On the use of ANOVA expansions in reduced basis methods for high-dimensional parametric partial differential equations

Jan S. Hesthaven · Shun Zhang

Abstract. We propose two different improvements of reduced basis (RB) methods to enable the efficient and accurate evaluation of an output functional based on the numerical solution of parametrized partial differential equations with a possibly high-dimensional parameter space. The element that combines these two techniques is that they both utilize ANOVA expansions to achieve the improvements.

The first method is a three-step RB-ANOVA-RB method, aiming at using a combination of reduced basis methods and ANOVA expansions to effectively compress the parameter space without impact the accuracy of the output of interest. This is achieved by first building a low-accuracy reduced model for the full high-dimensional parametric problem. This model is used to recover an approximate ANOVA expansion for the output functional at marginal cost, allowing the estimation of the sensitivity of the output functional to parameter variation and enabling a subsequent compression of the parameter space. A new accurate reduced model can then be constructed for the compressed parametric problem at a substantially lower computational cost than for the full problem.

In the second approach we explore the ANOVA expansion to drive an hp reduced basis method. This is initiated by setting up a maximum number of reduced bases that can be afforded during the online stage. If the offline greedy procedure for a given parameter domain converges with equal or less than the maximum bases, the offline algorithm stops. Otherwise, an approximate ANOVA expansion is performed for the output functional. The parameter domain is decomposed into several subdomains where the most important parameters according to the ANOVA expansion are split. The offline greedy algorithms are performed in these parameter subdomains. The algorithm is applied recursively until the offline greedy algorithms converge across all parameter subdomains.

We demonstrate the accuracy, efficiency, and generality of these two approaches through a number of test cases.

Keywords reduced basis method · RB · ANOVA · high-dimentional parameter space · hp method

Division of Applied Mathematics, Brown University, Providence, RI 02912.
E-mail: Jan.Hesthaven@Brown.edu, E-mail: Shun_Zhang@Brown.edu

1 Introduction

In recent years, there have been a growing interest in the efficient and accurate solution of parametrized problems often with a high-dimensional parameter spaces. Applications can be found in areas such as optimization, design, control, uncertainty quantification and so on. While there are many aspects of such problems, we shall here focus on the development of methods of reduced complexity to rapidly evaluate an output of interest over variations in parameter space. If the parameter space is indeed high-dimensional it is clear that the curse of dimensionality is a substantial challenge in such situations and ideas to increase efficiency is required.

Among several alternatives, we shall in this work focus on the reduced basis (RB) method as an efficient method to address parametric problems in many-query and real-time scenarios, see e.g. [14,15,12,13]. However, the ideas presented here may well apply to other situations and techniques also.

We use $\boldsymbol{\mu} = (\boldsymbol{\mu}_{(1)}, \cdots, \boldsymbol{\mu}_{(p)})$ to denote a parameter vector in some parameter domain $\mathcal{D} \subset \mathbb{R}^p$, where $\boldsymbol{\mu}_{(i)}$ is the (i)-th component of the parameter vector $\boldsymbol{\mu}$. The fundamental problem is the following: Let X be a suited functional space. For any parameter vector $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^p$, we wish to evaluate an output functional of interests

$$s(\boldsymbol{\mu}) = \ell(u(\boldsymbol{\mu}); \boldsymbol{\mu}), \tag{1.1}$$

where $u(\boldsymbol{\mu}) \in X$ is the solution of

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in X \tag{1.2}$$

for some parameter dependent bilinear and linear forms a and f.

In [9], we propose an efficient greedy algorithm to reduce the overall cost of the construction of the RB for the many parameter without impacting the overall accuracy of the model. However, when the number of parameters p is large, many basis elements are typically required to fully resolve the problem, making the overall computational cost, both online and offline, significant or perhaps even prohibitive.

In this paper, we consider two different improvements of reduced basis (RB) methods to enable the efficient and accurate evaluation of an output functional based on the numerical solution of parametrized partial differential equations with a possibly high-dimensional parameter space. The element that combines these two techniques is that they both utilize ANOVA expansions to achieve the improvements.

For the first approach, we observe that for many complex applications one often finds that only a small number of parameters are important for evaluation of the output functional while the remaining parameters only have limited impact on the output of interests. We shall explore the ability of the ANOVA expansion to systematically determine the relative importance of parameters by evaluating the sensitivity of an output of interest on each parameter. The value of this is discussed in [6], where a two step method is developed for ordinary differential equations with many parameters. First, the ANOVA expansion is computed based on coarse approximation of the full parameter problem and the parameter space is subsequently compressed through the sensitivity analysis, allowing for the development of a model based only on the important parameters without impacting the predictive accuracy of the model.

Seeking to follow a similar approach, we need evaluate the output functional on many quadrature points to compute the integrals in the ANOVA expansions over the parameter space. Traditionally, this is done by using standard discretization methods (for example finite element methods) to solve the problem over a large sample in parameter space. Thus, it quickly becomes costly to compute the ANOVA expansion. To overcome this bottleneck, we propose in this paper a new three step method, an RB-ANOVA-RB method. First, we build a relatively inaccurate RB model for the full parameter problem, thus introducing a cheap input-to-output evaluation. Based on this RB model, we can now compute an ANOVA expansion of the output of interest at little cost. This expansion may not be particularly accurate, but the accuracy suffices to analyze the sensitivity of all parameters and enables us to retain the important parameters, and freeze those of less relative importance. For the reduced parameter problem, we then build a new reduced model at limited cost and both the offline and the online costs of the new reduced RB model will be cheaper.

The second approach is an hp reduced basis method. The goal in this approach is to have a decomposition of the original parameter domain, such that only a fixed number of reduced bases are needed in each domain to achieve the prescribed tolerance. Thus, the online cost of evaluation of a functional is under control, even for large parameter domains.

There are several hp types of methods available. The algorithm proposed in [2] first does an "h" decomposition of the parameter domain into subdomains, and then reduced bases are build on those subdomains. The shortcoming of the method is that since the "h" and "p" steps are separate and performed only once, there are cases where some subdomains are not small enough and many reduced bases are still needed, or subdomains may be too small, requiring that only a small number of basis elements are needed. Both cases are clearly sub-optimal. Another hp approached is introduced for empirical interpolation methods, see [4,3] . The idea is that if the method does not converges in N bases, the parameter domain is decomposed into 2^p subdomains, and a new basis is build in these 2^p subdomains. The algorithm is performed recursively. The good property of the algorithm is that at termination, each subdomain will need less than N bases. The shortcoming of the algorithm is that when p is relatively large, 2^p will be a large number and hence computationally expensive while including many unnecessary cases.

Like the hp algorithm for the empirical interpolation method [4,3] , our hp reduced basis method first defines a maximum number of reduced bases that is affordable during the online stage. If the offline greedy procedure for a given parameter domain converges with equal or less than the maximum bases, the offline algorithm terminates. Otherwise, rather than decomposing the domain into 2^p subdomains, we only decompose the parameter domain into 2^K subdomains, where $K \leq p$ is a small number such that 2^K subdomains are not too large. An approximate ANOVA expansion is developed for the output functional to determine the K most important parameter directions. The parameter domain is decomposed into several subdomains where the most important K parameter directions are split. The offline greedy algorithms are performed in those parameter subdomains and the algorithm is implemented recursively until the offline greedy algorithm converges in all parameter subdomains . In our new hp approach, the computational cost of the online step is manageable compared to the hp algorithm developed in [2].

The rest of paper is organized as follows. In Section 2, we offer a brief introduction to reduced basis methods and Section 3 introduces the ANOVA expansions and recalls the Smolyak sparse grid quadrature, required to evaluate the numerical integrals in the ANOVA expansions. In Section 4, the three step RB-ANOVA-RB method is introduced and in Section 5 we presents numerical experiments to demonstrate the efficiency and robustness of this approach. The ANOVA-based hp reduced method is introduced in Section 6 with numerical experiments presented in Section 7. Some concluding remarks are offered in Section 8.

