Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods

Jan S. Hesthaven * Benjamin Stamm † Shun Zhang ‡

September 9, 2011

Abstract. We propose two new and enhanced algorithms for greedy sampling of high-dimensional functions. While the techniques have a substantial degree of generality, we frame the discussion in the context of methods for empirical interpolation and the development of reduced basis techniques for high-dimensional parametrized functions. The first algorithm, based on a assumption of saturation of error in the greedy algorithm, is shown to result in a significant reduction of the workload over the standard greedy algorithm. In an improved approach, this is combined with an algorithm in which the train set for the greedy approach is adaptively sparsefied and enriched. A safety check step is added at the end of the algorithm to certify the quality of the basis set. Both these techniques are applicable to high-dimensional problems and we shall demonstrate their performance on a number of numerical examples.

1 Introduction

Approximation of a function is a generic problem in mathematical and numerical analysis, involving the choice of some suitable representation of the function and a statement about how this representation should approximate the function. A traditional approach is polynomial representation where the polynomials coefficients are chosen to ensure that the approximation is exact at certain specific points, recognized as the Lagrange form of the interpolating polynomial representation. Such representations, known as linear approximations, are independent of the function being approximated and have been used widely for centuries. However, as problems becomes complex and high-dimensional, the direct extension of such ideas quickly becomes prohibitive.

More recently, there has been an increasing interest in the development of nonlinear approximations in which case the approximation is constructed in a problem specific manner to reduce the overall computational complexity of constructing and evaluating the approximation to a given accuracy. In this setting, the key question becomes how to add an element to the existing approximation such that the new enriched approximation improves as much as possible, measured in some reasonable manner. This approach, known as a greedy approximation, seeks to maximize a given function, say the maximum error, and enrich the basis to eliminate this specific error, hence increasing the accuracy in an optimal manner when measured in the

*Division of Applied Mathematics, Box F, Brown University, 182 George St., Providence, RI 02912, Jan.Hesthaven@Brown.edu. This work is supported in part by OSD/AFOSR FA9550-09-1-0613.

†Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, stamm@math.berkeley.edu

‡Division of Applied Mathematics, Box F, Brown University, 182 George St., Providence, RI 02912, Shun_Zhang@Brown.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
09 SEP 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Brown University ,Division of Applied Mathematics,Box F,Providence,RI,02912**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Tech Report 2011-23

14. ABSTRACT

We propose two new and enhanced algorithms for greedy sampling of highdimensional functions. While the techniques have a substantial degree of generality, we frame the discussion in the context of methods for empirical interpolation and the development of reduced basis techniques for high-dimensional parametrized functions. The first algorithm, based on a assumption of saturation of error in the greedy algorithm, is shown to result in a significant reduction of the workload over the standard greedy algorithm. In an improved approach, this is combined with an algorithm in which the train set for the greedy approach is adaptively sparsefied and enriched. A safety check step is added at the end of the algorithm to certify the quality of the basis set. Both these techniques are applicable to high-dimensional problems and we shall demonstrate their performance on a number of numerical examples.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**23**	

maximum norm. Such a greedy approach has proven themselves to be particularly valuable for the approximation of high-dimensional problems where simple approaches are excluded due to the curse of dimension. For a detailed recent overview of such ideas in a general context, we refer to [15].

In this work we consider greedy algorithms and improvements of particular relevance to high-dimensional problems. While the ideas are of a general nature, we motivate and frame the discussion in the context of reduced basis methods (RBM) and empirical interpolation methods (EIM) in which the greedy approximation approach plays a key role. In the generic greedy approach, one typically needs a fine enough train set $\Xi_{train} \subset \mathcal{D}$ over which a functional has to be evaluated to select the next element of the approximation. When the number of parameters is high, the size of this train set quickly becomes considerable, rendering the computational cost substantial and perhaps even prohibitive. As consequence, since a fine enough train set is not realistic in practice, one is faced with the problem of ensuring the quality of the basis set under a non-rich enough train set. It is worth noting that when dealing with certain high dimensional problems, one may encounter the situation that the optimal basis set itself is of large size. This situation is, however, caused by the general complexity of the problems and we shall not discuss this further. Strategies for such cases are discussed in see [6, 7].

In this paper, we propose two enhanced greedy algorithms related to the search/loop over the large train set. The first algorithm utilizes a saturation assumption, based on the assumption that the greedy algorithm converges, i.e., with enough bases, the error will decrease to zero. It is then reasonable to assume that the error (or the error estimator in the case of the reduced basis method) is likewise decreasing in some sense. With a simple and reasonable saturation assumption on the error or the error estimator, we demonstrate how to modify the greedy algorithm such only errors/error estimators are computed for those points in Ξ_{train} with a large enough predicted error. With this simple modification, the total workload of the standard greedy algorithm is significantly reduced.

The second algorithm is an adaptively enriching greedy algorithm. In this approach, the samples in the train set is adaptively removed and enriched, and a safety check step is added at the end of the algorithm to ensure the quality of the basis set. On each step of searching a new parameter for a basis, the size of the train set is maintained at a reasonable number. This algorithm can be applied to problems with high number of parameters with substantial savings.

What remains of this paper is organized as follows. In Section 2, we discuss the role greedy sampling plans in different computational methods, exemplified by empirical interpolation and reduced basis methods, to highlight shortcomings of a naive approach and motivate the need for improved methods. This sets the stage for Section 3 where we discuss the details of two enhanced greedy techniques. This is followed in Section 4 and 5 by a number of detailed numerical examples for the empirical interpolation methods and reduced basis techniques, respectively, to illustrate the advantages of using these new methods for problems with both low and high-dimensional parameter spaces. Section 6 contains a few concluding remarks.

2 On the need for improved greedy methods

In the following we give a brief background on two different computational techniques, both of which rely on greedy approximation techniques, to serve as motivation for the subsequent discussion of the new greedy techniques.

2.1 Reduced basis methods

Many applications related to computational optimization, control, and design require the ability to rapidly and accurately solve parameterized problems many times for different parameter values within a given parametric domain $\mathcal{D} \subset \mathbb{R}^p$. While there are many suitable methods for this, we shall focus here on the reduced basis method (RBM) [12, 14] which has proven itself to be an very accurate and efficient method for such scenarios.

For any $\boldsymbol{\mu} \in \mathcal{D}$, the goal is to evaluate an output functional $s(\boldsymbol{\mu}) = \ell(u(\boldsymbol{\mu}); \boldsymbol{\mu})$, where $u(\boldsymbol{\mu}) \in X$ is the solution of

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in X \tag{2.1}$$

for some parameter dependent bilinear and linear forms a and f and X is a suitable functional space.

