_M\left[\widetilde{s}_N(\boldsymbol{\theta}, \boldsymbol{y})\right] = \frac{1}{M}\sum_{m=1}^{M}\left(\sum_{k=1}^{K} \alpha_k(\boldsymbol{\theta}) v_k(\boldsymbol{y}_m)\right), \tag{3.8}$$

where $\boldsymbol{\alpha}(\boldsymbol{\theta})$ is calculated using (3.3) with $O(K(N^3+RN^2+K))$ operations. So, $E_M\left[\widetilde{s}_N(\boldsymbol{\theta}, \boldsymbol{y})\right]$ requires $O(K(N^3 + RN^2 + K + M))$ in complexity. Furthermore, its gradient can be computed with additional cost of $O(KRN^2)$. Due to the initial cost for performing the empirical interpolation procedure once, this approach makes sense only when we wish to compute the expectation and its gradient for many queries of $\boldsymbol{\theta}$.

Algorithm 1 Empirical interpolation method on $s_N(\boldsymbol{\theta}, \boldsymbol{y})$

Require: Training sets $\Theta_I = \{\boldsymbol{\theta}_i, 1 \leq i \leq I\}$ and $Y_J = \{\boldsymbol{y}_j, 1 \leq j \leq J\}$. The notation $\boldsymbol{u}_i$ refers to the i-th vector of a set of I vectors, and u_i^j denotes the j-th component of $\boldsymbol{u}_i$.

1: Compute $p_i^j = s_N(\boldsymbol{\theta}_i, \boldsymbol{y}_j)$ for all $(\boldsymbol{\theta}_i, \boldsymbol{y}_j) \in \Theta_I \times Y_J$, $1 \leq i \leq I, 1 \leq j \leq J$.

2: Choose the first sample point $\overline{\boldsymbol{\theta}}_1$ as

$$i_1 = \arg\max_{1 \leq i \leq I} \|\boldsymbol{p}_i\|_\infty, \qquad \overline{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_{i_1}.$$

3: Choose the first interpolation point $\overline{\boldsymbol{y}}_1$ and the first interpolation vector $\boldsymbol{z}_1$ as

$$j_1 = \arg\max_{1 \leq j \leq J} |p_{i_1}^j|, \qquad \overline{\boldsymbol{y}}_1 = \boldsymbol{y}_{j_1}, \qquad \boldsymbol{z}_1 = \boldsymbol{p}_{i_1}/p_{i_1}^{j_1}.$$

4: Initialize the matrix $\boldsymbol{C}$ as $C_{1\,1} = 1$. Set $k = 2$.

5: **while** $k \leq K$ **do**

6: For each $i \in \{1, \ldots, I\}$, set $c_l(\boldsymbol{\theta}_i) = s_N(\boldsymbol{\theta}_i, \overline{\boldsymbol{y}}_l), 1 \leq l \leq k-1$ and compute the interpolant function of $\boldsymbol{p}_i$ as

$$\mathcal{I}_{k-1}[\boldsymbol{p}_i] = \sum_{l=1}^{k-1} \alpha_l(\boldsymbol{\theta}_i)\boldsymbol{z}_l, \qquad \boldsymbol{\alpha}(\boldsymbol{\theta}_i) = \boldsymbol{C}^{-1}\boldsymbol{c}(\boldsymbol{\theta}_i).$$

7: Compute the interpolation error $\boldsymbol{e}_i = \boldsymbol{p}_i - \mathcal{I}_{k-1}[\boldsymbol{p}_i]$. Choose the next sample point $\overline{\boldsymbol{\theta}}_k$ as

$$i_k = \arg\max_{1 \leq i \leq I} \|\boldsymbol{e}_i\|_\infty, \qquad \overline{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_{i_k},$$

and define the mean interpolation error as $\overline{e}_i = J^{-1}\sum_j |e_i^j|$.

8: Choose the next interpolation point $\overline{\boldsymbol{y}}_k$ and the next interpolation vector $\boldsymbol{z}_k$ as

$$j_k = \arg\max_{1 \leq j \leq J} |e_{i_k}^j|, \qquad \overline{\boldsymbol{y}}_k = \boldsymbol{y}_{j_k}, \qquad \boldsymbol{z}_k = \boldsymbol{e}_{i_k}/e_{i_k}^{j_k}.$$

9: Update the matrix $\boldsymbol{C}$ as $C_{l\,l'} = z_{l'}^{j_l}, 1 \leq l, l' \leq k$. Set $k = k + 1$.

10: **end while**

4. Empirical Interpolation and Model-Variance Reduction Method. The method presented here is an extension of the multilevel control variates framework introduced in [44]. The extension is done by introducing the EI as an additional control variate level.

4.1. Two-level Monte Carlo estimator. Consider the scenario where, for each new design parameter $\boldsymbol{\theta}$, we pursue the computation of an estimate of $E[s_h(\boldsymbol{\theta}, \boldsymbol{y})]$. In [44] we introduced the estimator

$$E[s_h] = E[s_h - s_{N_1}] + E[s_{N_1}]. \tag{4.1}$$

where $s_{N_1}(\boldsymbol{\theta}, \boldsymbol{y})$ is the RB output developed in Section 2.3 for some $N_1 \in [1, N_{\max}]$. The underlying premise here is that the two expectation terms on the right hand side can be computed efficiently by MC simulations owing to variance reduction and model reduction: the first term requires a small number of samples because its variance is generally very small, and the second term is less expensive to evaluate because it involves the RB output.

In this paper, we expand the estimator (4.1) by proposing the following estimator

$$E[s_h] = E[s_h - s_{N_1}] + E[s_{N_1} - \widetilde{s}_{N_1}] + E[\widetilde{s}_{N_1}]. \tag{4.2}$$

The main advantage of this estimator over (4.1) is that it is faster to compute the sum $E[s_{N_1} - \widetilde{s}_{N_1}] + E[\widetilde{s}_{N_1}]$ than $E[s_{N_1}]$ by using MC simulation. In particular, let $Y^0_{M_0} = \{\boldsymbol{y}^0_m \in \Lambda, 1 \le m \le M_0\}$, $Y^1_{M_1} = \{\boldsymbol{y}^1_m \in \Lambda, 1 \le m \le M_1\}$ and $Y^1_{\widetilde{M}_1} = \{\widetilde{\boldsymbol{y}}^1_m \in \Lambda, 1 \le m \le \widetilde{M}_1\}$ be three independent sets of random samples drawn in Λ with the probability density function $\rho(\boldsymbol{y})$. Actually, the set where $\widetilde{s}_{N_1}$ is evaluated is precisely Y_M in (3.7), hence $Y_M = Y^1_{M_1} \cup Y^1_{\widetilde{M}_1}$. We calculate our Empirical Interpolation Model-Variance Reduction (EIMVR) estimate of $E[s_h(\boldsymbol{\theta}, \boldsymbol{y})]$ as

$$E_{M_0, M_1, \widetilde{M}_1}[s_h] = E_{M_0}[s_h - s_{N_1}] + E_{M_1}[s_{N_1} - \widetilde{s}_{N_1}] + E_{\widetilde{M}_1}[\widetilde{s}_{N_1}], \tag{4.3}$$

where

$$\begin{aligned} E_{M_0}[s_h - s_{N_1}] &= \frac{1}{M_0} \sum_{m=1}^{M_0} \left(s_h(\boldsymbol{\theta}, \boldsymbol{y}^0_m) - s_{N_1}(\boldsymbol{\theta}, \boldsymbol{y}^0_m) \right), \\ E_{M_1}[s_{N_1} - \widetilde{s}_{N_1}] &= \frac{1}{M_1} \sum_{m=1}^{M_1} \left(s_{N_1}(\boldsymbol{\theta}, \boldsymbol{y}^1_m) - \widetilde{s}_{N_1}(\boldsymbol{\theta}, \boldsymbol{y}^1_m) \right), \\ E_{\widetilde{M}_1}[\widetilde{s}_{N_1}] &= \frac{1}{\widetilde{M}_1} \sum_{m=1}^{\widetilde{M}_1} \widetilde{s}_{N_1}(\boldsymbol{\theta}, \widetilde{\boldsymbol{y}}^1_m). \end{aligned} \tag{4.4}$$

The EI is responsible for alleviating the computational effort in our method since $E_{\widetilde{M}_1}[\widetilde{s}_{N_1}]$ only requires the solution of (3.3). In addition, M_1 is often quite small since s_{N_1} and $\widetilde{s}_{N_1}$ are highly correlated.