2 Reduced basis methods

Let us first give a brief introduction to the reduced basis method to offer context for the subsequent discussion. For simplicity, we only discuss RBM for affine elliptic PDEs here; for non-affine and other types of PDEs, see e.g. [13] and [14].

Assume that the parameter dependent problem (1.2) satisfies an affine assumption, i.e.,

$$a(u,v;\boldsymbol{\mu}) = \sum_{i=1}^{Q_a} \Theta_i^a(\boldsymbol{\mu}) a_i(u,v), \quad f(v;\boldsymbol{\mu}) = \sum_{i=1}^{Q_f} \Theta_i^f(\boldsymbol{\mu}) f_i(v),$$
$$\text{and} \quad \ell(v;\boldsymbol{\mu}) = \sum_{i=1}^{Q_\ell} \Theta_i^\ell(\boldsymbol{\mu}) \ell_i(v), \tag{2.3}$$

where Θ_i^a, Θ_i^f, and Θ_i^ℓ are $\boldsymbol{\mu}$-dependent functions, and a_i, f_i, ℓ_i are $\boldsymbol{\mu}$-independent forms.

Let X^{fe} be a finite element discretization subspace of X. For a fixed parameter $\boldsymbol{\mu} \in \mathcal{D}$, let $u^{fe}(\boldsymbol{\mu}) \in X^{fe}$ be the numerical solution of the following Galerkin problem,

$$a(u^{fe}(\boldsymbol{\mu}),v;\boldsymbol{\mu}) = f(v;\boldsymbol{\mu}), \quad \forall v \in X^{fe}, \tag{2.4}$$

and let $s^{fe}(\boldsymbol{\mu}) = \ell(u^{fe}(\boldsymbol{\mu});\boldsymbol{\mu})$ be the corresponding output of interest.

The approximation problem (2.4) is assumed to be well-posed such that the inf-sup stability is satisfied for $\boldsymbol{\mu}$-dependent positive constant $\beta^{fe}(\boldsymbol{\mu})$:

$$\beta^{fe}(\boldsymbol{\mu}) = \inf_{u \in X^{fe}} \sup_{v \in X^{fe}} \frac{a(u,v;\boldsymbol{\mu})}{\|u\|_{X^{fe}} \|v\|_{X^{fe}}}, \tag{2.5}$$

where $\|\cdot\|_{X^{fe}}$ is the norm associated with the space X^{fe}, respectively.

For a collection of N parameters $S_N = \{\boldsymbol{\mu}^1, \cdots, \boldsymbol{\mu}^N\}$ in the parameter domain $\mathcal{D} \subset \mathbb{R}^p$, let $W_N = \{u^{fe}(\boldsymbol{\mu}^1), \cdots, u^{fe}(\boldsymbol{\mu}^N)\}$, where $u^{fe}(\boldsymbol{\mu}^i)$ is the numerical solution of problem (2.4) for the parameter values $\boldsymbol{\mu}^i$, for $1 \le i \le N$. Define the reduced basis space as $X_N^{rb} = \text{span}\{W_N\}$. The reduced basis approximation is then defined as: For a $\boldsymbol{\mu} \in \mathcal{D}$, find $u_N^{rb}(\boldsymbol{\mu}) \in X_N^{rb}$ such that

$$a(u_N^{rb}(\boldsymbol{\mu}),v;\boldsymbol{\mu}) = f(v;\boldsymbol{\mu}), \quad \forall v \in X_N^{rb}, \tag{2.6}$$

with the corresponding value of the output functional

$$s_N^{rb}(\boldsymbol{\mu}) = \ell(u_N^{rb}(\boldsymbol{\mu});\boldsymbol{\mu}). \tag{2.7}$$

Due to the affine assumption (2.3), an efficient offline-online strategy is immediately possible. For an RB space X_N^{rb} with N bases we precompute the matrices and vectors related to forms a_i, f_i, and ℓ_i, for $i = 1, \cdots, N$ during the offline step.

During the online step, the matrices and vectors in the RB formulation (2.6) are constructed, and the resulting small RB problem is solved to evaluate the output functional (2.7). The amount of work of the online step is independent of the degrees of freedom of X^{fe}, and depends only on the size of the reduced basis N and Q_a, Q_f, Q_ℓ. The operation count for an online evaluation of the output functional is

$$O(Q_a N^2) + O(N^3) + O(Q_f N) + O(Q_\ell N). \tag{2.8}$$

Thus, the RBM provides a path towards a cheap evaluation of the output functional $s(\boldsymbol{\mu})$ for an input $\boldsymbol{\mu}$ with the accuracy determined by the quality of the reduced basis.

To build the parameter set S_N, the corresponding basis set W_N and the reduced basis space X_N^{rb}, a greedy algorithm is used with the greedy approach based on an a posteriori error estimator. Let us define the error function $e(\boldsymbol{\mu}) = u_N^{rb}(\boldsymbol{\mu}) - u^{fe}(\boldsymbol{\mu}) \in X^{fe}$ as the difference between the reduced basis (RB) solution $u_N^{rb}(\boldsymbol{\mu})$ and the finite element solution $u^{fe}(\boldsymbol{\mu})$. The residual $r(v; \boldsymbol{\mu}) \in (X^{fe})'$ is defined as

$$r(v; \boldsymbol{\mu}) := f(v; \boldsymbol{\mu}) - a(u_N^{rb}, v; \boldsymbol{\mu}), \quad \forall v \in X^{fe}, \tag{2.9}$$

and its norm as

$$\|r(\cdot; \boldsymbol{\mu})\|_{(X^{fe})'} := \sup_{v \in X^{fe}} \frac{r(v; \boldsymbol{\mu})}{\|v\|_{X^{fe}}}. \tag{2.10}$$

We then define the estimator for the output as

$$\eta(\boldsymbol{\mu}, W_N) := \frac{\|r(\cdot; \boldsymbol{\mu})\|_{(X^{fe})'}^2}{\beta^{fe}(\boldsymbol{\mu})}. \tag{2.11}$$

Other types of error estimators can also be used, see e.g., [15]. A similar offline-online procedure can be employed for the computation of the error estimator.

For a train set $\Xi_{train} \subset \mathcal{D}$, consisting of a fine discretization of $\mathcal{D}$ of finite cardinality, let us suppose we already have S_N and the corresponding W_N and X_N^{rb}. We choose

$$\boldsymbol{\mu}^{N+1} := \operatorname{argmax}_{\boldsymbol{\mu} \in \Xi_{train}} \eta(\boldsymbol{\mu}; W_N), \tag{2.12}$$

to be the next sample point and let $S_{N+1} := S_N \cup \{\boldsymbol{\mu}^{N+1}\}$. This allows building the corresponding spaces W_{N+1} and X_{N+1}^{rb}. This procedure is repeated until N is large enough that $\max_{\boldsymbol{\mu} \in \Xi_{train}} \eta(\boldsymbol{\mu}; W_N)$ is less than a prescribed tolerance. For problems with high dimensional parameter spaces, one can consider the newly developed Adaptively Enriching Greedy Algorithm (AEGA) [9] which adaptively removes and enriches samples in the train set to control the computational cost at each searching step (2.12).

The AEGA addresses the problem of the richness of the train set for the problems with high dimensional parameter spaces, ensuring that finding each new reduced basis is under control even as the number of parameter is very large. However, for a problem with many parameters, one usually finds that many bases are required to ensure that the estimated error is less than a small tolerance. Furthermore, for non-affine problems, Q_a, Q_f, and Q_ℓ may be sizable, directly impacting the online cost of the evaluation of the output of interest.

3 The ANOVA expansion

In this section, we outline the Lebeque ANOVA expansion, and sparse Smolyak quadratures used for the computation of the ANOVA expansion. The presentation here generally follows [6,5].