Let X^{fe} be a finite element discretization subspace of X. Here, finite elements are used for simplicity, and other types of discretizations can likewise be considered. 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,

$$a(u^{fe}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in X^{fe}, \tag{2.2}$$

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

Both the variational problem (2.1) and the approximation problem (2.2) are assumed to be well-posed. The following inf-sup stabilities are satisfied for $\boldsymbol{\mu}$-dependent positive constants $\beta(\boldsymbol{\mu})$ and $\beta^{fe}(\boldsymbol{\mu})$ respectively:

$$\beta(\boldsymbol{\mu}) = \inf_{u \in X} \sup_{v \in X} \frac{a(u, v; \boldsymbol{\mu})}{\|u\|_X \|v\|_X} \quad \text{and} \quad \beta^{fe}(\boldsymbol{\mu}) = \inf_{u \in X^{fe}} \sup_{v \in X^{fe}} \frac{a(u, v; \boldsymbol{\mu})}{\|u\|_{X^{fe}} \|v\|_{X^{fe}}}, \tag{2.3}$$

where $\|\cdot\|_X$ and $\|\cdot\|_{X^{fe}}$ are norms of the spaces X and 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 (2.2) corresponding to the parameter values $\boldsymbol{\mu}^i$, for $1 \le i \le N$. Then, define the reduced basis space as $X_N^{rb} = \text{span}\{W_N\}$.

The reduced basis approximation is now defined as: For a $\boldsymbol{\mu} \in \mathcal{D}$, find $u_N^{rb}(\boldsymbol{\mu}) \in X_N^{rb}$ such that

$$a(u_N^{rb}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in X_N^{rb}, \tag{2.4}$$

with the corresponding value of the output functional

$$s_N^{rb}(\boldsymbol{\mu}) = \ell(u_N^{rb}(\boldsymbol{\mu}); \boldsymbol{\mu}). \tag{2.5}$$

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 highly accurate finite element solution $u^{fe}(\boldsymbol{\mu})$. The residual $r(v; \boldsymbol{\mu}) \in (X^{fe})'$ is defined as

$$r(v; \boldsymbol{\mu}) := f(v; \boldsymbol{\mu}) - a(u_N^{rb}, v; \boldsymbol{\mu}), \quad \forall v \in X^{fe}, \tag{2.6}$$

and its norm as

$$\|r(\cdot; \boldsymbol{\mu})\|_{(X^{fe})'} := \sup_{v \in X^{fe}} \frac{r(v; \boldsymbol{\mu})}{\|v\|_{X^{fe}}}. \tag{2.7}$$

We define the relative estimator for the output as

$$\eta(\boldsymbol{\mu}, W_N) := \frac{\|r(\cdot; \boldsymbol{\mu})\|_{(X^{fe})'} \|\ell^{fe}(\cdot; \boldsymbol{\mu})\|_{(X^{fe})'}}{\beta^{fe}(\boldsymbol{\mu}) |s_N^{rb}(\boldsymbol{\mu})|}. \tag{2.8}$$

Other types of error estimators can also be used, see e.g., [14].

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. For a train set $\Xi_{train} \subset \mathcal{D}$, which consists of a fine discretization of $\mathcal{D}$ of finite cardinality, we first pick a $\boldsymbol{\mu}^1 \in \Xi_{train}$, and compute the corresponding basis $u^{fe}(\boldsymbol{\mu}^1)$. Let $S_1 = \{\boldsymbol{\mu}^1\}$, $W_1 = \{u^{fe}(\boldsymbol{\mu}^1)\}$, and $X_1^{rb} = \text{span}\{u^{fe}(\boldsymbol{\mu}^1)\}$. Now, suppose that we already found N points in Ξ_{train} to form S_N, the corresponding W_N and X_N^{rb}, for some integer $N \geq 1$. Then, choose

$$\boldsymbol{\mu}^{N+1} := \text{argmax}_{\boldsymbol{\mu} \in \Xi_{train}} \eta(\boldsymbol{\mu}; W_N), \tag{2.9}$$

to fix the next sample point and let $S_{N+1} := S_N \cup \{\boldsymbol{\mu}^{N+1}\}$. We then build the corresponding spaces W_{N+1} and X_{N+1}^{rb}. The above 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 this approach to be successful, it is essential that the training set is sufficiently fine, i.e., for problems with many parameters, the size of the train set Ξ_{train} could become very large. Even with a rapid approach for evaluating $\eta(\boldsymbol{\mu}; W_N)$ for all $\boldsymbol{\mu} \in \Xi_{train}$ the cost of this quickly becomes a bottleneck in the construction of the reduced basis.

2.2 Empirical interpolation method

One of the main attractions of the reduced basis method discussed above becomes apparent if we assume that the parameter dependent problem (2.1) satisfies an affine assumption, that is,

$$a(u, v; \boldsymbol{\mu}) = \sum_{i=1}^{Q_a} \Theta_i^a(\boldsymbol{\mu}) a_i(u, v), \quad f(v; \boldsymbol{\mu}) = \sum_{i=1}^{Q_f} \Theta_i^f(\boldsymbol{\mu}) f_i(v), \quad \text{and} \quad \ell(v; \boldsymbol{\mu}) = \sum_{i=1}^{Q_\ell} \Theta_i^\ell(\boldsymbol{\mu}) \ell_i(v), \tag{2.10}$$

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. With this assumption, for a reduced basis space X_N^{rb} with N bases, we can applu the so-called offline/online strategy. In the offline step, one can precompute the matrices and vectors related to forms a_i, f_i, and ℓ_i, for $i = 1, \cdots, N$. The cost of this may be substantial but is done only once. In the online step, we now construct the matrices and vectors in the reduced basis formulation (2.4), solve the resulting reduced basis problem, and then evaluate the output functional (2.5). The amount of work of the online step is independent of the degrees of freedom of X^{fe}, and only depends on the size of reduced basis N and the affine constants Q_a, Q_f, and Q_ℓ. Hence, for a fixed $\boldsymbol{\mu} \in \mathcal{D}$, the computation $\eta(\boldsymbol{\mu}; W_N)$ includes the solving procedure of the reduced basis problem, the evaluation of the residual (and output functional) in the dual norm, and a possible linear programming problem to evaluate $\beta^{fe}(\boldsymbol{\mu})$, see [2]. As already anticipated, the amount of work does not depend on the size of X^{fe}, but only on N and is, hence, very fast.

However, when the parameter dependent problem does not satisfy the affine assumption (2.10), this key benefit is lost. To circumvent this, the empirical interpolation method (EIM) [2, 9, 8] has been developed to enable one to treat the non-affine operators and approximate them on the form (2.10) to maintain computational efficiency.

To explain the EIM, consider a parameter dependent function $\mathcal{F} : \Omega \times \mathcal{D} \to \mathbb{R}$ or $\mathcal{F} : \Omega \times \mathcal{D} \to \mathbb{C}$. The EIM is introduced in [2, 9, 11] and serves to provide parameter values $S_N = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N\}$ such that the interpolant

$$\mathcal{I}_N(\mathcal{F})(\mathbf{x}; \boldsymbol{\mu}) := \sum_{n=1}^{N} \beta_n(\boldsymbol{\mu}) q_n(\mathbf{x}) \tag{2.11}$$

is an accurate approximation to $\mathcal{F}(\mathbf{x};\boldsymbol{\mu})$ on $\Omega \times \mathcal{D}$.