Moreover, independence of the sample sets allows us (see [44]) to invoke the CLT to obtain an error estimate for the expectation error as

$$\lim_{M_0 \to \infty} \lim_{M_1 \to \infty} \lim_{\widetilde{M}_1 \to \infty} \Pr\left(\left| E[s_h] - E_{M_0, M_1, \widetilde{M}_1}[s_h] \right| \le \Delta^E_{M_0, M_1, \widetilde{M}_1} \right) = \operatorname{erf}\left(\frac{a}{\sqrt{2}} \right), \tag{4.5}$$

where the *a posteriori* error bound reads

$$\Delta^E_{M_0, M_1, \widetilde{M}_1} = a \sqrt{\frac{V_{M_0}[s_h - s_{N_1}]}{M_0} + \frac{V_{M_1}[s_{N_1} - \widetilde{s}_{N_1}]}{M_1} + \frac{V_{\widetilde{M}_1}[\widetilde{s}_{N_1}]}{\widetilde{M}_1}}, \tag{4.6}$$

with $a > 0$ being the confidence level in the CLT.

Similarly, the EIMVR estimate of the true variance $V[s_h]$ is given by

$$V_{M_0,M_1,\widetilde{M}_1}[s_h] = E_{M_0}[\zeta_h - \zeta_{N_1}] + E_{M_1}[\zeta_{N_1} - \widetilde{\zeta}_{N_1}] + E_{\widetilde{M}_1}[\widetilde{\zeta}_{N_1}]\,, \tag{4.7}$$

with new variables $\zeta_h = (s_h - E_{M_0,M_1,\widetilde{M}_1}[s_h])^2$, $\zeta_{N_1} = (s_{N_1} - E_{M_0,M_1,\widetilde{M}_1}[s_h])^2$ and $\widetilde{\zeta}_{N_1} = (\widetilde{s}_{N_1} - E_{M_0,M_1,\widetilde{M}_1}[s_h])^2$. The application of the CLT yields

$$\lim_{M_0\to\infty}\lim_{M_1\to\infty}\lim_{\widetilde{M}_1\to\infty} \Pr\left(\left|V[s_h] - V_{M_0,M_1,\widetilde{M}_1}[s_h]\right| \le \Delta^V_{M_0,M_1,\widetilde{M}_1}\right) = \operatorname{erf}\left(\frac{a}{\sqrt{2}}\right), \tag{4.8}$$

where

$$\Delta^V_{M_0,M_1,\widetilde{M}_1} = a\sqrt{\frac{V_{M_0}[\zeta_h - \zeta_{N_1}]}{M_0} + \frac{V_{M_1}[\zeta_{N_1} - \widetilde{\zeta}_{N_1}]}{M_1} + \frac{V_{\widetilde{M}_1}[\widetilde{\zeta}_{N_1}]}{\widetilde{M}_1}}. \tag{4.9}$$

For further details on the derivation of (4.5)-(4.9) we refer the reader to [44].

4.2. Multilevel Monte Carlo estimator. The method can be further generalized and improved by pursuing a multilevel control variate strategy. Given L different RB output models $s_{N_\ell}(\boldsymbol{y}), 1 \le \ell \le L$, with $N_1 > N_2 > \ldots > N_L$, and an EI decomposition of the coarsest RB model $\widetilde{s}_{N_L}$, we first express the expected value as

$$E[s_h] = E[s_h - s_{N_1}] + \sum_{\ell=1}^{L-1} E[s_{N_\ell} - s_{N_{\ell+1}}] + E[s_{N_L} - \widetilde{s}_{N_L}] + E[\widetilde{s}_{N_L}].$$

We next introduce $L+2$ independent sample sets $Y^\ell_{M_\ell} = \{\boldsymbol{y}^\ell_m \in \Lambda, 1 \le m \le M_\ell\}$, $0 \le \ell \le L$, and $Y^L_{\widetilde{M}_L} = \{\widetilde{\boldsymbol{y}}^L_m \in \Lambda, 1 \le m \le \widetilde{M}_L\}$, which are drawn in Λ with probability density function $\rho(\boldsymbol{y})$. We remark that $\widetilde{s}_{N_L}$ is only evaluated on the sets $Y^L_{M_L}$ and $Y^L_{\widetilde{M}_L}$. We then define our estimate of $E[s_h]$ as

$$E_{M_0,\ldots,\widetilde{M}_L}[s_h] = E_{M_0}[s_h - s_{N_1}] + \sum_{\ell=1}^{L-1} E_{M_\ell}[s_{N_\ell} - s_{N_{\ell+1}}] + E_{M_L}[s_{N_L} - \widetilde{s}_{N_L}] + E_{\widetilde{M}_L}[\widetilde{s}_{N_L}]\,. \tag{4.10}$$

Similarly, the estimate of the variance is defined as

$$V_{M_0,\ldots,\widetilde{M}_L}[s_h] = E_{M_0}[\zeta_h - \zeta_{N_1}] + \sum_{\ell=1}^{L-1} E_{M_\ell}[\zeta_{N_\ell} - \zeta_{N_{\ell+1}}] + E_{M_L}[\zeta_{N_L} - \widetilde{\zeta}_{N_L}] + E_{\widetilde{M}_L}[\widetilde{\zeta}_{N_L}]\,, \tag{4.11}$$

where the auxiliary variables are $\zeta_h = (s_h - E_{M_0,\ldots,\widetilde{M}_L}[s_h])^2$, $\zeta_{N_\ell} = (s_{N_\ell} - E_{M_0,\ldots,M_L}[s_h])^2$ for $\ell = 1,\ldots,L$ and $\widetilde{\zeta}_{N_L} = (\widetilde{s}_{N_L} - E_{M_0,\ldots,\widetilde{M}_L}[s_h])^2$. The probabilistic guarantees on the error of the proposed statistics arise from the extension of the CLT results above

$$\lim_{M_0\to\infty}\cdots\lim_{\widetilde{M}_L\to\infty} \Pr\left(\left|E[s_h] - E_{M_0,\ldots,\widetilde{M}_L}[s_h]\right| \le \Delta^E_{M_0,\ldots,\widetilde{M}_L}\right) = \operatorname{erf}\left(\frac{a}{\sqrt{2}}\right),$$

$$\lim_{M_0\to\infty}\cdots\lim_{\widetilde{M}_L\to\infty} \Pr\left(\left|V[s_h] - V_{M_0,\ldots,\widetilde{M}_L}[s_h]\right| \le \Delta^V_{M_0,\ldots,\widetilde{M}_L}\right) = \operatorname{erf}\left(\frac{a}{\sqrt{2}}\right).$$

where the *a posteriori* error bounds read

$$\Delta^E_{M_0,\ldots,\widetilde{M}_L} = a\sqrt{\frac{V_{M_0}[s_h - s_{N_1}]}{M_0} + \sum_{\ell=1}^{L-1} \frac{V_{M_\ell}[s_{N_\ell} - s_{N_{\ell+1}}]}{M_\ell} + \frac{V_{M_L}[s_{N_L} - \widetilde{s}_{N_L}]}{M_L} + \frac{V_{\widetilde{M}_L}[\widetilde{s}_{N_L}]}{\widetilde{M}_L}}\,,$$

(4.12)

$$\Delta^V_{M_0,\ldots,\widetilde{M}_L} = a\sqrt{\frac{V_{M_0}[\zeta_h - \zeta_{N_1}]}{M_0} + \sum_{\ell=1}^{L-1} \frac{V_{M_\ell}[\zeta_{N_\ell} - \zeta_{N_{\ell+1}}]}{M_\ell} + \frac{V_{M_L}[\zeta_{N_L} - \widetilde{\zeta}_{N_L}]}{M_L} + \frac{V_{\widetilde{M}_L}[\widetilde{\zeta}_{N_L}]}{\widetilde{M}_L}}\,.$$

Note that all expectations and variances are MC estimates through the sample sets $Y^\ell_{M_\ell}$ for $0 \le \ell \le L$ and $Y^L_{\widetilde{M}_L}$.