Without loss of generality, we assume the parametric domain to be $\mathcal{D} = [0,1]^p$, and the output of interests $s(\boldsymbol{\mu}) \in L^2(\mathcal{D})$. Let $P = \{1, \cdots, p\}$ and let $t \subset P$ be any subset of the coordinate indices and t be its cardinality. The $|t|$-vector that contains the components of the vector $\boldsymbol{\mu} \in [0,1]^{|t|}$ indexed by t is denoted by $\boldsymbol{\mu}_t$. Let $A^{|t|}$ denote the $|t|$-dimensional unit hypercube which is the projection of the p-dimensional unit hypercube A^p onto the hypercube indexed by t. Then $s(\boldsymbol{\mu})$ can be expressed exactly as an ANOVA expansion as [18]:

$$s(\boldsymbol{\mu}) = s_0 + \sum_{t \subset P} s_t(\boldsymbol{\mu}_t), \tag{3.13}$$

where the term $s_t(\boldsymbol{\mu}_t)$ is defined recursively by

$$s_t(\boldsymbol{\mu}_t) = \int_{A^{p-|t|}} s(\boldsymbol{\mu}) d\boldsymbol{\mu}_{P \setminus t} - \sum_{w \subset t} u_w(\boldsymbol{\mu}_w) - s_0, \tag{3.14}$$

with

$$s_0 = \int_{A^p} s(\boldsymbol{\mu}) d\boldsymbol{\mu} \quad \text{and} \quad \int_{A^0} s(\boldsymbol{\mu}) d\boldsymbol{\mu}^0 = s(\boldsymbol{\mu}). \tag{3.15}$$

Let the variances $V_t(s)$ and the total variance $V(s)$ be

$$V_t(s) = \int_{A^p} (s_t(\boldsymbol{\mu}_t))^2 d\boldsymbol{\mu} \quad \text{and} \quad V(s) = \sum_{|t|>0} V_t(s) = \int_{A^p} (s(\boldsymbol{\mu}))^2 d\boldsymbol{\mu} - s_0^2. \tag{3.16}$$

The value $V_t(s)$ can be viewed as a direct measure of the variability of s restricted to a given set t. We then define the sensitivity measure

$$S(t) = \frac{V_t}{V}, \tag{3.17}$$

where V_t and V are defined in (3.16).

Since we only need a rough sense of the sensitivity, we can measure the sensitivity only by its first level of expansion. Let

$$s_{lin}(\boldsymbol{\mu}) = s_0 + \sum_{i=1}^{p} s_i(\boldsymbol{\mu}_i), \tag{3.18}$$

with $s_i(\boldsymbol{\mu}_i)$ defined in (3.14). We then define $V_i(s)$ as in (3.16): Define

$$V_{coarse} = \sum_{i=1}^{p} V_i \quad \text{and} \quad S_{coarse}(i) = \frac{V_i}{V_{coarse}}, \forall 1 \le i \le p. \tag{3.19}$$

Then S_{coarse} is a very coarse estimation of the sensitivity of the parameters.

3.1 Sparse Smolyak grids quadrature

To effectively compute the high-dimensional integrals (3.14) and (3.15) we utilize sparse grid methods based on the Smolyak construction [16]. This allows for the construction of sparse multivariate quadrature formulas based on sparse tensor products of one dimensional quadrature formulas.

Consider the numerical integration of a function $s(\boldsymbol{\mu})$ over a p-dimensional unit hypercube $A^p = [0, 1]^p$.

$$I[s] := \int_{A^p} s(\boldsymbol{\mu}) d\boldsymbol{\mu}. \tag{3.20}$$

A one-dimensional quadrature formula for a univariate function s is chosen as

$$Q_l^1 s = \sum_{i=1}^{n_l^1} \omega_i s(\mu_i^1), \tag{3.21}$$

Define

$$\Delta_i^1 s = (Q_i^1 - Q_{i-1}^1) s \quad \text{for} \quad i \in N_+,$$

with $Q_0^1 s = 0$. The Smolyak algorithm for the p-dimensional quadrature formula is given by

$$Q_l^p s = \sum_{|\mathbf{k}|_1 \leq l+p-1} \left(\Delta_{k_1}^1 \otimes \cdots \otimes \Delta_{k_p}^1 \right) s \quad \text{for} \quad l \in N \text{ and } \mathbf{k} = (k_1, \cdots, k_p) \in N^p. \tag{3.22}$$

An alternative formula to (3.22) is

$$Q_l^p s = \sum_{1 \leq |\mathbf{k}| \leq l+p-1} (-1)^{l+p-|\mathbf{k}|_1-1} \binom{p-1}{|\mathbf{k}|_1 - l} \left(Q_{k_1}^1 \otimes \cdots \otimes Q_{k_p}^1 \right) s. \tag{3.23}$$

If the one-dimensional quadrature nodes are nested, the sparse quadrature formulas inherit this property, yielding reduced computational cost. We use the Smolyak construction based on the Gauss-Patterson rule which is indeed nested. As addressed recently by [7, 11], this is the most efficient approach.

Remark 31 *Other numerical quadrature for (3.14) and (3.15) can also be used, such as Sobol' approach [17], Stroud formula [19], or other sparse grid based quadratures [7].*

4 The three-step RB-ANOVA-RB Method

Let us now combine the different elements discussed so far and introduce the three-step RB-ANOVA-RB method to enable the efficient and accurate evaluation of the output functional (1.1) for (1.2) in the case of a high-dimensional function, enabled by a parametric compression through the ANOVA expansion.

In the first stage of this, we build a reduced model with a high tolerance in the greedy approach for the full parametric problem. This yields a cheap but not very accurate approach for evaluating the output functional $s(\boldsymbol{\mu})$ for any intput $\boldsymbol{\mu} \in \mathcal{D}$.

In the second stage, we shall use the ANOVA expansion of the output functional to identify important parameters through the sensitivity analysis. To motivate this, let $\boldsymbol{\mu}^* = (\mu_1, \cdots, \mu_p)$ be a parameter vector in A^p with some parameters μ_i,

$i \in Q \subset \{1, \cdots, p\}$ frozen. Let $s^{rb}(\boldsymbol{\mu}^*)$ be a numerical approximation of s with a partially fixed parameter $\boldsymbol{\mu}^*$. A simple triangle inequality shows that

$$|s(\boldsymbol{\mu}) - s^{rb}(\boldsymbol{\mu}^*)| \leq |s(\boldsymbol{\mu}) - s(\boldsymbol{\mu}^*)| + |s(\boldsymbol{\mu}^*) - s^{rb}(\boldsymbol{\mu}^*)|. \qquad (4.24)$$

If we can guarantee the first term of the righthand side of this inequality to under control, we only need to approximate the reduced parameter problem.

The goal of the second stage of the RB-ANOVA-RB algorithm is to identify these (few) important parameters, and freeze the parameters of less importance. For this we apply the approximate ANOVA expansion based on sparse grid quadrature of the coarse reduced model computed during the first stage of the approach.

With the second step, we now have a rough estimate of the first term on the righthand side of (4.24). The last stage is to build a new and accurate reduced model for the reduced parameterized model.

As mentioned, there are several advantages associated with the reduced model over the smaller parameter space. First, since a smaller number of parameters makes the behavior of the solutions of the problem simpler, we generally need less basis elements to reduce the error to below a certain tolerance. Secondly, a smaller number of parameters makes the affine decomposition (see (2.3)) of a, f, and s shorter, thus reducing both the offline and the online computational costs. This is especially true for the non-affine problems, where the numbers Q_a, Q_f, and Q_ℓ often are large if there is a large number of parameters.

5 Numerical Experiments for RB-ANOVA-RB method

In the following we demonstrate the application of the RB-ANOVA-RB approach for two different problem types.