The sample points S_N are chosen by the following greedy algorithm. Again, using a train set $\Xi_{train} \subset \mathcal{D}$ which consists of a fine discretization of $\mathcal{D}$ of finite cardinality, we first pick a $\boldsymbol{\mu}^1 \in \Xi_{train}$, compute $\mathbf{x}^1 = \arg\max_{\mathbf{x}\in\Omega} \mathcal{F}(\mathbf{x};\boldsymbol{\mu}^1)$ and the corresponding basis $q_1(\cdot) = \mathcal{F}(\cdot;\boldsymbol{\mu}^1)/\mathcal{F}(\mathbf{x}^1;\boldsymbol{\mu}^1)$. Then, let $S_1 = \{\boldsymbol{\mu}^1\}$ and $W_1 = \{q_1\}$.

Now, suppose that we already found N points in Ξ_{train} to form S_N and $W_N = \{q_1, \ldots, q_N\}$ such that $\text{span}\{q_1, \ldots, q_N\} = \text{span}\{\mathcal{F}(\cdot;\boldsymbol{\mu}^1), \ldots, \mathcal{F}(\cdot;\boldsymbol{\mu}^N)\}$, for some integer $N \geq 1$. We further assume that a set of N points $T_N = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ is given such that

$$q_j(\mathbf{x}^i) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i < j, \\ q_j(\mathbf{x}^i) & \text{otherwise,} \end{cases} \tag{2.12}$$

for all $i, j = 1, \ldots, N$. Then, we denote the lower triangular interpolation matrix $B_{ij}^N = q_j(\mathbf{x}^i)$, $i, j = 1, \ldots, N$, to define the coefficients $\{\beta^n(\boldsymbol{\mu})\}_{n=1}^N$, for a given $\boldsymbol{\mu} \in \mathcal{D}$, which are the solution of the linear system

$$\sum_{j=1}^{N} B_{ij}^N \beta_j(\mu) = \mathcal{F}(\mathbf{x}^i;\boldsymbol{\mu}), \qquad \forall i = 1, \ldots, N.$$

The approximation of level N of $\mathcal{F}(\cdot;\boldsymbol{\mu})$ is given by the interpolant defined by (2.11). We then set

$$\eta(\boldsymbol{\mu}; W_N) := \|\mathcal{F}(\cdot;\boldsymbol{\mu}) - \mathcal{I}_N(\mathcal{F})(\cdot;\boldsymbol{\mu})\|_{L^\infty(\Omega)}$$

and choose

$$\boldsymbol{\mu}^{N+1} := \text{argmax}_{\boldsymbol{\mu}\in\Xi_{train}} \eta(\boldsymbol{\mu}; W_N), \tag{2.13}$$

to fix the next sample point and let $S_{N+1} := S_N \cup \{\boldsymbol{\mu}^{N+1}\}$. The interpolation point $\mathbf{x}^{N+1}$ is defined by

$$\mathbf{x}^{N+1} = \text{argmax}_{\mathbf{x}\in\Omega}\big(\mathcal{F}(\cdot;\boldsymbol{\mu}) - \mathcal{I}_N(\mathcal{F})(\cdot;\boldsymbol{\mu})\big)$$

and the next basis function by

$$q_{N+1} = \frac{\mathcal{F}(\cdot;\boldsymbol{\mu}) - \mathcal{I}_N(\mathcal{F})(\cdot;\boldsymbol{\mu})}{\mathcal{F}(\mathbf{x}^{N+1};\boldsymbol{\mu}) - \mathcal{I}_N(\mathcal{F})(\mathbf{x}^{N+1};\boldsymbol{\mu})}.$$

By construction, we therefore satisfy the condition (2.12) since the interpolation is exact for all previous sample points in S_N. The algorithm is stopped once the error $\max_{\boldsymbol{\mu}\in\Xi_{train}} \eta(\boldsymbol{\mu}; W_N)$ is below some prescribed tolerance. As one can observe, the EIM also uses a greedy algorithm to choose the sample points with only slight differences to the case of the reduced basis method. Hence, in the case of a high dimensional parameter space, the computational cost of constructing the empirical interpolation can be substantial.

3 Improved greedy algorithms

Having realized the key role the greedy approach plays in the two methods discussed above, it is clear that even if the greedy approach is used only in the offline phase, it can result in a very considerable computational cost, in particular in the case of a high-dimensional parameter space. Let us in the following discuss two ideas aimed to help reduce the computational of the greedy approach in this case.

3.1 A typical greedy algorithm

To make the algorithm more general than discussed in the above, we make several assumptions. For each parameter $\boldsymbol{\mu}$ in the parameter domain $\mathcal{D} \subset \mathbb{R}^p$, a $\boldsymbol{\mu}$-dependent basis function $v(\boldsymbol{\mu})$ can be computed. Let

$$S_N = \{\boldsymbol{\mu}^1, \cdots, \boldsymbol{\mu}^N\}$$

be a collection of N parameters in $\mathcal{D}$ and

$$W_N = \{v(\boldsymbol{\mu}^1), \cdots, v(\boldsymbol{\mu}^N)\}$$

be the collection of N basis functions based on the parameter set S_N. For each parameter $\boldsymbol{\mu} \in \mathcal{D}$, suppose that we can compute an error estimator $\eta(\boldsymbol{\mu}; W_N)$ of the approximation based on W_N.

The following represent a typical greedy algorithm.

Input: A train set $\Xi_{train} \subset \mathcal{D}$, a tolerance $tol > 0$
Output: S_N and W_N

1: **Initialization**: Choose an initial parameter value $\boldsymbol{\mu}^1 \in \Xi_{train}$, set $S_1 = \{\boldsymbol{\mu}^1\}$, compute $v(\boldsymbol{\mu}^1)$, set $W_1 = \{v(\boldsymbol{\mu}^1)\}$, and $N = 1$;
2: **while** $\max_{\boldsymbol{\mu} \in \Xi} \eta(\mu; W_N) > tol$ **do**
3: For all $\boldsymbol{\mu} \in \Xi_{train}$, compute $\eta(\boldsymbol{\mu}; W_N)$;
4: Choose $\boldsymbol{\mu}^{N+1} = \text{argmax}_{\boldsymbol{\mu} \in \Xi_{train}} \eta(\boldsymbol{\mu}; W_N)$;
5: Set $S_{N+1} = S_N \cup \{\boldsymbol{\mu}^{N+1}\}$;
6: Compute $v(\boldsymbol{\mu}^{N+1})$, and set $W_{N+1} = W_N \cup \{v(\boldsymbol{\mu}^{N+1})\}$;
7: $N \leftarrow N + 1$;
8: **end while**

Algorithm 1: A Typical Greedy Algorithm

Note that S_N and W_N are hierarchical:

$$S_N \subset S_M, \quad W_N \subset W_M \quad \text{if} \quad 1 \le N \le M.$$

3.2 An improved greedy algorithm based on a saturation assumption

As mentioned before, on step 3 of the greedy algorithm 1, we have to compute $\eta(\boldsymbol{\mu}; W_N)$ for every $\boldsymbol{\mu} \in \Xi_{train}$. When the size of Ξ_{train} is large, the computational cost of this task is very high. Fortunately, in many cases, the following saturation assumption holds:

Definition 3.1. Saturation Assumption
Assume that $\eta(\boldsymbol{\mu}; W_N) > 0$ is an error estimator depending on a parameter $\boldsymbol{\mu}$ and a hierarchical basis space W_N. If the following property

$$\eta(\boldsymbol{\mu}; W_M) \le C_{sa}\, \eta(\boldsymbol{\mu}; W_N) \quad \textit{for some} \quad C_{sa} > 0 \quad \textit{for all} \quad 0 < N < M \tag{3.14}$$

holds, we say that the Saturation Assumption is satisfied.