We will refer to the general EIMVR method with a sequence of L RB models as the L-EIMVR method. For clarity of notation, we identify $s_h - s_{N_1}$ as level 0, $\widetilde{s}_{N_L}$ as level $\widetilde{L}$, the subsequent $s_{N_\ell} - s_{N_{\ell+1}}$ as level $\ell \in (0, L-1)$, and $s_{N_L} - \widetilde{s}_{N_L}$ as level L. The method allows us to transfer the computational burden from the higher-fidelity (expensive) outputs to the lower-fidelity (inexpensive) outputs. In particular, we can choose $N_1, N_2, \ldots, N_L$ so as to have $M_0 \ll M_1 \ll \ldots \ll M_L \ll \widetilde{M}_L$. Hence, the number of evaluations of the higher-fidelity outputs are significantly smaller than those of the lower-fidelity outputs, thereby resulting in a significant reduction in the overall computational cost.

4.3. Selection method. We wish to determine the RB dimensions $N_1, N_2, \ldots, N_L$ and the number of samples $M_0, M_1, \ldots, M_L, \widetilde{M}_L$ so as to satisfy the error condition $\Delta^E_{M_0,\ldots,\widetilde{M}_L} \le \epsilon_{\text{tol}}$, while minimizing the computational cost. Obviously, the error condition is satisfied if we take

$$\begin{aligned} a^2\frac{V_{M_0}[s_h - s_{N_1}]}{M_0} &= w_0\epsilon^2_{\text{tol}}\,, \qquad a^2\frac{V_{M_\ell}[s_{N_\ell} - s_{N_{\ell+1}}]}{M_\ell} = w_\ell\epsilon^2_{\text{tol}}, \quad 1 \le \ell \le L-1\,, \\ a^2\frac{V_{M_L}[s_{N_L} - \widetilde{s}_{N_L}]}{M_L} &= w_L\epsilon^2_{\text{tol}}\,, \qquad a^2\frac{V_{\widetilde{M}_L}[\widetilde{s}_{N_L}]}{\widetilde{M}_L} = \widetilde{w}_L\epsilon^2_{\text{tol}}\,, \end{aligned} \tag{4.13}$$

for any given positive weights such that $w_0 + \ldots + w_L + \widetilde{w}_L = 1$. The choice of the weights depends on how we would like to distribute the error among the levels. Furthemore, the terms $v_k(\boldsymbol{y}_m)$ need to be precomputed for a predetermined N_L on a set Y_M for $k = 1, \ldots, K$, $m = 1, \ldots, M$. Hence, both N_L and M are already fixed, and the M samples will be split between levels L and $\widetilde{L}$, that is $Y_M = Y^L_{M_L} \cup Y^L_{\widetilde{M}_L}$.

Let t_{N_ℓ} denote the (online) wall time to compute the RB output $s_{N_\ell}(\boldsymbol{y})$ for $\ell \ge 1$, and t_h denote the wall time to compute the HDG output $s_h(\boldsymbol{\theta}, \boldsymbol{y})$ for any given $\boldsymbol{y} \in \Lambda$. Note that t_{N_ℓ} depends on N_ℓ, while t_h depends on the finite element approximation spaces. Let us also introduce t_α, which refers to the wall time needed to compute $\boldsymbol{\alpha}(\boldsymbol{\theta})$ in (3.3)—the cost of evaluating the sum (3.8) is very small and thus neglected. The total wall time T_L of the L-EIMVR is given by

$$T_L = (t_h + t_{N_1})\, M_0 + \sum_{\ell=1}^{L-1} M_\ell \left(t_{N_\ell} + t_{N_{\ell+1}}\right) + t_{N_L} M_L + t_\alpha. \tag{4.14}$$

We can rewrite this expression by using (4.13) to substitute sample sizes by weights,

$$C_L = \frac{T_L \epsilon_{\text{tol}}^2}{a^2} = \frac{V_{M_0}[s_h - s_{N_1}]}{w_0}\left(t_h + t_{N_1}\right) + \sum_{\ell=1}^{L-1} \frac{V_{M_\ell}[s_{N_\ell} - s_{N_{\ell+1}}]}{w_\ell}\left(t_{N_\ell} + t_{N_{\ell+1}}\right), \\ + \frac{V_{M_L}[s_{N_L} - \widetilde{s}_{N_L}]}{w_L} t_{N_L} + t_\alpha \frac{\epsilon_{\text{tol}}^2}{a^2}. \tag{4.15}$$

We wish to determine $(w_0, \ldots, w_L, \widetilde{w}_L)$ and $(N_1, N_2, \ldots, N_{L-1})$ that minimize C_L. Unfortunately, this is a nonlinear integer optimization problem which is difficult to solve exactly. We thus solve an approximate problem as follows.

We first introduce a test sample set $Y_{\widehat{M}} = \{\widehat{\boldsymbol{y}}_m \in \Lambda, 1 \leq m \leq \widehat{M}\}$ and fix parameter vector $\boldsymbol{\theta} \in \Theta$. We then precompute and store the HDG outputs $s_h(\boldsymbol{\theta}, \widehat{\boldsymbol{y}}_m)$ for $m = 1, \ldots, \widehat{M}$ and the RB outputs $s_N(\boldsymbol{\theta}, \widehat{\boldsymbol{y}}_m)$ for $m = 1, \ldots, \widehat{M}$ and $N = N_L, \ldots, N_{\max}$. In addition, we also precompute and store t_h and t_N for $N = N_L, \ldots, N_{\max}$. For any given strictly decreasing $(L-1)$-tuple $\boldsymbol{I} = (I_1, I_2, \ldots, I_{L-1}) \in [N_L, N_{\max}]^{L-1}$ and valid weights $\boldsymbol{w} = (w_0, \ldots, w_L)$, we evaluate an *a priori* cost function

$$\widehat{C}_L(\boldsymbol{I}, \boldsymbol{w}) = \frac{V_{\widehat{M}}[s_h - s_{I_1}]}{w_0}\left(t_h + t_{I_1}\right) + \sum_{\ell=1}^{L-1} \frac{V_{\widehat{M}}[s_{I_\ell} - s_{I_{\ell+1}}]}{w_\ell}\left(t_{I_\ell} + t_{I_{\ell+1}}\right) + \frac{V_{\widehat{M}}[s_{N_L} - \widetilde{s}_{N_L}]}{w_L} t_{N_L}, \\ \equiv \frac{\widehat{C}_L^0(\boldsymbol{I})}{w_0} + \sum_{\ell=1}^{L-1} \frac{\widehat{C}_L^\ell(\boldsymbol{I})}{w_\ell} + \frac{\widehat{C}_L^L}{w_L}.$$

Here all the variances are computed using the test sample set $Y_{\widehat{M}}$, and the t_α term is dropped as it does not depend on the decision variables. We now set

$$\boldsymbol{N} \equiv (N_1, N_2, \ldots, N_{L-1}) = \arg\min_{\boldsymbol{I}} \widehat{C}_L(\boldsymbol{I}, \boldsymbol{w}(\boldsymbol{I})), \\ \text{s.t.} \quad N_{\max} \geq I_1 > I_2 > \ldots > I_{L-1} \geq N_L \tag{4.16}$$

where the weights $\boldsymbol{w}(\boldsymbol{I})$ are the minimizers of the equivalent cost for any given $(L-1)$-tuple $\boldsymbol{I}$, thus

$$(w_0(\boldsymbol{I}), w_1(\boldsymbol{I}), \ldots, w_L(\boldsymbol{I})) = \arg\min_{\boldsymbol{w}} \widehat{C}_L(\boldsymbol{I}, \boldsymbol{v}), \\ \text{s.t.} \quad \sum_{\ell=0}^{L} w_\ell = 1 - \widetilde{w}_L, \quad w_\ell > 0. \tag{4.17}$$