Thermal block problem

In the first experiment, we consider a thermal block problem [12], [15]. Let $\Omega = (0,1)^2$, and assume it is decomposed into 16 subdomains: $R_k = (\frac{i-1}{4}, \frac{i}{4}) \times (\frac{j-1}{4}, \frac{j}{4})$, for $i = 1, 2, 3, 4$, $j = 1, 2, 3, 4$, and $k = 4(i-1) + j$. The problem is

$$\begin{cases} -\nabla \cdot (\alpha \nabla u) = 0 \text{ in } \Omega, \\ \qquad u = 0 \text{ on } \Gamma_{top} = \{x \in (0,1), y = 1\}, \\ \alpha \nabla u \cdot \mathbf{n} = 0 \text{ on } \Gamma_{side} = \{x = 0 \text{ and } x = 1, y \in (0,1)\}, \\ \alpha \nabla u \cdot \mathbf{n} = 1 \text{ on } \Gamma_{base} = \{x \in (0,1), y = 0\}, \end{cases} \qquad (5.25)$$

where the diffusion constant α is assumed to be

$$\alpha = \begin{cases} \alpha_k = 100^{2\mu_k - 1}, \ x \in R_k, \ k = 1, 5, 9, 13, \\ \alpha_k = 1.1^{2\mu_k - 1}, \ \ x \in R_k, \ k = 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16. \end{cases}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_{16}) \in [0,1]^{16}$. Define $H_D^1 = \{v \in H^1 : v|_{\Gamma_{top}} = 0\}$. The goal is to evaluate the output functional

$$s(\boldsymbol{\mu}) = \int_{\Gamma_{base}} u(\boldsymbol{\mu}) ds, \qquad (5.26)$$

where the function $u(\boldsymbol{\mu}) \in H_D^1(\Omega)$ is the solution of a variational problem for the given $\boldsymbol{\mu} \in \mathcal{D}$,

$$a(u, v; \boldsymbol{\mu}) = f(v) \quad \forall v \in H_D^1(\Omega). \tag{5.27}$$

Here, the bilinear and linear forms are

$$a(u, v; \boldsymbol{\mu}) = \sum_{k=1}^{16} \alpha_k \int_{R_k} \nabla u \cdot \nabla v dx \quad \text{and} \quad f(v) = \int_{\Gamma_{base}} ds. \tag{5.28}$$

Let $\mathcal{T}$ be a uniform mesh on Ω with 80'401 of nodes (degrees of freedom), and $P_1(K)$ be the space of linear polynomials on an element $K \in \mathcal{T}$. Define our finite element approximation space

$$X^{fe} = \{v \in X : v|_K \in P_1(K), \quad \forall K \in \mathcal{T}\}.$$

For a given $\boldsymbol{\mu}$, the finite element problem seeks $u^{fe}(\boldsymbol{\mu}) \in X^{fe}$, such that

$$a(u^{fe}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v) \quad v \in X^{fe}. \tag{5.29}$$

We use the following functional based relative error estimator,

$$\eta(\boldsymbol{\mu}; W_N) = \frac{\|r(\cdot; \boldsymbol{\mu})\|_{X'}^2}{\beta_h(\boldsymbol{\mu})}, \tag{5.30}$$

with the $\|v\|_X = \sqrt{\int_\Omega |\nabla v|^2 dx}$, then $\beta_h(\boldsymbol{\mu}) = \min_k \alpha_k$.

As the first stage of the RB-ANOVA-RB, a reduced basis algorithm is computed for the full 16-parameter problem with the adaptively enriching greedy algorithm [9] with the size of the train set to be 200, the tolerance to be 1 and a 20.000-point safety check. We find that only 33 basis elements are needed to reduced the error to less than 1.

We continue and compute the AVOVA expansion based on sparse Smolyak grid quadratures as described in Section 3.1. The one-dimensional quadrature formula is based on a 63 point Gauss-Patterson rule in $[0, 1]$. An ANOVA expansion shows that $D = 3.81$, and $S_t \approx 0.99$, with $t = \{1, 5, 9, 13\}$. Thus, μ_1, μ_5, μ_9, and μ_{13} are the 4 most important parameters and their variation reflects 99% of the variation in the output of interest. Based on the setup of the problem, this is entirely expected.

For the third stage of the RB-ANOVA-RB algorithm, the reduced problem only has 4 parameters with

$$a(u, v; \boldsymbol{\mu}) = \sum_{k \in \{1,5,9,13\}} \alpha_k \int_{R_k} uv ds + a_{rest}(u, v), \tag{5.31}$$

where

$$a_{rest}(u, v) = \sum_{k \in \{2,3,4,6,7,8,11,12,14,15,16\}} \alpha_k(\mu_i^*) \int_{R_k} \nabla u \nabla v dx.$$

We build reduced bases for this 4-parameter problem with different tolerance. To measure the error, we generate 100 random parametric vectors in the full 16-parameter space. Output functionals for these 100 vectors are computed by finite elements, denoted by $\bar{s}_i^{fe}$, $i = 1, \cdots, 100$. For different tolerances, the reduced model are generated, and the output functionals are computed for these 100 vectors by the reduced form (5.39), denoted by $\bar{s}^{rb}$. $i = 1, \cdots, 100$.

Let us denote the relative error

$$e_i^{rel} = \frac{|\bar{s}_i^{fe} - \bar{s}_i^{rb}|}{|\bar{s}_i^{fe}|}, \quad i = 1, \dots, 100.$$

We shall compute the relative error in two norms.

$$e_{max} = \max_{i \in \{1, \cdots, 100\}} e_i^{rel} \quad \text{and} \quad e_{ave} = \frac{\sum_{i=1}^{100} e_i^{rel}}{100}. \tag{5.32}$$

The comparison of different tolerances are presented in Table 1. It is clear from this that after having a sufficiently well resolved reduced model, the first term in the righthand side of (4.24) is dominant. We need only around 20 elements to reduce the relative error below 4%, and the affine length of a is only 5.

In practice, since it is relatively cheap to acquire a smaller tolerance for the reduced parameter problem, we can over-compute a little to ensure the quality of the reduced bases.

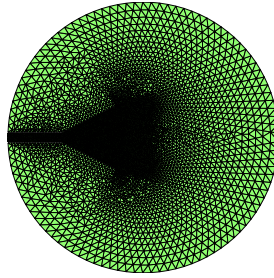**Table 1** Results of RB for the thermal block problem with different tolerances

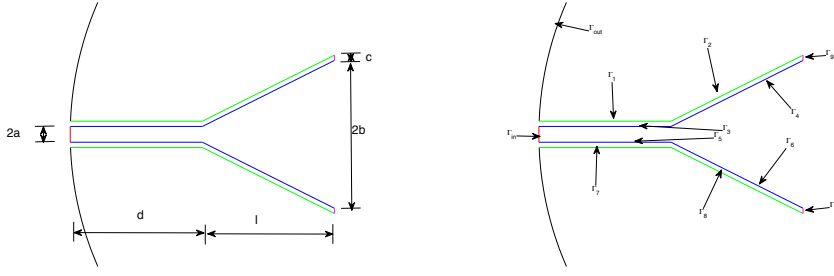| tol	Number of RB	e_{max}	e_{ave}
100	16	5.091×10^{-2}	7.732×10^{-3}
10	21	3.912×10^{-2}	7.177×10^{-3}
1	24	3.900×10^{-2}	7.192×10^{-3}
10^{-1}	30	3.893×10^{-2}	7.190×10^{-3}
10^{-2}	32	3.892×10^{-2}	7.190×10^{-3}

Acoustic horn problem

As a second example, consider an acoustic horn problem with 8 parameters [20].
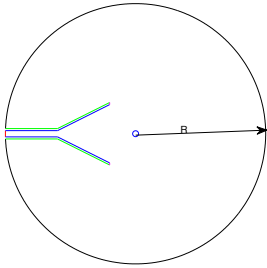
Fig. 1 The domain of the horn problem.