Remark 3.2. *When $C_{sa} = 1$, it implies that $\eta(\boldsymbol{\mu}; W_N)$ is not increasing for a fixed $\boldsymbol{\mu}$ and increasing N. When $C_{sa} < 1$, this is a more aggressive assumption, ensuring that $\eta(\boldsymbol{\mu}; W_N)$ is strictly decreasing. This assumption with $C_{sa} < 1$ is very common in the adaptive finite element method community, see [1]. The assumption $C_{sa} > 1$ is a more relaxed assumption, allowing that $\eta(\boldsymbol{\mu}; W_N)$ might not be monotonically decreasing, but can oscillating. Since the*

underlying assumption of the greedy algorithm is that $\eta(\boldsymbol{\mu}, W_N)$ will converge to zero as N approaches infinity, we can safely assume that even if $\eta(\boldsymbol{\mu}, W_N)$ might exhibit intermittend non-monotone as N is increasing, overall it is decreasing.

Utilizing this assumption, we can design an improved greedy algorithm. First, for each parameter value $\boldsymbol{\mu} \in \Xi_{train}$, we create an error profile $\eta_{saved}(\boldsymbol{\mu})$. For instance, initially, we can set $\eta_{saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}; W_0) = \infty$. Now suppose that S_N and W_N are determined and that we aim to find the next sample point $\boldsymbol{\mu}^{N+1} = \text{argmax}_{\boldsymbol{\mu} \in \Xi_{train}} \eta(\boldsymbol{\mu}; W_N)$. When $\boldsymbol{\mu}$ runs through over the train set Ξ_{train}, we naturally keep updating a temporary maximum (over Ξ_{train}), until we have searched the whole set Ξ_{train}. In this loop, we may require that, for each $\boldsymbol{\mu}$, $\eta_{saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}; W_L)$ for some $L < N$. Now, since the Saturation Assumption $\eta(\boldsymbol{\mu}; W_N) \leq C_{sa}\, \eta(\boldsymbol{\mu}; W_L)$ for $L < N$ holds and if $C_{sa}\, \eta_{saved}(\boldsymbol{\mu})$ is less than the current temporary maximum, $\eta(\boldsymbol{\mu}, W_N)$ can not be greater than the current temporary maximum. Hence, we may skip the computation of $\eta(\boldsymbol{\mu}, W_N)$, and leave $\eta_{saved}(\boldsymbol{\mu})$ untouched. On the other hand, if $C_{sa}\, \eta_{saved}(\boldsymbol{\mu})$ is greater than the current temporary maximum, it is potentially the maximizer. Hence, we compute $\eta(\boldsymbol{\mu}, W_N)$, update $\eta_{saved}(\boldsymbol{\mu})$, and compare it with the current maximum to see if an update of the current temporary maximum is needed. We notice that if we proceed in this manner, then the requirement that for each $\boldsymbol{\mu}$, $\eta_{saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}; W_L)$ for some $L < N$ holds.

The algorithm 2 in pseudo-code of the Saturation Assumption based algorithm is given as:

Input: A train set $\Xi_{train} \subset \mathcal{D}$, C_{sa}, and a tolerance *tol*
Output: S_N and W_N

1: Choose an initial parameter value $\boldsymbol{\mu}^1 \in \Xi_{train}$, set $S_1 = \{\boldsymbol{\mu}^1\}$; compute $v(\boldsymbol{\mu}^1)$, set $W_1 = \{v(\boldsymbol{\mu}^1)\}$, and $N = 1$;
2: Set a vector η_{saved} with $\eta_{saved}(\boldsymbol{\mu}) = \infty$ for all $\boldsymbol{\mu} \in \Xi_{train}$;
3: **while** $\max_{\boldsymbol{\mu} \in \Xi_{train}} \eta_{saved}(\boldsymbol{\mu}) \geq tol$ **do**
4: $error_{tmpmax} = 0$;
5: **for all** $\boldsymbol{\mu} \in \Xi_{train}$ **do**
6: **if** $C_{sa}\eta_{saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
7: Compute $\eta(\boldsymbol{\mu}; W_N)$, and let $\eta_{saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}, W_N)$;
8: **if** $\eta_{saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
9: $error_{tmpmax} = \eta_{saved}(\boldsymbol{\mu})$, and let $\boldsymbol{\mu}_{max} = \boldsymbol{\mu}$;
10: **end if**
11: **end if**
12: **end for**
13: Choose $\boldsymbol{\mu}^{N+1} = \boldsymbol{\mu}_{max}$, set $S_{N+1} = S_N \cup \{\boldsymbol{\mu}^{N+1}\}$;
14: Compute $v(\boldsymbol{\mu}^{N+1})$, set $W_{N+1} = W_N \cup \{v(\boldsymbol{\mu}^{N+1})\}$;
15: $N \leftarrow N + 1$;
16: **end while**

Algorithm 2: A greedy algorithm based on a saturation assumption

Remark 3.3. *Initially, we set $\eta_{saved}(\boldsymbol{\mu}) = \infty$ for any $\boldsymbol{\mu} \in \Xi_{train}$ to make ensure every $\eta(\boldsymbol{\mu}, W_1)$ will be computed.*

Remark 3.4. *Due to round-off errors, the error estimator may stagnate even if we add more bases, or the greedy algorithm will select some point already in S_N. In this case, the greedy algorithm should be terminated.*

3.3 An adaptively enriching greedy algorithm

Even though the above saturation assumption based algorithm has the potential to reduce the overall computational cost, it may still be too costly for problems with a high number of parameters. Notice that, in the above algorithms, the assumption that the train set Ξ_{train} is a fine enough discrete subset of $\mathcal{D}$ is essential; otherwise, we might miss some phenomena that are not represented by Ξ_{train}. The consequence of this is that for the final sets of S_N and W_N, there may be some parameter $\tilde{\boldsymbol{\mu}}$ in $\mathcal{D}$ but not in Ξ_{train} such that $\eta(\tilde{\boldsymbol{\mu}}, W_N)$ is greater (or even much greater) than tol.

Thus, for high dimensional parameter problems, we will likely have to face the problem that the train set is not rich enough. To address this problem, we first build the set of basis W_N based on a train set with a relatively small number of points.