It only remains to determine the weight $\widetilde{w}_L$ of level $\widetilde{L}$. The samples of this level must come from the precomputed set Y_M, but a fraction of the samples are also needed for level L. We express this condition with an allocation parameter $\beta \in (0, 1)$, such that $\widetilde{M}_L = \beta M$. In order to find the optimal weights, we pursue an iterative process for β, first initializing $\widetilde{w}_L = a^2 V_M[\widetilde{s}_{N_L}]/\epsilon_{\text{tol}}^2 \beta M$. For any β, the KKT conditions for (4.17) yield the optimal weights

for any $(L-1)$-tuple $\boldsymbol{I}$ as

$$w_\ell(\boldsymbol{I}) = \frac{\sqrt{\widehat{C}_L^\ell(\boldsymbol{I})/\widehat{C}_L^0(\boldsymbol{I})}}{\sum_{\ell'=0}^{L}\sqrt{\widehat{C}_L^{\ell'}(\boldsymbol{I})/\widehat{C}_L^0(\boldsymbol{I})}}(1-\widetilde{w}_L), \quad 0 \le \ell \le L. \tag{4.18}$$

The above weights provide a first estimation of $(M_0, \dots, M_L)$, hence only need to update the allocation parameter as $\beta = 1 - M_L/M$ and the weight $\widetilde{w}_L$ with the new β. We then recompute $w_\ell(\boldsymbol{I})$, $0 \le \ell \le L$ with (4.18) and repeat until convergence. We solve the minimization problem (4.16) by simply evaluating the cost function $\widehat{C}_L(\boldsymbol{I}, \boldsymbol{w}(\boldsymbol{I}))$ for all feasible $(L-1)$-tuples, and choosing $(\boldsymbol{I}^*, \boldsymbol{w}^*(\boldsymbol{I}^*))$ that yields the smallest objective value. We finally set $\boldsymbol{N} = \boldsymbol{I}^*$ and $\boldsymbol{w} = \boldsymbol{w}^*(\boldsymbol{I}^*)$.

Nevertheless, if s_{N_L} does not have significant statistical correlation with its EI counterpart $\widetilde{s}_{N_L}$, the selection method described above will not deliver the optimal multilevel configuration. For instance, this situation can occur whenever the new design $\boldsymbol{\theta}$ lies in a zone not properly explored by the $\boldsymbol{\theta}$-training set Θ_I. We overcome this limitation by incorporating the new design $\boldsymbol{\theta}$ to the EI basis using Algorithm 1 if no variance reduction is achieved, that is $V_{\widehat{M}}[s_{N_L} - \widetilde{s}_{N_L}] > V_{\widehat{M}}[s_{N_L}]$. Moreover, if updating the EI basis is less expensive than not performing an update (based on the *a priori* cost), the new design $\boldsymbol{\theta}$ is also added to the EI basis. Once the EI basis is updated we can select the optimal configuration $(\boldsymbol{N}, \boldsymbol{w})$ using the procedure described above.

Having determined the RB dimensions and the weights, we can now proceed with the MC simulations for all levels. We execute the MC processes for all the levels and enforce the error constraint $\Delta^E_{M_0,\dots,\widetilde{M}_L} = \epsilon_{\text{tol}}$ by adding new random parameters to the sample sets until the following inequalities

$$M_L \ge \frac{a^2 V_{M_L}[s_{N_L} - \widetilde{s}_{N_L}]}{w_L^{\boldsymbol{N}^*}\epsilon_{\text{tol}}^2}, \; M_0 \ge \frac{a^2 V_{M_0}[s_h - s_{N_1}]}{w_0^{\boldsymbol{N}^*}\epsilon_{\text{tol}}}, \; M_\ell \ge \frac{a^2 V_{M_\ell}[s_{N_\ell} - s_{N_{\ell+1}}]}{w_\ell^{\boldsymbol{N}^*}\epsilon_{\text{tol}}}, \; \ell = 1, \dots, L-1,$$

are satisfied. Therefore, the sample sets $Y^\ell_{M_\ell}, \ell = 0, \dots, L-1$ are continuously updated during the MC runs, whereas $Y^L_{M_L}$, $Y^L_{\widetilde{M}_L}$ remain fixed. Finally, to provide confidence in the application of the CLT we also need to enforce that M_ℓ are greater than a certain threshold, say 30.

Although we have assumed that the number of levels L is fixed, our approach also allows us to compare the computational costs for several values of L. Hence, we can determine not only the RB dimensions and the weights, but also the optimal number of levels, by repeating the process above with variable L. This analysis provides inexpensive means to determine the optimal multilevel structure for each different design parameter $\boldsymbol{\theta}$. Furthermore, the method can also be improved by constructing the EI basis for different values of N_L, and enforcing the selection method to pick the optimal multilevel structure amongst the optimal multilevel structures for each value of N_L.

4.4. Gradient of estimators. The L-EIMVR method proposed above enables the fast evaluation of output statistics for multiple instantiations of the design parameters. In addition,

it is also desirable to simultaneously compute gradients (and maybe Hessians) of the L-EIMVR estimators using the results in 2.4.

The computation of gradients for expectations is usually addressed by interchanging the gradient and expectation operators, which is valid under certain regularity assumptions [25, 26] of the output functional. The procedure then resorts to regular MC simulation on the gradient of the output to approximate the gradient of the expectation. In our context, it is natural to capitalize the multilevel structure introduced above to compute derivatives of the relevant statistics. We therefore define

$$\begin{aligned}\boldsymbol{\nabla} E_{M_0,\ldots,\widetilde{M}_L}[s_h] &= E_{M_0}[\boldsymbol{\nabla} s_h - \boldsymbol{\nabla} s_{N_1}] + \sum_{\ell=1}^{L-1} E_{M_\ell}[\boldsymbol{\nabla} s_{N_\ell} - \boldsymbol{\nabla} s_{N_{\ell+1}}] \\ &\quad + E_{M_L}[\boldsymbol{\nabla} s_{N_L} - \boldsymbol{\nabla} \widetilde{s}_{N_L}] + E_{\widetilde{M}_L}[\boldsymbol{\nabla} \widetilde{s}_{N_L}]\end{aligned} \tag{4.19}$$

and

$$\begin{aligned}\boldsymbol{\nabla} V_{M_0,\ldots,\widetilde{M}_L}[s_h] &= E_{M_0}[\boldsymbol{\nabla} \zeta_h - \boldsymbol{\nabla} \zeta_{N_1}] + \sum_{\ell=1}^{L-1} E_{M_\ell}[\boldsymbol{\nabla} \zeta_{N_\ell} - \boldsymbol{\nabla} \zeta_{N_{\ell+1}}] \\ &\quad + E_{M_L}[\boldsymbol{\nabla} \zeta_{N_L} - \boldsymbol{\nabla} \widetilde{\zeta}_{N_L}] + E_{\widetilde{M}_L}[\boldsymbol{\nabla} \widetilde{\zeta}_{N_L}]\end{aligned} \tag{4.20}$$

where the gradients for the auxiliary variables read

$$\boldsymbol{\nabla} \zeta_h = 2\sqrt{\zeta_h}\left(\boldsymbol{\nabla} s_h - \boldsymbol{\nabla} E_{M_0,\ldots,\widetilde{M}_L}[s_h]\right),$$

and analogously for ζ_{N_ℓ}, $\widetilde{\zeta}_{N_L}$. The computation of gradients for the HDG and RB outputs is detailed in Section 2.4, whereas for the EI output $\widetilde{s}_{N_L}$ we refer the reader to Section 3. Finally, we can trivially compute *a posteriori* error bounds for each component of the gradient by combining the results in (4.12) with the multilevel expression for the gradients in (4.19)-(4.20).

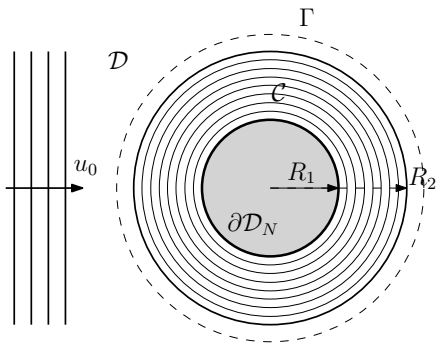
5. Numerical Results. The method described above aims to provide an efficient computation of the statistical outputs and their gradients for any parameters in the parameter space. This method is well suited for analysis of a stochastic system modeled by parametrized stochastic PDEs, when we need to study the behavior of the underlying stochastic system and its sensitivity for many different inputs. The method is also particularly useful for solving stochastic optimization problems, whose objective value (usually a function of the expectation and variance) needs to be evaluated many times in order to find an optimal solution. In this section, we demonstrate our method on an acoustic scattering problem in both the analysis and optimization contexts.