Fig. 2 The finite element mesh $\mathcal{T}$ of the horn problem.

Fig. 3 Geometry description of the domain.   **Fig. 4** Boundaries of the domain.

The acoustic field is described by time-harmonic Helmholtz equation

$$
\begin{cases}
\Delta u + 4u = 0 & \text{in } \Omega, \\
(2i + \dfrac{1}{25})u + \dfrac{\partial u}{\partial n} = 0 & \text{on } \Gamma_{out}, \\
2iu + \dfrac{\partial u}{\partial n} = 4i & \text{on } \Gamma_{in}, \\
i\mu_j u + \dfrac{\partial u}{\partial n} = 0 & \text{on } \Gamma_j, j = 1, \cdots, 8, \\
\dfrac{\partial u}{\partial n} = 0 & \text{on other boundaries.}
\end{cases}
\tag{5.33}
$$

with parameters $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_8) \in [0,1]^8$. The geometry of the problem is depicted in Figure 1 with terminology identified in Figures 3 and 4. We set $R = 12.5$, $a = 0.5$, $b = 3$, $c = 0.1$, $d = 5$, and $l = 5$.

The variational formulation of the problem is to find $u \in H^1(\Omega)$ such that

$$
a(u, v; \boldsymbol{\mu}) = 4i \int_{\Gamma_{in}} v\,ds, \quad \forall v \in H^1(\Omega).
\tag{5.34}
$$

The affine representation of $a(u, v; \boldsymbol{\mu})$ is

$$
a(u, v; \boldsymbol{\mu}) = \sum_{i=1}^{8} \mu_i a_i(u, v) + a_9(u, v).
\tag{5.35}
$$

where

$$
a_i(u, v) = \int_{\Gamma_i} uv\,ds, \quad \text{for} \quad i = 1, \cdots, 8,
\tag{5.36}
$$

and

$$
a_9(u, v) = (\nabla u, \nabla v)_\Omega - 4(u, v)_\Omega + (2i + \frac{1}{25})(u, v)_{\Gamma_{out}} + 2i(u, v)_{\Gamma_{in}}.
\tag{5.37}
$$

We choose the output of interest as

$$
s(\boldsymbol{\mu}) = \ell(u) = \text{real}(\int_{\Gamma_{in}} u\,ds).
\tag{5.38}
$$

Let the finite element space be $X^{fe} = \{v \in H^{\cdot}(\Omega) : v|_K \in P_1(K), \forall K \in \mathcal{T}\}$. The mesh $\mathcal{T}$ is illustrated in Figure 2 with the degrees of freedom of the finite element grid being 14.438. Let $\mathbf{r}_N$ be the vector representation of the residual $f(v) - a(u_N^{rb}, v; \boldsymbol{\mu})$, $\forall v \in X^{fe}$. We use a simple residual error estimator in the reduced basis, i.e.,

$$\eta_N = \|\mathbf{r}_N\|_{\ell^2}$$

with $\| \cdot \|_{\ell^2}$ be the standard Euclidean ℓ^2 norm.

A reduced model is computed for the full 8-parameter problem with the adaptively enriching greedy algorithm [9] using the size of the train set to be 500, the tolerance to be 0.001 and a 5.000-point safety check. The analysis finds that only 31 reduced basis elements are required.

This is utilized to compute an AVOVA expansion based on sparse Smolyak grid quadratures as described in Section 3.1. The one-dimensional quadrature formula is based on a 63-point Gauss-Patterson in $[0, 1]$. The ANOVA expansion shows that $S_3 = 0.4321$, $S_5 = 0.4314$, and $S_{35} = 0.1256$, and $D = 3.08 \times 10^{-4}$. Thus $S_3 + S_5 + S_{35} = 0.9891$. The numbers are clearly not very accurate, since in theory, S_3 should equal S_5 due to symmetry. However, this illustrates that the μ_3 and μ_5 are the most important parameters.

Finally, we perform the third step of the RB-ANOVA-RB algorithm. The new problem now has only two parameters with

$$a(u, v; \boldsymbol{\mu}) = \mu_3 \int_{\Gamma_3} uv ds + \mu_5 \int_{\Gamma_5} uv ds + a_{rest}(u, v), \tag{5.39}$$

where

$$a_{rest}(u, v) = \sum_{i \in \{1,2,4,,6,7,8\}} \mu_i^* a_i(u, v) + a_9(u, v)$$

with a_i defined in (5.36) and (5.37).

We build a reduced model for the 2-parameter problem with different tolerance. To quantify the error, we generate 100 random parameter vectors in the full 8-parameter space. Output functionals for these 100 vectors are computed by finite elements, denoted by $\bar{s}_i^{fe}$, $i = 1, \cdots, 100$. For different tolerances, reduced bases are generated, and the output functionals are computed for these 100 vectors by the reduced form (5.39), denoted by $\bar{s}^{rb}$. $i = 1, \cdots, 100$. The error e_{max} and e_{ave} are defined same as those in (5.32).

The comparison for different tolerances are presented in Table 2. It is clear from the table that with a sufficiently rich reduced model, the first term in the righthand side of (4.24) is dominant. We need only around 10 basis elements to reduce the relative error to near 1%, and the affine length of a is only 3.

Table 2 Results of reduced model for the horn problem with different tolerances

tol	Number of RB	e_{max}	e_{ave}
10^{-2}	6	1.172×10^{-2}	2.404×10^{-3}
10^{-3}	11	1.214×10^{-2}	1.516×10^{-3}
10^{-4}	15	1.1213×10^{-2}	1.516×10^{-3}
10^{-5}	17	1.1213×10^{-2}	1.516×10^{-3}

6 ANOVA-based *hp* reduced basis method

In this section, we will introduce the ANOVA-based *hp* reduced basis method. The algorithm is a recessive algorithm like that for the empirical interpolation method [3]. Unlike the method in [3] in which each parameter domain is decomposed into 2^p subdomains, we use a coarse version of the ANOVA sensitivity analysis to pick only K ($\ll p$ - a relatively small number) important parameters and decompose the parameter domain into only 2^K subdomains.

In the following, we introduce some notations and describe the recursive offline procedure to do parameter domain decomposition (*h*-part) and build the reduced bases locally in those subdomains with less than a prescribed basis size (*p*-part). Then in the second part, we described the modules of the offline procedure in detail. Finally an online procedure is described to perform efficient evaluation of the solution and the functional of interest.

6.1 Offline Procedure

Suppose D is a closed n-tube domain in $\mathbb{R}^p$. The number $N_{\max}$ is the maximum size of the reduced bases allowed in each domain, i.e., this controls cost. The prescribed tolerance is denoted by *tol*. $S_{N,D} = \{\boldsymbol{\mu}^1, \cdots, \boldsymbol{\mu}^N\}$ is a collection of N different parameter vectors in D. The space $X_{N,D}^{rb}$ is a reduced basis space with N bases on $S_{N,D}$, that is, $X_{N,D}^{rb} = \mathrm{span}\{u^{fe}(\boldsymbol{\mu}^1), \cdots, u^{fe}(\boldsymbol{\mu}^N)\}$, where $u^{fe}(\boldsymbol{\mu}^i)$ is the finite element solution corresponding to the parameter vector $\boldsymbol{\mu}^i$. Let closed n-tubes D_i, $i = 1, \cdots, T$ be subsets of D. We define a domain decomposition (partition) of D as

$$\mathcal{T}(D) = \{D_i : D = \cup_{i=1}^T D_i \quad \text{with} \quad \mathrm{interior}(D_i) \cap \mathrm{interior}(D_j) = \emptyset, \text{ if } i \neq j\}. \tag{6.40}$$

The collection of selected RB parameter vectors on a domain decomposition $\mathcal{T}(D)$ containing T subdomains of the parametric domain D is then defined as

$$S(\mathcal{T}(D)) = \{S_{N_i, D_i}, i = 1, \cdots, T, \text{ with } N_i \leq N_{\max}\}.$$

The corresponding collection of RB spaces on a domain decomposition $\mathcal{T}(D)$ containing T subdomains of the parametric domain D is defined as

$$X^{rb}(\mathcal{T}(D)) = \{X_{N_i, D_i}^{rb}, i = 1, \cdots, T, \text{ with } N_i \leq N_{\max}\}.$$

The triplet $(\mathcal{T}(D), S(\mathcal{T}(D)), X^{rb}(\mathcal{T}(D)))$ will be the output of the recursive function `AhpRB`, i.e., the main function of the algorithm of an "ANOVA-Based *hp* Reduced Basis Method". The pseudo-code is given in Algorithm 1. We give a brief description here, detailed modules/functions of the algorithm will be presented in the subsequent subsection.