To ensure that the set of basis functions corresponding to this train set is good enough, we add a "safety check" step at the end, that is, we test the bases by a larger number of test parameters, to see if the resulting error estimators are less than the prescribed tolerance on this larger parameter set too. If for all test points, the estimated errors are less then the tolerance, we may conclude that the basis set is good enough. Otherwise, we add the first failed test point (whose error is larger than the tolerance) to S_N, and redo the "safety check" step until the set of basis W_N passes the "safety check".

For problems with a high number of of parameters, it is in practice hard to construct a rich enough train set. First, it is almost impossible to construct a tensor product based train set for $p > 10$. Even if we only put 3 points for each parameter, which is of course far from rich, a train set of $3^{11} = 177'147$ points is quite big. For a bigger p, the curse of dimensionality is inevitable.

Instead of starting from a tensor product train set, we consider a (quasi-)random train set. However, the random train set faces the same problem: it is far from "rich enough" for a high dimensional parameter problem. Fortunately, we can adaptively change the train set by removing useless points and enriching new points. Notice that, after we have determined a set of basis functions, the estimated errors corresponding to some points in the train set are already smaller than the prescribed tolerance. There is no value in retaining these points in S_N they should be removed from the train set. Indeed, we can add some new random points to the train set to make the size of the train set of constant cardinality.

Notice that for the unchanged part of the train set, we can still apply the saturation assumption based algorithm to save working load, and thus combine the two ideas.

The algorithm 3 is the pseudo-code of the adaptively enriching greedy algorithm.

Remark 3.5. *The algorithm can further be modified in the way that any new parameter point in Ξ_{train} is subject to some optimization procedure. For instance, one can apply a random walk with decreasing step size and only accept a new state if the error estimator is increased. Or, a more complex procedure such as the simulated annealing algorithm can be applied. Such a procedure will additionally (at some cost though) increase the quality of each added parameter point.*

4 Application to the empirical interpolation method

In the following we study how the previous ideas can be used to improve the greedy algorithm incorporated in the empirical interpolation method (EIM).

Input: M: the number of sample points in each round of searching,
N_{sc}: the number of sample points to pass the safety check,
C_{sa}, and a tolerance tol.
Output: S_N and W_N

Set $N_{\text{safe}} = \text{ceil}(N_{sc}/M)$;
Generate an initial train set Ξ_{train} with M parameter samples (randomly, or do your best);
Choose an initial parameter value $\boldsymbol{\mu}^1 \in \Xi_{train}$ and set $S_1 = \{\boldsymbol{\mu}^1\}$ and $N = 1$;
Set a vector η_{saved} with $\eta_{saved}(\boldsymbol{\mu}) = \infty$ for all $\boldsymbol{\mu} \in \Xi_{train}$;
Compute $v(\boldsymbol{\mu}^1)$, set $W_1 = \{v(\boldsymbol{\mu}^1)\}$, set safe $= 0$, $error_{tmpmax} = 2 * tol$;
while ($error_{tmpmax} \geq tol$ or safe $\leq N_{\text{safe}}$) **do**
  $error_{tmpmax} = 0$;
  for all $\boldsymbol{\mu} \in \Xi$ **do**
    if $C_{sa}\eta_{saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
      Compute $\eta(\boldsymbol{\mu}; W_N)$, and let $\eta_{saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}, W_N)$;
      if $\eta_{saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
        $error_{tmpmax} = \eta_{saved}(\boldsymbol{\mu})$, and let $\boldsymbol{\mu}_{max} = \boldsymbol{\mu}$;
      end if
      if $\eta_{saved}(\boldsymbol{\mu}) < tol$ **then**
        flag $\boldsymbol{\mu}$; `// all flagged parameters will be removed later`
      end if
    end if
  end for
  if $error_{tmpmax} > tol$ **then**
    Choose $\boldsymbol{\mu}^{N+1} = \boldsymbol{\mu}_{max}$, set $S_{N+1} = S_N \cup \boldsymbol{\mu}^{N+1}$;
    Compute $v(\mu^{N+1})$, set $W_{N+1} = W_N \cup \{v(\boldsymbol{\mu}^{N+1})\}$;
    Discard all flagged parameters from Ξ_{train} and their corresponding saved error estimation in η_{saved};
    Generate $M - \text{sizeof}(\Xi_{train})$ new samples, add them into Ξ_{train} such that $\text{sizeof}(\Xi_{train}) = M$; set η_{saved} of all new points to ∞;
    $N \leftarrow N + 1$;
    safe =0;
  else
    safe = safe +1;
    Discard Ξ_{train}, generate M new parameters to form Ξ_{train} and set η_{saved} to be ∞ for all $\boldsymbol{\mu} \in \Xi_{train}$;
  end if
end while

Algorithm 3: An Adaptively Enriching Greedy Algorithm

4.1 Saturation assumption

We first test the saturation assumption based algorithm for two test problems with low dimensional parameter spaces.

Test 4.1. Consider the complex-valued function

$$\mathcal{F}_1(x;k) = \frac{e^{ikx} - 1}{x}$$

where $x \in \Omega = (0, 2]$ and $k \in \mathcal{D} = [1, 25]$. The interval Ω is divided into a equidistant point set of cardinality 15'000 points to build Ω_h where the error $\|\mathcal{F}_1(\mathbf{x};\boldsymbol{\mu}) - \mathcal{I}_N(\mathcal{F}_1)(\mathbf{x};\boldsymbol{\mu})\|_{L^\infty(\Omega_h)}$ is computed. For the standard greedy algorithm, the train set Ξ_{train} consists of 5'000 equidistant points in $\mathcal{D}$.

Test 4.2. As a second and slightly more complicated example, consider the complex-valued function

$$\mathcal{F}_2(\mathbf{x};\boldsymbol{\mu}) = e^{ik\hat{\boldsymbol{k}}\cdot\mathbf{x}}$$

where the directional unit vector $\hat{\boldsymbol{k}}$ is given by

$$\hat{\boldsymbol{k}} = -(\sin\theta\,\cos\varphi, \sin\theta\,\sin\varphi, \cos\theta)^T$$

and $\boldsymbol{\mu} = (k, \theta, \varphi) \in \mathcal{D} = [1, 5]\times[0, \pi]\times[0, 2\pi]$. As domain Ω we take a unit sphere. For practical purpose, we take a polyhedral approximation to the sphere, as illustrated in Figure 1, and the discrete number of points Ω_h (where again the error $\|\mathcal{F}_2(\mathbf{x};\boldsymbol{\mu}) - \mathcal{I}_N(\mathcal{F}_2)(\mathbf{x};\boldsymbol{\mu})\|_{L^\infty(\Omega_h)}$ is computed) consists of the three Gauss points on each triangle. For the standard greedy algorithm, the train set Ξ_{train} consists of a rectangular grid of $50 \times 50 \times 50$ points. In computational

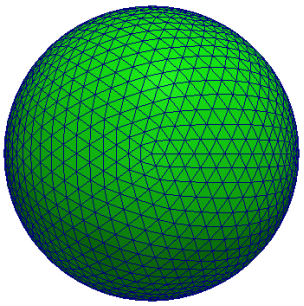

Figure 1: Discrete surface for the unit sphere.

electromagnetics, this function is widely used since $\boldsymbol{p}\mathcal{F}_2(\mathbf{x};\boldsymbol{\mu})$ represents a plane wave with wave direction $\hat{\boldsymbol{k}}$ and polarization $\boldsymbol{p} \perp \hat{\boldsymbol{k}}$ impinging onto the sphere. See [8].