5.1. Problem description. We consider a wave scattering problem as depicted in Figure 1a. Let us consider a region $\mathcal{C}$, comprised between the circles of radii R_1 and R_2, formed by Q disjoint rings of constant (relative) permittivity ε_q, $q = 1, \ldots, Q$, see Figure 1a. An incident plane wave u_0 with wavenumber k interacts with the cylinder of radius R_1, generating a scattering pattern in the domain. If we express the total wave amplitude u_t as the linear

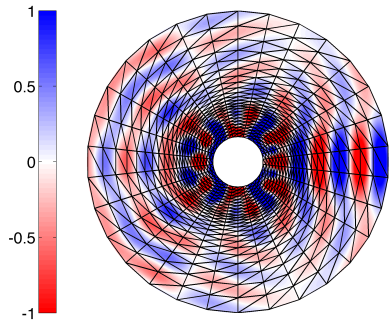
superposition of the incident and scattered fields as $u_t = u_0 + u$, the equations driving the system are

$$
\begin{aligned}
-\Delta u - k^2 \varepsilon u &= \Delta u_0 + k^2 \varepsilon u_0, && \forall \boldsymbol{x} \in \mathcal{D}\ , \\
\boldsymbol{\nabla} u \cdot \boldsymbol{n} &= iku - \mathcal{H} u, && \forall \boldsymbol{x} \in \partial \mathcal{D}_R\ , \\
\boldsymbol{\nabla} u \cdot \boldsymbol{n} &= -\boldsymbol{\nabla} u_0 \cdot \boldsymbol{n}, && \forall \boldsymbol{x} \in \partial \mathcal{D}_N\ .
\end{aligned} \tag{5.1}
$$

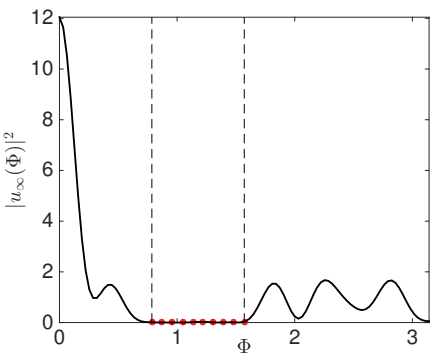
The outer boundary $\mathcal{D}_R$, with mean curvature $\mathcal{H}$, is located farfield, limiting the effect of first-



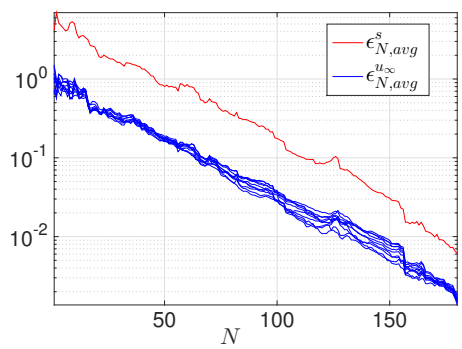
(a) Geometry of the wave scattering problem.



(b) Scattered field for $\boldsymbol{\theta}_{\mathrm{d}}^*$ and $\boldsymbol{y} = 0$, using 1380 elements of order $p = 4$ for $\mathcal{N} = 31200$ dof.



(c) Radar cross section vs $\Phi \in [0, \pi]$ for $\boldsymbol{\theta}_{\mathrm{d}}^*$ and $\boldsymbol{y} = 0$. Target zone and observation angles are highlighted.



(d) Average farfield pattern error and cumulative RCS error for a test set vs size of RB N.

Figure 1: Problem specification, optimal deterministic solution and RCS distribution

order radiating boundary conditions on the numerical simulation near the cylinder, whereas the inner boundary $\mathcal{D}_N$ guarantees total wave reflection (sound-hard condition). The objective is to design a material configuration that produces a reduced farfield radiation pattern on a specific direction. Furthermore, we seek a robust material distribution by introducing uncertainty in the permittivity field to account for errors in the manufacturing process,

$$
\varepsilon_q = \theta_q + y_q, \qquad \theta_q \in [\theta_{LB}, \theta_{UB}], \quad y_q \sim N(0; \sigma^2). \tag{5.2}
$$

The farfield radiation pattern can be computed with the expression

$$u_\infty(\boldsymbol{\theta}, \boldsymbol{y}; \Phi) = \int_\Gamma u \frac{\partial e^{-ik\Phi\cdot\boldsymbol{x}}}{\partial \boldsymbol{n}} - \frac{\partial u}{\partial \boldsymbol{n}} e^{-ik\Phi\cdot\boldsymbol{x}} = \ell^\dagger(u(\boldsymbol{\theta}, \boldsymbol{y})), \tag{5.3}$$

evaluated on the curve Γ located at $R_2 + (R_2 - R_1)/Q$. The strength of the radiating waves, known as radar cross section (RCS), is therefore given by $\left|u_\infty(\Phi)\right|^2$. This quantity measures the detectability of the scattering cylinder with a radar. The quantity of interest for this example is the cumulative radar cross section for multiple observation angles

$$s(\boldsymbol{\theta}, \boldsymbol{y}) = \sum_{b=1}^{B} \left|u_\infty(\boldsymbol{\theta}, \boldsymbol{y}; \Phi^b)\right|^2. \tag{5.4}$$

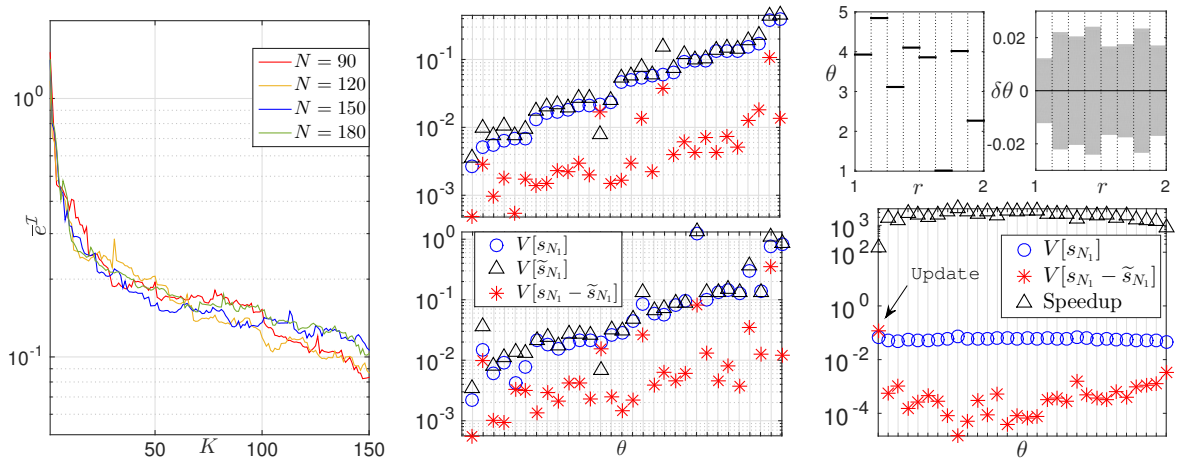
The parameters used for this example are $k = 4$, $R_1 = 1$, $R_2 = 2$, $Q = 8$, whereas the outer boundary $\mathcal{D}_R$ is modeled as a concentric circle of radius 6. We adopt $(\theta_{LB}, \theta_{UB}) = (1, 5)$, $\sigma = 0.03$ for the material properties and fabrication error, and aim to minimize the cumulative RCS on $B = 10$ observation angles within $\Phi \in [\pi/4, \pi/2]$. The physical domain is discretized into a triangular mesh of 1380 elements as shown in Figure 1b and polynomials of degree $p = 4$ are used to represent the numerical solution u_h. Furthermore, Figure 1b also depicts the numerical solution obtained using the HDG method for the optimal deterministic ($\boldsymbol{y} = 0$) design $\boldsymbol{\theta}^*_{\mathrm{d}}$. The radar cross section distribution for this design is shown in Figure 1c, exhibiting low values on the region of interest.