Given a parameter domain D, a prescribed tolerance *tol*, a number $N_{\max}$ to restrict the maximum size of reduced bases, an initial parameter set $S_{N_0, D}$ and the corresponding RB space $X_{N_0, D}^{rb}$. When $N_0 = 0$, both $S_{0,D}$ and $X_{0,D}^{rb}$ are empty sets. We perform the Adaptively Enriching Greedy Algorithm (`AEGA`) to generate the reduced basis. If the algorithm converges, i.e., the error estimators over a large set of "safety check" points are less then *tol*, with less than or equal to $N_{\max}$ basis

element, then we already have a good RB space with a limited number of bases. If `AEGA` does not converge with $N_{\max}$ bases, we need to split the parameter domain D into smaller subdomains. To achieve this, we first use an `ANOVA` expansion of a functional output s to determine the relative importance of the different components of the p-vector $\boldsymbol{\mu} = (\boldsymbol{\mu}_{(1)}, \cdots, \boldsymbol{\mu}_{(p)})$. Some $\boldsymbol{\mu}_{(i)}$s are more important, or in other words, leads to larger variations in the output. These components $\boldsymbol{\mu}_{(i)}$ will be split. We `MARK` the components of the p-vector with either Maximum or Bulk marking strategies, and make sure to only mark K components with $K \leq K_{\max} \leq p$. We then `REFINE` the parameter domain. The important K parameter components are split with the gravity center point being the pivotal point, and we get 2^K subdomains $D_k{}_{k=1}^{2^K}$. The $S_{N,D}$ are distributed into these subdomains to have an S_{N_k, D_k} for each subdomain D_k, and form corresponding X_{N_k, D_k}^{rb}. With S_{N_k, D_k} and X_{N_k, D_k}^{rb} as initial parameter selections and RB spaces, a new procedure `AhpRB` is performed for each subdomain D_k, $k = 1, \cdots, 2^K$ to generate $\mathcal{T}(D_k), S(\mathcal{T}(D_k)), X^{rb}(\mathcal{T}(D_k))$. We `COMBINE` these sets into $\mathcal{T}(D), S(\mathcal{T}(D)), X^{rb}(\mathcal{T}(D))$. Now, for each subdomain of D, we have an RB space with less than or equal to $N_{\max}$ bases.

1: **function** $(\mathcal{T}(D), S(\mathcal{T}(D)), X^{rb}(\mathcal{T}(D))) = $ `AhpRB` $(S_{N_0, D}, X_{N_0, D}^{rb}, D, N_{max}, tol, s)$

2: $(X_{N,D}^{rb}, S_{N,D}, conv) = $ `AEGA` $(S_{N_0, D}, X_{N_0, D}^{rb}, D, N_{max}, tol)$;
3: **if** conv **then**
4: $\quad \mathcal{T}(D) = \{D\}, S(\mathcal{T}(D)) = \{S_{N,D}\}, X^{rb}(\mathcal{T}(D)) = \{X_{N,D}^{rb}\}$;
5: **else**
6: $\quad$ **sens** $= $ `ANOVA` (s, D) ;
7: $\quad$ (**sensd**, K) $= $ `MARK` (**sens**);
8: $\quad (\{D_k\}_{k=1}^{2^K}, \{S_{N_k, D_k}\}_{k=1}^{2^K}, \{X_{N_k, D_k}^{rb}\}_{k=1}^{2^K}) = $ `REFINE` $(D, S_{N,D}, X_{N,D}^{rb}, $ **sensd**, $K)$;
9: $\quad$ **for** $k = 1$ to 2^K **do**
10: $\quad\quad (\mathcal{T}(D_k), S(\mathcal{T}(D_k)), Y^{rb}(\mathcal{T}(D_k))) = $ `AhpRB` $(X_{N_k, D_k}^{rb}, X_{N_k, D_k}^{rb}, D_k, N_{max}, tol, s)$;
11: $\quad$ **end for**
12: $\quad (\mathcal{T}(D), S(\mathcal{T}(D)), X^{rb}(\mathcal{T}(D))) = $
$\quad$ `COMBINE` $(\{\mathcal{T}(D_k)\}_{k=1}^{2^K}, \{S(\mathcal{T}(D_k))\}_{k=1}^{2^K}, \{X^{rb}(\mathcal{T}(D_k))\}_{k=1}^{2^K})$;
13: **end if**

Algorithm 1: Recursive definition of the main function `AhpRB` (ANOVA-hp-Reduced Basis)

6.2 Modules of ANOVA-based hp Reduced Basis Method

Module `AEGA`. The main idea of the AEGA is introduced in [9]. Some modifications are made here, but the general procedure is similar to that in [9].

To describe the procedure, we define some global constants. Let M be the size of train set, N_{sc} be the number of the points used to check the quality of a reduced basis space. The number C_{sa} is the constant of the saturation assumption and we can choose $C_{sa} = 1$ since we always perform a "safety check". If we start from $N_0 = 0$, that is, there is no inheritance from the old domain and reduced basis space, a randomly picked parameter vector will be used in the initialization of $S_{N,D}$ and $X_{N,D}^{rb}$. Otherwise, we inherit $S_{N,D}$ from the the bigger parameter domain and its corresponding $X_{N,D}$. A standard Adaptively Enriching Greedy

Algorithm is performed to add new parameter vectors and build the reduced basis. If the algorithm converges, i.e., for N_{sc} randomly generated parameter vectors, the error estimators is less than *tol* with less than or equal to $N_{\max}$ reduced bases, we set $conv = \textbf{true}$, otherwise we set $conv = \textbf{false}$. The algorithm is presented in Algorithm 2.

1: **function**$(X_{N,D}^{rb}, S_{N,D}, conv) = \texttt{AEGA}\,(S_{N_0,D}, X_{N_0,D}^{rb}, D, N_{max}, tol)$

2: Generate an initial train set Ξ_{train} with M parameter samples;

3: **if** N_0 is 0 **then**

4:    $N = 1$, choose an initial parameter value $\boldsymbol{\mu}^1 \in \Xi_{train}$ and set $S_{N,D} = \{\boldsymbol{\mu}^1\}$ and $X_{N,D}^{rb} = \text{span}\{u^{fe}(\boldsymbol{\mu}^1)\}$;

5: **else**

6:    $N = N_0$, $S_{N,D} = S_{N_0,D}$, $X_{N,D}^{rb} = X_{N_0,D}^{rb}$

7: **end if**

8: Use Adaptive Enriching Greedy Algorithm idea to select new parameter vectors to be added into $S_{N,D}$ and new bases to be added to $X_{N,D}^{rb}$.

9: **if** convergence with $N \leq N_{\max}$ bases **then**

10:    Set conv = **true**, and let $S_{N,D}$ and $X_{N,D}^{rb}$ be the resultant selection of parameter vectors and reduced basis space, respectively.