In Figure 2 we show the evolution of the average and maximum value of

$$C(N) = \frac{\eta(\boldsymbol{\mu}, W_N)}{\eta(\boldsymbol{\mu}, W_{N-1})} \tag{4.15}$$

over Ξ_{train} along with the standard greedy sampling process Algorithm 1. This quantity is an indication of C_{sa}. We observe that assuming, in this particular case, that $C_{sa} = 2$ is a safe choice, for both cases. For this particular choice of C_{sa}, we illustrate in Figure 3 the savings in the loop over Ξ_{train} at each iteration of the standard greedy sampling. Indeed, the result

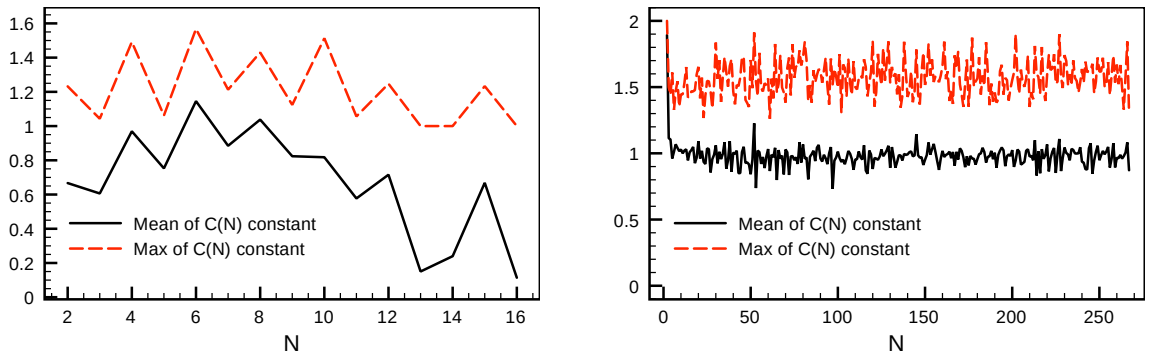

Figure 2: Evolution of the quantity (4.15) along the greedy sampling for $\mathcal{F}_1$ (left) and $\mathcal{F}_2$ (right).

indicates, that at each step N, the percentage of the work that needs to be done by using the saturation assumption compared to using the standard greedy algorithm and thus compares the different workloads, at each loop over Ξ_{train} of Algorithm 1 and 2. One observes that, while for the first example the improvement is already significant, one achieves an average (over the different values of N) saving of workload of about 50% (dotted red line) for the second example.

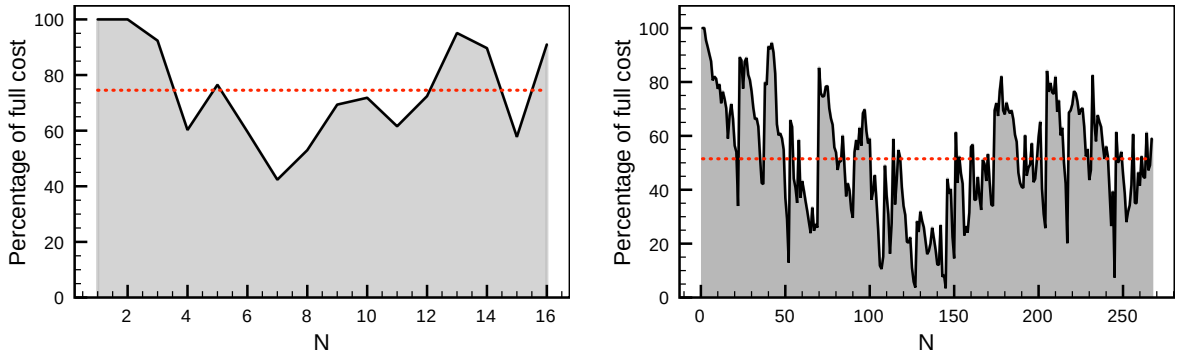

Figure 3: Percentage of work at each step N, using the Saturation Assumption based greedy algorithm, compared to the workload using the standard greedy algorithm, for $\mathcal{F}_1$ (left) and $\mathcal{F}_2$ (right).

4.2 Adaptively enriching greedy algorithm

Let us also test the adaptively enriching greedy algorithm (for convenience denoted by AEGA) first for the previously introduced function $\mathcal{F}_2$, and then for two test problems with high dimensional parameter spaces.

Considering $\mathcal{F}_2$, we set $M = 1'000, 5'000, 25'000$, $N_{sc} = 125'000$ and $C_{sc} = 2$. In Figure 4 we plot the convergence of the new algorithm (in red solid lines) and the corresponding error over the large control set Ξ_{train} (in dotted lines) consisting of 125'000 equidistant points. For comparison, we illustrate the convergence of the standard greedy algorithm using the train set Ξ_{train} (in dashed lines).

We observe that as we increase the value of M, the convergence error provided by the AEGA and the error of the AEGA over the larger control set become (not surprisingly) closer and closer and converge to the error provided by the standard EIM using training set Ξ_{train}.

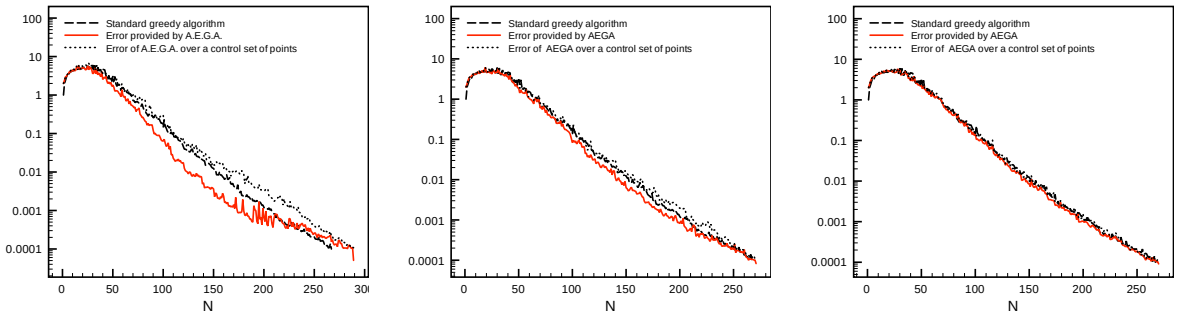

Figure 4: Convergence behavior of the adaptively enriching greedy algorithm for $\mathcal{F}_2$ with $M = 1$'000 (left), $M = 5$'000 (middle) and $M = 25$'000 (right).