We develop a RB for the problem (5.1) and the linear output (5.3) jointly over $(\boldsymbol{\theta}, \boldsymbol{y}) \in \Theta \times \Lambda$. In Figure 1d we show the convergence of the farfield pattern for all angles considered, together with the convergence of the cumulative RCS, computed as the average errors $\epsilon^{u_\infty}_{N,\mathrm{avg}}, \epsilon^{s}_{N,\mathrm{avg}}$ over a testing set Y_{test} of 100 samples with precomputed HDG solutions. The larger error of the output s compared to the farfield radiation pattern as a function of the RB size is a consequence of its quadratic nature.

Secondly, we construct an empirical interpolation basis using Algorithm 1. We monitor the mean interpolation error $\overline{e}_i$ for all sample points $\boldsymbol{\theta}_i$ that have not been included in the basis as an indicator of the quality of the interpolant. A key point is the model used to construct the interpolation basis, which needs to be determined beforehand. We construct the interpolation basis for several reduced basis sizes, setting $K = 150$, and show $\overline{e}_i$ averaged over $\boldsymbol{\theta}_i \in \Theta_I$ that are not in the basis, which we define as $\overline{e}^{\mathcal{I}}$, in Figure 2a. The convergence of the EI basis is similar for all cases, implying that the decision upon the reduced basis model for which we develop an EI basis should be independent of the EI procedure.

5.2. Features of L-EIMVR. Firstly, it is important to examine some important features of the L-EIMVR method. To that end, we select for each case a region of designs that best illustrates the desired property. To compare the efficiency of L-EIMVR with regular MC we analyze the speedup $\pi_{MC} = t_h M_{MC}/T_L$, where M_{MC} is the amount of samples needed for MC to achieve the same standard error as L-EIMVR, that is $a\sqrt{V_{M_{MC}}[s_h]/M_{MC}} = \Delta^E_{M_0,\ldots,\widetilde{M}_L}$.

5.2.1. Reduction in variance. To show the benefit of including the EI term in terms of variance reduction, we compute multiple realizations of the output for several design parameters chosen at random and compare $V[s_{N_1}]$ with $V[s_{N_1} - \widetilde{s}_{N_1}]$ and $V[\widetilde{s}_{N_1}]$. Results are



(a) Averaged error $\bar{e}^{\mathcal{I}}$ vs size of EI basis.

(b) Reduction in variance for a random set of $\boldsymbol{\theta}$ and $N_1 = 90$ (top), $N_1 = 180$ (bottom).

(c) Initial design (top left), variations explored (top right) and results (bottom) for EI updates.

Figure 2: Convergence of empirical interpolation, variance reduction and EI updates

collected in Figure 2b for $N_1 = 90$ and $N_1 = 180$. Even though the set of $\boldsymbol{\theta}$ is small, it suffices to show that further variance reduction can be achieved with the introduction of the EI level. Indeed, since $\widetilde{s}_{N_1}$ is precomputed offline, for the same amount of samples for s_{N_1} the error attained will be smaller for the EI case as $V[s_{N_1} - \widetilde{s}_{N_1}] < V[s_{N_1}]$. Note that when exploring a wide range of designs there is no optimal way of choosing N_1, since the amount of variance reduction can exhibit large variations for different N_1 across designs.

5.2.2. EI updates. We now illustrate the necessity of allowing updates on the EI basis. To that end, small perturbations of an initial design $\boldsymbol{\theta}_0$ are explored, see Figure 2c (top). For the sake of clarity we set $L = 1$ with $N_1 = 180$, so we simplify the selection process to only determining the weights. In Figure 2c (bottom) we show the reduction in variance for $s_{N_1} - \widetilde{s}_{N_1}$ before and after the EI update. The EI basis is unable to approximate the initial design due to poor correlation ($\rho(s_{N_1}, \widetilde{s}_{N_1}) = 0.014$) causing an increase in variance. This situation is corrected by updating the EI basis, which enables for a significant reduction of variance (2-3 orders of magnitude) in the subsequent designs by increasing the correlation between both outputs.

The speedup π_{MC} is also presented, showing that the EI update performed for $\boldsymbol{\theta}_0$ greatly impacts the computations for neighbors of $\boldsymbol{\theta}_0$. Indeed, after the EI update the 1-EIMVR achieves an extra order of magnitude of speedup with respect to MC for the same *a posteriori* error.

5.3. Stochastic optimization. The original stochastic optimization problem (SP) is replaced by the following stochastic optimization problem

$$\text{(SP-EIMVR)} \qquad \min_{\boldsymbol{\theta}\in\Theta} \widetilde{F}_\gamma = E_{M_0,\ldots,\widetilde{M}_L}[s(\boldsymbol{\theta},\boldsymbol{y})] + \gamma\sqrt{V_{M_0,\ldots,\widetilde{M}_L}[s(\boldsymbol{\theta},\boldsymbol{y})]}.$$

The optimal design for the deterministic problem is $\boldsymbol{\theta}^*_{\mathrm{d}}$, generating the scattered field and radar cross section shown in Figures 1b-1c, and the purpose is to find a new design $\boldsymbol{\theta}$ less sensitive to material imperfections ($\boldsymbol{y} \neq 0$). The optimization algorithm used is the implementation of preconditioned truncated Newton [13] available in the `nlopt` [22] optimization package, with multiple starts to search for the local optima —results correspond to a particular initialization.

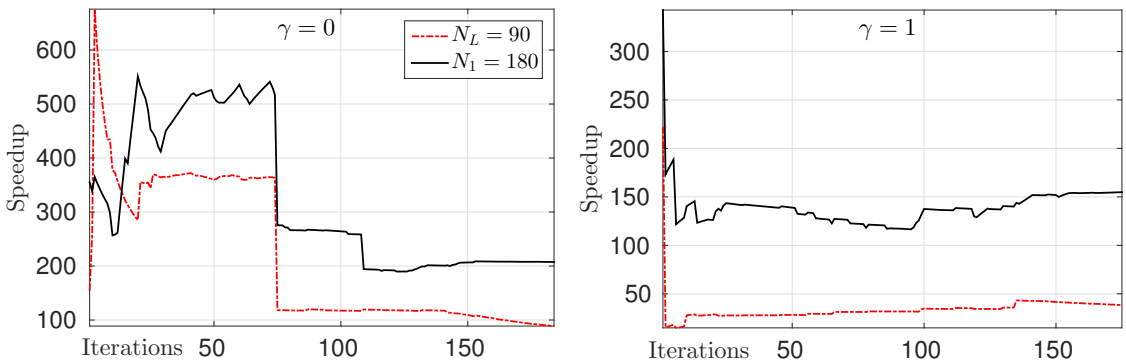


Figure 3: Accumulated speedup for EI models $N_L = 90$ and $N_1 = 180$ with respect to MC.

The first case corresponds to $\gamma = 0$, thus minimizing the expected value of the functional only. We depict the accumulated speedup, that is the ratio of accumulated costs at each iteration, in Figure 3 (left) for $N_L = 90$ and $N_1 = 180$. For this results we have used $N_1 = 180$ to compute the optimal design $\boldsymbol{\theta}^*_0$, and then evaluated the L-EIMVR with $N_L = 90$ on the same set of designs to have fair comparisons. Note that there is a large variability in the accumulated speedups with respect to MC, and that $L-$EIMVR converges to the optimum 100-200 faster than MC simulation. For this particular example $N_1 = 180$ provides a greater acceleration, although it can be difficult to predict *a priori*, especially in wave propagation problems for which the observable outputs exhibit large variations for small changes on the design parameters. The results are similar for $\gamma = 1$, also shown in Figure 3 (right) for an optimal design $\boldsymbol{\theta}^*_1$.