11: **else**

12:    Set conv = **false**, $S_{N,D} = S_{N_{\max},D}$, and $X_{N,D}^{rb} = X_{N_{\max},D}^{rb}$;

13: **end if**

Algorithm 2: An Adaptively Enriching Greedy Algorithm

Module `ANOVA`. Unlike the three-step RB-ANOVA-RB algorithm, where the ANOVA expansion is used to determine which parameters can be frozen, in the *hp* reduced basis algorithm, the ANOVA expansion is used to determine which parameter domain needs to be split, i.e., those parameters with large variances. The requirement of the accuracy of the ANOVA expansion and sensitivity is very low and we can use the coarse ANOVA sensitivity defined in (3.19).

Remark 1 The functional of interest s and the error estimator should be a pair. That is, the error estimator should be the error estimator based on the functional. If a standard norm based error estimator is used, we should use the norm of the solution as a functional.

The algorithm is presented in Algorithm 3.

1: **function sens** $= \texttt{ANOVA}\,(s, D)$

2: s is the functional of interest.

3: Map the parameter domain D to A^p;

4: Compute the sensitivity using formula defined in (3.19), let $\textbf{sens}_{(i)} = S_{coarse}(i)$.

Algorithm 3: function ANOVA

Module `MARK`. We describe a `MARK` function here. As in adaptive finite element methods, we have two strategies to mark the parameters, maximum marking and

Dörfler's bulk marking strategies, see [1]. First, let $K_{\max} \leq p$ be the maximum number of parameter components to be split. Note, if we split a parameter component in domain $[a, b]$ into two subdomains, K parameter components will lead to 2^K subdomains in the p-dimensional domain. So we have to keep the number $K_{\max}$ reasonable small. The algorithm is presented in Algorithm 4, where two strategies are implemented.

1: **function** $(\mathbf{sensd}, K) = \texttt{MARK}\,(\mathbf{sens})$
2: Sort $\mathbf{sens}$ in descending order, such that $\mathbf{sensd} = \{\mathbf{sens}_{(i_1)}, \cdots, \mathbf{sens}_{(i_p)}\}$, with $\mathbf{sens}_{(i_1)} \geq \mathbf{sens}_{(i_2)} \geq \cdots \geq \mathbf{sens}_{(i_p)}$
3: $K = 0, 0 < \theta < 1$
4: **if** Maximum Marking Strategy **then**
5:    **while** $(K < K_{\max})$ **and** $(\mathbf{sensd}_{(K)} > \theta\,\mathbf{sensd}_{(1)})$ **do**
6:       $K = K + 1$
7:    **end while**
8: **else if** Bulk Marking Strategy **then**
9:    $SumSens = 0$
10:    **while** $(K < K_{\max})$ **and** $(SumSens < \theta)$ **do**
11:       $K = K + 1$
12:       $SumSens = SumSens + \mathbf{sens}_{(i_K)}$
13:    **end while**
14: **end if**

Algorithm 4: function $\texttt{MARK}$ based on Maximum/Bulk Marking Strategy

Module $\texttt{REFINE}$**.** We then describe the $\texttt{REFINE}$ procedure. For a parametric subdomain $D = [a_{(1)}, b_{(1)}] \times \cdots \times [a_{(P)}, b_{(P)}]$, and a vector $\mathbf{m}$ with values 0 and 1. We will split those parameters with value 1. The algorithm is an adaptation of the gravity center splitting scheme described in [3]. We define an operation $op(k, j)$ as

$$op(k, j) = \begin{cases} \leq \text{ if } \mathrm{bin}(k-1)_j = 0 \\ \geq \text{ if } \mathrm{bin}(k-1)_j = 1 \end{cases} \tag{6.41}$$

where $\mathrm{bin}(k)$ is the binary representation of k as a vector in $\{0, 1\}^K$. A slight difference from the definition in [3] is the use $\geq$ instead of $>$ since we assume each subdomain is a closed p-tube.

The domain D is split into 2^K distinct subdomains D_k, $1 \leq k \leq 2^K$, with some integer $1 \leq K \leq P$. The subdomain D_k may be written in a tensor-product form $D_k = [a_{(1)}^k, b_{(1)}^k] \times \cdots \times [a_{(p)}^k, b_{(p)}^k]$,

We define the gravity center point of the point cloud $S = \{\boldsymbol{\mu}_i, i = 1, \cdots, N_{max}\}$ by

$$\mathbf{g} = \frac{1}{N_{\max}} \sum_{i=1}^{N_{\max}} \boldsymbol{\mu}_i. \tag{6.42}$$

The k'th subdomain D_k is defined by

$$D_k = \{\boldsymbol{\mu} \in D : \boldsymbol{\mu}_{(i_j)} op(k, j) \mathbf{g}_{(i_j)}, 1 \leq j \leq K\} \tag{6.43}$$

where $i_1, \cdots, i_p$ are the indices such that $\mathbf{sens}_{(i_1)} \geq \mathbf{sens}_{(i_2)} \geq \cdots \mathbf{sens}_{(i_p)}$. Notice that in reduced basis methods, the basis elements are often needed to be orthogonalized. When we distribute the bases into subdomains, we should distribute

---

1: **function**
$(\{D_k\}_{k=1}^{2^K}, \{S_{N_k,D_k}\}_{k=1}^{2^K}, \{X_{N_k,D_k}^{rb}\}_{k=1}^{2^K}) = \texttt{REFINE}\,(D, S_{N,D}, X_{N,D}^{rb}, \mathbf{sensd}, K)$

2: Compute GC point $\mathbf{g}$ of $S_{N,D}$ based on eq. (6.42)

3: Generate $D_k, k = 1, \cdots, 2^K$ from D based on (6.43)

4: **for** $k = 1$ to 2^K **do**

5:     Let $S_{N_k,D_k} = S_{N,D} \cap D_k$.

6:     Get X_{N_k,D_k}^{rb} from S_{N_k,D_k}.

7: **end for**

Algorithm 5: function REFINE

the original pre-orthogonalized solutions. Then, in each new subdomain, these solutions need to be orthogonalized. Another way to do it is that no bases from previous level will be distributed. This way is simpler to implement, but comes at additional cost.

Module COMBINE. The COMBINE function first combines the decompositions $\{\mathcal{T}(D_i)\}_{i=1}^{2^K}$ into a decomposition of $D, \mathcal{T}(D)$. Since all these subdomains are disjoint, the combination is straightforward.

Remark 2 Notice that all these subdomains generated by the recursive algorithm form a tree [10]. $\mathcal{T}(D)$ is then the collection of all leaf (terminal) nodes of the tree with D as the root.

We should also keep a tree structure of the domains. This will be used in the online step to find the subdomain.

### 6.3 Online algorithm of the ANOVA-based hp reduced basis methods

The online procedure is relatively simple. For a given new parameter vector $\boldsymbol{\mu}_{new}$, we first find the subdomain $D_i \in \mathcal{T}(D)$ of $\boldsymbol{\mu}_{new}$. Since we keep the tree structure of subdomains, the cost is relatively low, although it depends on the number of parameter components p and the depth of the destiny node. After finding the subdomain to which the new parameter vector belongs, we have its corresponding reduced bases, and can compute the reduced basis solution and functional of interest with controlled computational cost.