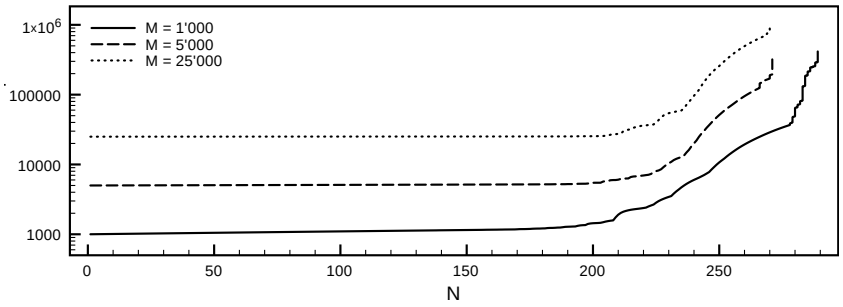

Figure 5: Evolution of the number of points where the accuracy is checked of the adaptively enriching greedy algorithm for different values of M (Safety check not included).

Figure 5 shows the evolution of the number of points which were part of the train set during new greedy algorithm for all values of M. We observe that in all three cases the error was also tested on at least 125'000 different points over the iterations of the algorithm, but of low number M at each iteration.

In Fig. 6, we present, for all values of M, two quantities. The first one consists of the percentage of work (w.r.t. M), at each step N, that needs to be effected and cannot be skipped by using the saturation assumption. The second one consists of the percentage of points (w.r.t. M) that remain in the train set after each step N. One can observe that at the end, almost all points in the train set are withdrawn (and thus the corresponding errors need to be computed). During a long time, we observe that the algorithm works with the initial train set until the error tolerance is satisfied on those points before they are withdrawn. In consequence, the accuracy need only to be checked for a low percentage of points of the trial set using the saturation assumption. Towards the end, the number of points that remain in the train set after each iteration decreases and consequently for each new sample point in the train set the saturation assumption cannot be used.

Remark 4.1. *Algorithm 3 is subject to randomness. However, we plot only one realization of the adaptively enriching greedy algorithm. Due to the presence of a lot of repeated randomness (each newly generated parameter value of Ξ is random) these illustration are still meaningful.*

Test 4.3. Introduce the following real-valued function

$$\mathcal{F}_3(\mathbf{x};\boldsymbol{\mu}) = \sin(2\pi\mu_1(x_1-\mu_2))\sin(2\pi\mu_3(x_2-\mu_4))\sin(4\pi\mu_5(x_1-\mu_6))\sin(4\pi\mu_7(x_2-\mu_8))$$

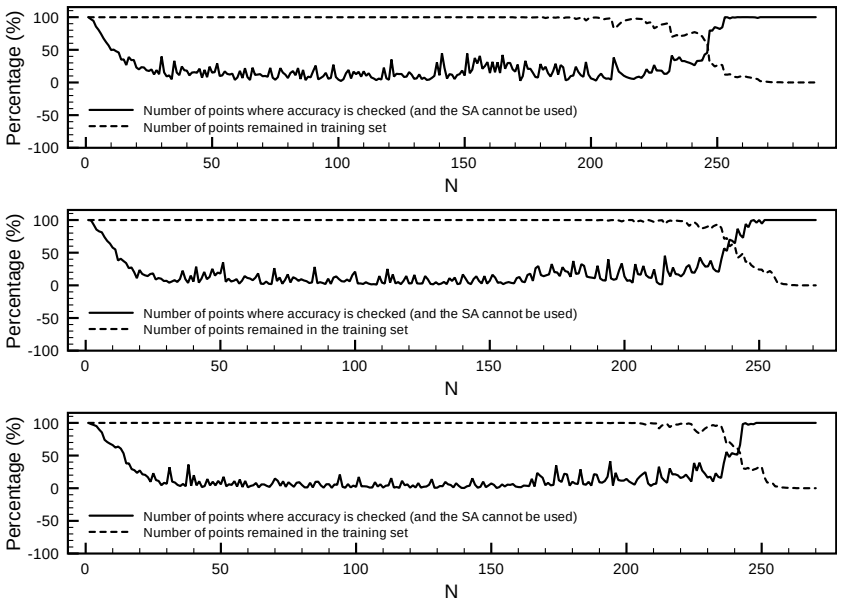

Figure 6: Percentage of work (effected at each step N) w.r.t. to the total number of points M and of the number of points remained in the train set (at each step N) of the adaptively enriching greedy algorithm for $\mathcal{F}_2$ and different values of $M = 1'000, 5'000, 25'000$.

with $\mathbf{x} = (x_1, x_2) \in \Omega = [0,1]^2$ and $\mu_1, \mu_3 \in [0,2]$, $\mu_2, \mu_4, \mu_6, \mu_8 \in [0,1]$, $\mu_5, \mu_7 \in [1,2]$. The domain Ω is divided into a grid of 100×100 equidistant points to build Ω_h. Recall that in the implementation, Ω_h is used to compute the norm $\|\cdot\|_{L^\infty(\Omega)}$.

In Fig. 7 and 8 we plot the convergence behaviour of the adaptive enriching greedy algorithm for the function $\mathcal{F}_3$ and $tol = 10^{-4}$. The value of N_{sc} is set for all different choices of $M = 100$, $M = 1'000$ and $M = 10'000$ equal to $N_{sc} = 100'000$. Figure 8 is a zoom of Figure 7 towards the end of the convergence to highlight how the safety check acts. We observe that in the case of $M = 10'000$ the safety check is passed in only a few attempts whereas for $M = 100$ the safety check fails 34 times until finally successful. This means that during each safety check there was at least one parameter value where the interpolation error was above the tolerance. Finally, passing the safety check means that the interpolation error on 100'000 random sample points was below the prescribed tolerance, in all three cases.

In Fig. 9, the accumulated number of points belonging to the train set Ξ is illustrated. We observe an increase of this quantity towards the end where the safety check is active. During this period all parameter points are withdrawn at each iteration, leading to the increase.

Test 4.4. Finally we consider the following real-valued function

$$\mathcal{F}_4(\mathbf{x};\boldsymbol{\mu}) = \left(1 + \exp\left(-\frac{(x_1-\mu_1)^2}{\mu_9} - \frac{(x_2-\mu_2)^2}{\mu_{10}}\right)\right)\left(1 + \exp\left(-\frac{(x_1-\mu_3)^2}{\mu_{11}} - \frac{(x_2-\mu_4)^2}{\mu_{12}}\right)\right)$$
$$\cdot \left(1 + \exp\left(-\frac{(x_1-\mu_5)^2}{\mu_{13}} - \frac{(x_2-\mu_6)^2}{\mu_{14}}\right)\right)\left(1 + \exp\left(-\frac{(x_1-\mu_7)^2}{\mu_{15}} - \frac{(x_2-\mu_8)^2}{\mu_{16}}\right)\right)$$

with $\mathbf{x} = (x_1, x_2) \in \Omega = [0,1]^2$ and $\mu_1, \ldots, \mu_8 \in [0.3, 0.7]$, $\mu_9, \ldots, \mu_{16} \in [0.01, 0.05]$. The domain $\Omega = [0,1]^2$ is divided into a grid of 100×100 equidistant points to build Ω_h.