Finally, we compare the performances of 1-EIMVR and the L-MVR estimator introduced in [44]. The nature of the L-EIMVR estimator makes it particularly suitable for cases where many expectations need to be computed, and especially when most of the cost lies on the coarsest level. We restrict the L-EIMVR method to $L = 1$ with $N_1 = 180$, whereas we let the selection algorithm introduced in [44] choose the optimal number of levels for L-MVR, and compute the online wall time of both methods to achieve the same *a posteriori* error. In Figure 4 we show the accumulated speedup for both $\gamma = 0$ and $\gamma = 1$, that is the ratio

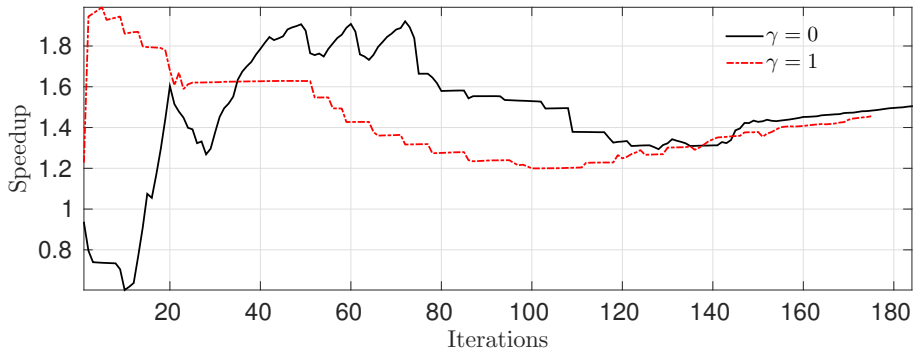


Figure 4: Accumulated speedup 1-EIMVR ($N_1 = 180$) with respect to L-MVR.

of accumulated L-MVR and 1-EIMVR costs. The 1-EIMVR method outperforms L-MVR since the cost for the last level in the 1-EIMVR case is minimal. The accumulated speedup of 1-EIMVR is 1.5 ($\gamma = 0$) and 1.46 ($\gamma = 1$), that is the same tolerance is achieved with around 2/3 of the L-MVR cost. These savings correspond to only one optimization run, but in general multiple initializations are needed to attain the optimum. Hence, if we sequentially reuse the EI basis for different initializations further computational savings with respect to L-MVR (and obviously MC) may be attained. The fact that the EI basis is dynamically adapted to the problem makes the L-EIMVR method attractive for optimization, as opposed to L-MVR and MC which treat every objective function evaluation independently.

Deterministic	$s_h(\boldsymbol{\theta}_{\mathrm{d}}^*, \mathbf{0}) = \min_{\boldsymbol{\theta}\in\Theta} s_h(\boldsymbol{\theta}, \mathbf{0}) = 1.0688\text{e-}1$	
Stochastic (MC)	$E_M[s_h(\boldsymbol{\theta}, \boldsymbol{y})] \pm \Delta_M^E$	$\sqrt{V_M[s_h(\boldsymbol{\theta}, \boldsymbol{y})]} \pm \Delta_M^{\sqrt{V}}$
$\boldsymbol{\theta}_{\mathrm{d}}^*$	2.14e-1 $\pm$ 2.5e-3 a	2.49e-1 $\pm$ 8.6e-3 a
$\boldsymbol{\theta}_0^*$	1.70e-1 $\pm$ 8.7e-4 a	8.69e-2 $\pm$ 5.1e-3 a
$\boldsymbol{\theta}_1^*$	1.79e-1 $\pm$ 4.8e-4 a	4.78e-2 $\pm$ 1.9e-3 a

Table 1: Expectation and variance of output evaluated at deterministic and stochastic optima for $M = 10^4$ MC realizations of random perturbations, for generic confidence level $a > 0$.

In order to assess the quality of the solutions, we generate random noise and analyse the robustness of the various solutions evaluating (SP-MC). Results are collected in Table 1 for designs $\boldsymbol{\theta}_{\mathrm{d}}^*$, $\boldsymbol{\theta}_0^*$ and $\boldsymbol{\theta}_1^*$ and $M = 10^4$ samples of noise $\boldsymbol{y}$. The error bounds for the expectation and the standard deviation are computed with the standard results from limiting distributions, that is

$$\Delta_M^E(\boldsymbol{\theta}) = a\sqrt{\frac{V_M[s_h(\boldsymbol{\theta}, \boldsymbol{y})]}{M}}, \qquad \Delta_M^{\sqrt{V}}(\boldsymbol{\theta}) = a\sqrt{\frac{V_M[s_h(\boldsymbol{\theta}, \boldsymbol{y})](\kappa - 1)}{4M}}$$

where κ is the fourth standarized moment, or kurtosis, and a refers to the level of confidence for the CLT. The designs $\boldsymbol{\theta}_\gamma^*$ are consistent with respect to the corresponding objective function $\widetilde{F}_\gamma$, since $\widetilde{F}_{\gamma=0}(\boldsymbol{\theta}_0^*) < \widetilde{F}_{\gamma=0}(\boldsymbol{\theta}_1^*) < \widetilde{F}_{\gamma=0}(\boldsymbol{\theta}_{\mathrm{d}}^*)$ and $\widetilde{F}_{\gamma=1}(\boldsymbol{\theta}_1^*) < \widetilde{F}_{\gamma=1}(\boldsymbol{\theta}_0^*) < \widetilde{F}_{\gamma=1}(\boldsymbol{\theta}_{\mathrm{d}}^*)$. In addition

we observe considerable degradation of the deterministic optimum when noise is included, see $s_h(\boldsymbol{\theta}_{\rm d}^*)/\widetilde{F}_{\gamma=1}(\boldsymbol{\theta}_{\rm d}^*) < 0.25$, that is the deterministic optimum is not robust in the presence of randomness as it exhibits a large variance. Conversely, the non-deterministic optimum $\boldsymbol{\theta}_1^*$ is a robust design as the objective function for $\gamma = 1$ is already smaller than $\sqrt{V_M[s_h(\boldsymbol{\theta}_{\rm d}^*, \boldsymbol{y})]}$, with a ratio of standard deviations of $\sqrt{V_M[s_h(\boldsymbol{\theta}_1^*, \boldsymbol{y})]}/\sqrt{V_M[s_h(\boldsymbol{\theta}_{\rm d}^*, \boldsymbol{y})]} < 0.2$. We therefore see the necessity of performing stochastic optimization to attain robust designs that may not be achieved if deterministic optimization is pursued.

6. Conclusions. We have presented an empirical interpolation and model-variance reduction method for computing statistical outputs of stochastic elliptic PDEs. We first reviewed the construction of a reduced basis for the hybridizable discontinuous Galerkin method, and the computation of gradients in the reduced basis setting. We next applied the empirical interpolation to develop a coefficient-function approximation of the RB output. We then incorporated the RB and EI outputs into the multilevel control variate framework that exploits the statistical correlation between the RB and EI approximations and the high-fidelity HDG discretization to accelerate the convergence rate of the MC simulations by several orders of magnitude. Moreover, we have combined the multilevel structure with the computation of gradients to propose estimates of the gradients of the output statistics. In addition we introduced *a posteriori* error bounds for the estimates of the statistical outputs together with a selection method for efficiently choosing the RB dimensions, the weights, and the sample sizes. We presented numerical results for a scattering problem, where we have shown that the present method is several orders of magnitude faster than the regular MC method for the same *a posteriori* error bound.

We conclude the paper by pointing out several possible extensions and directions for further research. In terms of the EIMVR method itself, there are a number of ideas that can be pursued: (1) enable online updates on the reduced basis to better approximate regions on the parameter space that are not properly represented by the surrogate, thus not achieving sufficient variance reduction in level 0 of the estimator (4.10); (2) devise estimators with multiple levels of EI outputs, which can increase the competitiveness of the method by introducing additional levels with small variance at virtually no extra online cost; (3) capitalize Automatic Differentiation methods for a faster evaluation of the gradients [17]; and (4) examine randomized quasi-Monte Carlo approaches not only for a faster sampling convergence of the estimators [15], but also for an improved exploration of the stochastic space for the empirical interpolation.

Finally, we would like to compare the performance of the method with existing approaches in stochastic optimization that combine the RB method and sparse grids [9, 8]. Furthermore, it woul also be interesting to analyze how the L-EIMVR method can be applied in the stochastic approximation framework, leveraging the research and results in stochastic gradient methods with the fast computation of output statistics of L-EIMVR. We would like to apply this methodology to fabrication adaptivity problems [40, 24, 28], where manufacturing tolerances must be addressed to ensure robust designs.