7 Numerical Experiments for ANOVA-based hp Method

Since it very hard to illustrate the domain decomposition in high-dimensional space, we first consider the thermal block equation (5.25) with two parameters to demonstrate the algorithm. Let $R_1 = (0,0.5)^2$, $R_2 = (0.5,1)^2$, and $R_3 = (1,1)^2 \backslash \overline{R_1 \cup R_2}$. The diffusion constant α is assumed to be

$$\alpha = \begin{cases} \alpha_k = 10^{2\mu_i - 1}, \ x \in R_k, \ k = 1, 2, \\ \alpha_k = 1, \qquad x \in R_k, \ k = 3. \end{cases}$$

The tolerance is set to be 10^{-5} and N_{max} is set to be 7. We set $K_{\max} = 1$, implying that only one component is split. During the final stage, subdomain D_{10} needs 6 bases to converge, and all others need 7 bases.
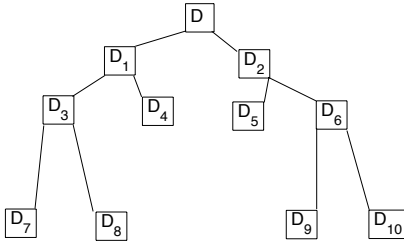
Fig. 5 Tree structure of the subdomains
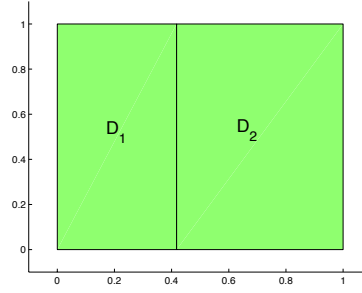
Fig. 6 First level of domain decomposition.
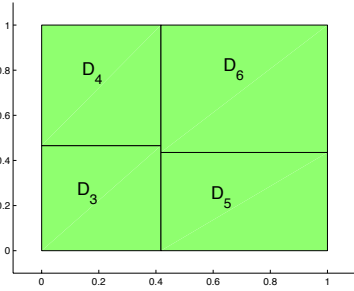
Fig. 7 Second level of domain decomposition.
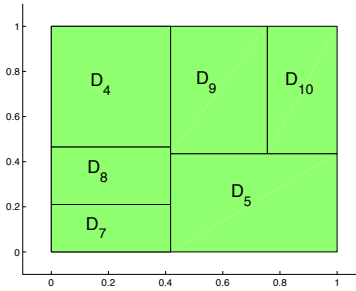
Fig. 8 Third level of domain decomposition.

We then we consider the thermal block equation (5.25) with 15 parameters. The domain $[0,1]^2$ is decomposed into 16 subdomains: $R_k = (\frac{i-1}{4}, \frac{i}{4}) \times (\frac{j-1}{4}, \frac{j}{4})$, for $i = 1, 2, 3, 4$, $j = 1, 2, 3, 4$, and $k = 4(i-1) + j$. The diffusion constant α is assumed to be

$$\alpha = \begin{cases} \alpha_k = 5^{2\mu_i - 1}, \ x \in R_k, \ k = 1, \cdots, 15 \\ \alpha_k = 1, \qquad x \in R_k, \ k = 16. \end{cases}$$

The tolerance is set to be 0.1 and N_{max} is set to be 35. We set $K_{\max} = 2$ and 3, and use maximum marking strategy with $\theta = 0.5$. The safety check step contains $10'000$ points. For $K_{\max} = 1$, 29 subdomains are generated with the deepest level is 5. For $K_{\max} = 2$, 58 subdomains are generated with the deepest level is 4. For $K_{\max} = 3$, 85 subdomains are generated with the deepest level is 3. It's obvious that when $K_{\max} = 1$, the fewest number of subdomains are generated, but the level of domain decomposition is also the deepest. In general, $K_{\max} = 1$ or 2 is suggested to balance the computational cost. $K_{\max} \geq 4$ is not recommended, at least not for this problem.

8 Concluding Remarks

In this paper, two new ANOVA-accelerated reduced basis methods are proposed to increase the efficiency for high-dimensional parametric partial differential equations.

Ahree step Reduced Basis-ANOVA-Reduced Basis method is proposed and validated for high-dimensional parametric partial differential equations. It provides an efficient and accurate way to reduce the parametric complexity and enable the rapid and accurate evaluation of the output of interests for a high-dimensional problem.

Extensions to the primal-dual approach of the reduced basis method is straightforward, and the primal-dual approach is very natural if the righthand side f and the output functional s depend on parameters explicitly. Extensions to non-affine problems is likewise straightforward, and the RB-ANOVA-RB algorithm is likely to result in additional savings in this case, in particular for the online phase.

If the ANOVA expansion of an output functional is the sole interest, the first two steps of the method are likely the most efficient way to get the ANOVA expansion to evaluate parametric sensitivity.

The ANOVA-based hp method enables us to construct a domain decomposition of the parameter domain such that in each subdomain, a fixed number of basis elements are needed. This method can be applied extended to the empirical interpolation method without any difficulties, hence allowing us to treat non-affine problems [8] at a fixed computational cost. Thus, combined with the adaptively enriching greedy algorithm developed for the empirical interpolation and reduced basis methods in [9], for the successive constraint method in [21], we now have a framework that allows for the use of reduced basis methods for problems with a high-dimensional parameter space.

References

1. W. Dörfler, *A convergent adaptive algorithm for Poissons equation*, SIAM J. Numer. Anal. 33, 1106–1124 (1996).
2. J. L. Eftang, A. T. Patera, and E. M. Ronquist, *An "hp" certified reduced basis method for parametrized elliptic partial differential equations*, SIAM J. Sci. Comput., 32(6):3170–3200, 2010.
3. J. L. Eftang and B. Stamm, *Parameter multi-domain hp empirical interpolation*, accepted in Int. J. Numer. Meth. Eng., NTNU preprint Numerics 03/11.
4. B. Fares, J. S. Hesthaven, Y. Maday, and B. Stamm, *The reduced basis method for the electric field integral equation*, J. Comput. Phys. 230(14), 2011, pp.5532–5555.
5. Z. Gao and J. S. Hesthaven, *On ANOVA expansions and strategies for choosing the anchor point*, Appl. Math. and Comp. 217(2010), 3274–3285.
6. Z. Gao and J. S. Hesthaven, *Efficient solution of ordinary differential equations with high-dimensional parametrized uncertainty*, Comm. Comput. Phys. 10(2011), 253-278.
7. T. Gerstner and M. Griebel, *Numerical integration using sparse grid*, Numer. Algor. 18(1998), 209–232.
8. M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, Math. Model. Numer. Anal., 41 (2007), pp. 575–605.

9. J. S. Hesthaven, B. Stamm, and S. Zhang, *New greedy algorithms for the empirical interpolation and reduced basis methods: With applications to high dimensional parameter spaces*, Scientific Computing Group Report, 2011-15, Division of Applied Mathematics, Brown University, Providence, RI.

10. D. Knuth, *The Art of Computer Programming: Fundamental Algorithms*, Vol. 1, 3rd Edition, Reading, Massachusetts: Addison-Wesley, 1997.

11. M. Liu, Z. Gao and J.S. Hesthaven, *Adaptive sparse grid algorithms with applications to electromagnetics scattering under uncertainty*, Appl. Numer. Math. 61(2011), 24-37.

12. A.T. Patera and G. Rozza, *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*, Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering.

13. A. Quarteroni, G. Rozza, and A. Manzoni, *Certified reduced basis approximation for parametrized partial differential equations and applications*, J. Math. Indust., 2011, 1:3.

14. *Reduced Basis at MIT.* [http://augustine.mit.edu/methodology.html]

15. G. Rozza, D.B.P. Huynh, and A.T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations - Application to transport and continuum mechanics*, Archives of Computational Methods in Engineering 15(3):229 - 275, 2008.

16. S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions.* Soviet Math. Dokl. 4, 240-243

17. I.M. Sobol, *Sensitivity estimates for nonlinear mathematical models*, Math. Model. Comput. Exp. 1(1993), 407-414.

18. I.M. Sobol, *Theorems and examples on high dimensional model representation*, Reliab. Engng. Syst. Safety, 79, 2003, 187-193.

19. A. Stroud, *Remarks on the disposition of points in numerical integration formulas*, Math. Comput., 11(1957), 257–261.

20. R. Udawalpola and M. Berggren, *Optimization of an acoustic horn with respect to efficiency and directivity*, Internat. J. Numer. Methods Engrg., 73(11):1571–1606, 2007.

21. S. Zhang, *Efficient greedy algorithms for successive constraints methods with high-dimensional parameters*, Brown Division of Applied Math Scientific Computing Tech Report 2011-23.