Acknowledgements. We would like to thank Dr. Xevi Roca, Professor Youssef Marzouk and Professor Robert Freund for countless fruitful conversations, suggestions and comments.

REFERENCES

[1] Maxime Barrault, Yvon Maday, Ngoc-Cuong Nguyen, and Anthony T Patera, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique, 339 (2004), pp. 667–672.

[2] Andrea Barth, Christoph Schwab, and Nathaniel Zollinger, *Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients*, Numerische Mathematik, 119 (2011), pp. 123–161.

[3] Güzin Bayraksan and David P Morton, *Assessing solution quality in stochastic programs*, Mathematical Programming, 108 (2006), pp. 495–514.

[4] Albert Benveniste, Michel Métivier, and Pierre Priouret, *Adaptive algorithms and stochastic approximations*, vol. 22, Springer Science & Business Media, 2012.

[5] Sébastien Boyaval, *A fast Monte–Carlo method with a reduced basis of control variates applied to uncertainty propagation and bayesian estimation*, Computer Methods in Applied Mechanics and Engineering, 241 (2012), pp. 190–205.

[6] Sébastien Boyaval, Claude Le Bris, Tony Lelièvre, Yvon Maday, Ngoc-Cuong Nguyen, and Anthony T Patera, *Reduced basis techniques for stochastic problems*, Archives of Computational methods in Engineering, 17 (2010), pp. 435–454.

[7] Russel E Caflisch, *Monte Carlo and quasi-Monte Carlo methods*, Acta numerica, 7 (1998), pp. 1–49.

[8] Peng Chen and Alfio Quarteroni, *Weighted reduced basis method for stochastic optimal control problems with elliptic pde constraint*, SIAM/ASA Journal on Uncertainty Quantification, 2 (2014), pp. 364–396.

[9] Peng Chen, Alfio Quarteroni, and Gianluigi Rozza, *Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by stokes equations*, Numerische Mathematik, (2015), pp. 1–36.

[10] ———, *Reduced order methods for uncertainty quantification problems*, ETH Report 03, Submitted, (2015).

[11] Philippe G Ciarlet, *The finite element method for elliptic problems*, Elsevier, 1978.

[12] K Andrew Cliffe, Michael B Giles, Robert Scheichl, and Aretha L Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Computing and Visualization in Science, 14 (2011), pp. 3–15.

[13] Ron S Dembo and Trond Steihaug, *Truncated-Newton algorithms for large-scale unconstrained optimization*, Mathematical Programming, 26 (1983), pp. 190–212.

[14] Michael B Giles, *Multilevel Monte Carlo path simulation*, Operations Research, 56 (2008), pp. 607–617.

[15] Michael B Giles and Ben J Waterhouse, *Multilevel quasi-Monte Carlo path simulation*, Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics, (2009), pp. 165–181.

[16] Martin A Grepl, *Reduced-basis approximations for time-dependent partial differential equations: application to optimal control*, PhD thesis, Massachusetts Institute of Technology, 2005.

[17] Andreas Griewank and Andrea Walther, *Evaluating derivatives: principles and techniques of algorithmic differentiation*, SIAM, 2008.

[18] Bernard Haasdonk, Karsten Urban, and Bernhard Wieland, *Reduced basis methods for parameterized partial differential equations with stochastic influences using the Karhunen–Loève expansion*, SIAM/ASA Journal on Uncertainty Quantification, 1 (2013), pp. 79–105.

[19] John M Hammersley, David C Handscomb, and George Weiss, *Monte Carlo methods*, Physics Today, 18 (1965), p. 55.

[20] Stefan Heinrich, *Multilevel monte carlo methods*, in Large-scale scientific computing, Springer, 2001, pp. 58–67.

[21] Stefan Heinrich and Eugène Sindambiwe, *Monte Carlo complexity of parametric integration*, Journal of Complexity, 15 (1999), pp. 317–341.

[22] Steven G Johnson, *The NLopt nonlinear-optimization package*, http://ab-initio.mit.edu/nlopt.

[23] Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello, *The sample average approximation method for stochastic discrete optimization*, SIAM Journal on Optimization, 12 (2002), pp. 479–502.

[24] Boyan Stefanov Lazarov, Mattias Schevenels, and Ole Sigmund, *Topology optimization with geometric uncertainties by perturbation techniques*, International Journal for Numerical Methods in

Engineering, 90 (2012), pp. 1321–1336.

[25] Pierre L'Ecuyer, *A unified view of the IPA, SF, and LR gradient estimation techniques*, Management Science, 36 (1990), pp. 1364–1383.

[26] ———, *Note: On the interchange of derivative and expectation for likelihood ratio derivative estimators*, Management Science, 41 (1995), pp. 738–747.

[27] Wai-Kei Mak, David P Morton, and R Kevin Wood, *Monte Carlo bounding techniques for determining solution quality in stochastic programs*, Operations Research Letters, 24 (1999), pp. 47–56.

[28] Han Men, Robert M Freund, Ngoc-Cuong Nguyen, Joel Saa-Seoane, and Jaime Peraire, *Fabrication-adaptive optimization with an application to photonic crystal design*, Operations Research, 62 (2014), pp. 418–434.

[29] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization, 19 (2009), pp. 1574–1609.

[30] Leo WT Ng, Dinh BP Huynh, and Karen Willcox, *Multifidelity Uncertainty Propagation for Optimization Under Uncertainty*, in 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2012.

[31] Ngoc-Cuong Nguyen, Karen Veroy, and Anthony T Patera, *Certified real-time solution of parametrized partial differential equations*, in Handbook of Materials Modeling, Springer, 2005, pp. 1529–1564.

[32] Ahmed K Noor and Jeanne M Peters, *Reduced basis technique for nonlinear analysis of structures*, AIAA Journal, 18 (1980), pp. 455–462.

[33] Niles A Pierce and Michael B Giles, *Adjoint recovery of superconvergent functionals from PDE approximations*, SIAM review, 42 (2000), pp. 247–264.

[34] Boris T Polyak and Anatoli B Juditsky, *Acceleration of stochastic approximation by averaging*, SIAM Journal on Control and Optimization, 30 (1992), pp. 838–855.

[35] Christophe Prudhomme, Dimitrios V Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T Patera, and Gabriel Turinici, *Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods*, Journal of Fluids Engineering, 124 (2002), pp. 70–80.

[36] Herbert Robbins and Sutton Monro, *A stochastic approximation method*, The annals of mathematical statistics, (1951), pp. 400–407.

[37] Gianluigi Rozza, DBP Huynh, and Anthony T Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Archives of Computational Methods in Engineering, 15 (2008), pp. 229–275.

[38] Reuven Y Rubinstein and Alexander Shapiro, *Discrete event systems: Sensitivity analysis and stochastic optimization by the score function method*, vol. 346, Wiley New York, 1993.

[39] Tjendera Santoso, Shabbir Ahmed, Marc Goetschalckx, and Alexander Shapiro, *A stochastic programming approach for supply chain network design under uncertainty*, European Journal of Operational Research, 167 (2005), pp. 96–115.

[40] Mattias Schevenels, Boyan Stefanov Lazarov, and Ole Sigmund, *Robust topology optimization accounting for spatially varying manufacturing errors*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 3613–3627.

[41] Alexander Shapiro, *Asymptotic analysis of stochastic programs*, Annals of Operations Research, 30 (1991), pp. 169–186.

[42] ———, *Monte Carlo sampling methods*, Handbooks in operations research and management science, 10 (2003), pp. 353–425.

[43] Aretha L Teckentrup, Robert Scheichl, Michael B Giles, and Elisabeth Ullmann, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Numerische Mathematik, 125 (2013), pp. 569–600.

[44] Ferran Vidal-Codina, Ngoc-Cuong Nguyen, Michael B Giles, and Jaime Peraire, *A model and variance reduction method for computing statistical outputs of stochastic elliptic partial differential equations*, Journal of Computational Physics, 297 (2015), pp. 700–720.