In Fig. 10, the convergence behavior of the adaptive enriching greedy algorithm for the function $\mathcal{F}_4$ and $tol = 10^{-4}$ is plotted. The value of N_{sc} is set for all different choices of $M = 10'000$, $M = 100'000$ and $M = 1'000'000$ equal to $N_{sc} = 10'000'000$. Fig. 11 is

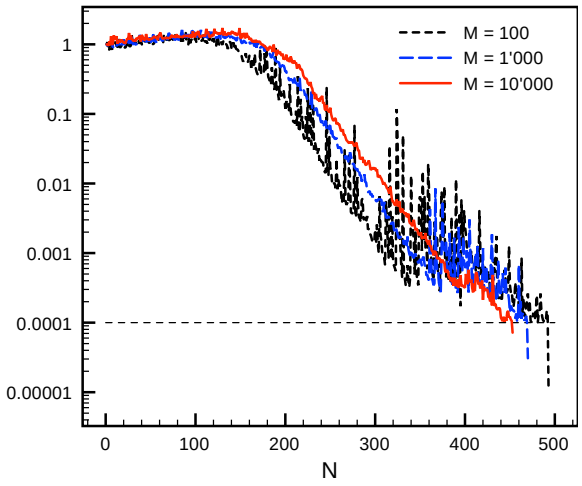

Figure 7: Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_3$.

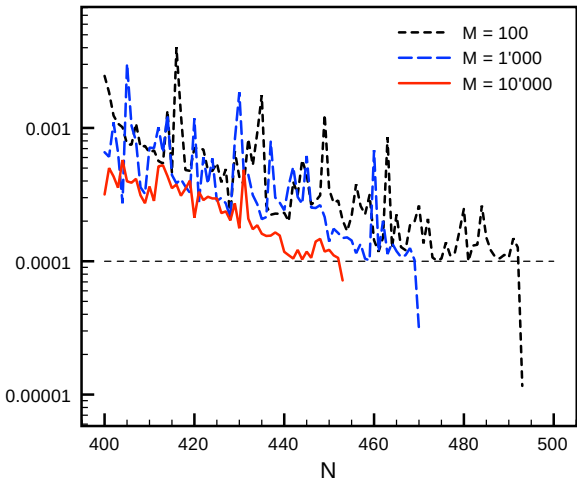

Figure 8: Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_3$ zoomed in towards the end where the safety check is active.

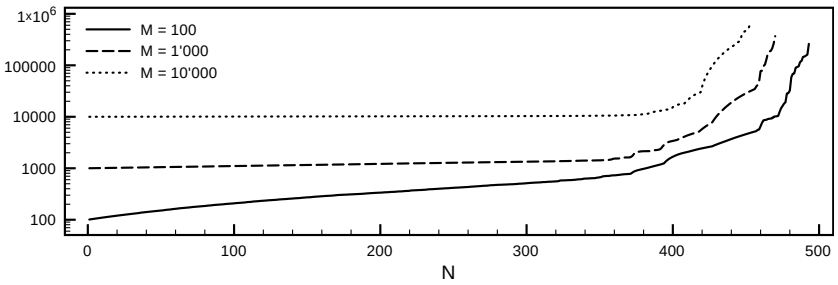

Figure 9: Evolution of the number of points where the accuracy is checked of the adaptively enriching greedy algorithm for different values of M (Safety check included).

again a zoom of Fig. 10 towards the end of the convergence. We observe a similar behavior as in the previous case. Note that in all three cases, the safety check is passed and the tolerance is satisfied for 10'000'000 subsequent parameter points. A bit surprisingly, both cases of $M = 10'000$ and $M = 100'000$ results in the same number of added modes N.

5 Application to the Reduced Basis Method

In this section we apply the improved greedy algorithms in the context of the Reduced Basis Method (RBM). As in the last section, we first test the benefit of the saturation assumption for some low dimensional parametric problems, and then proceed to test a problem with 15 parameters using the adaptively enriching greedy algorithm.

5.1 Saturation Assumption

Before performing the numerical test, we shall show that for some variational problems, the saturation assumption is satisfied with $C_{sa} = 1$ if the error is measured in the intrinsic energy

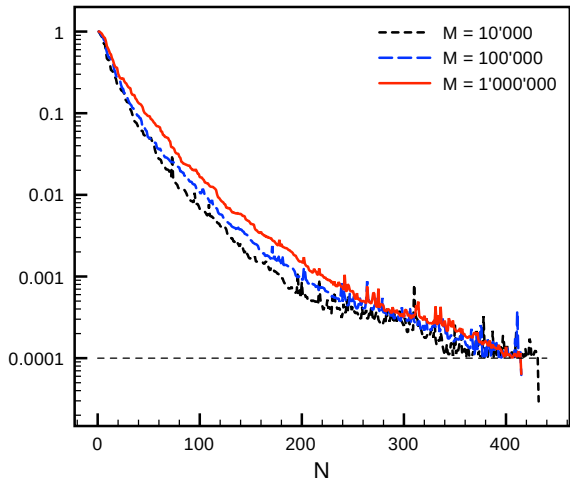

Figure 10: Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_4$.

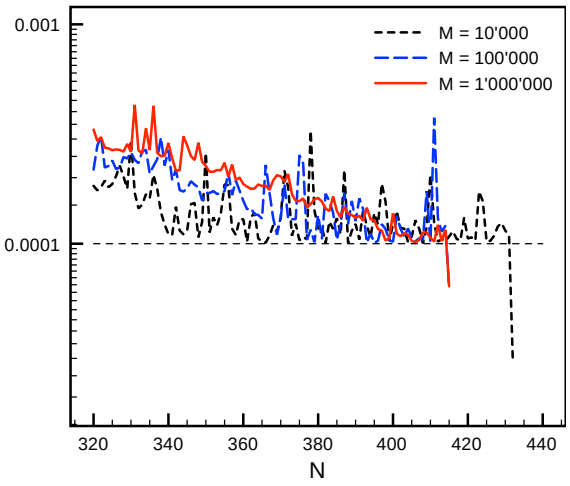

Figure 11: Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_4$ zoomed in towards the end where the safety check is active.

norm.

Suppose the variational problem is based on a minimization principle,

$$u = \text{argmin}_{v \in X} J(v), \tag{5.16}$$

where $J(v)$ is an energy functional. Then the finite element solution u^{fe} on $X^{fe} \subset X$ is

$$u^{fe} = \text{argmin}_{v \in X^{fe}} J(v). \tag{5.17}$$

Similarly, the reduced basis solution u_N^{rb} on $X_N^{rb} \subset X^{fe} \subset X$ is

$$u_N^{rb} = \text{argmin}_{v \in X_N^{rb}} J(v). \tag{5.18}$$

The error between u^{fe} and u_N^{rb} can be measured by the nonnegative quantity

$$J(u_N^{rb}) - J(u^{fe}). \tag{5.19}$$

Since $W_N \subset W_{N+1}$ and thus $X_N^{rb} \subset X_{N+1}^{rb}$, we have $J(u_{N+1}^{rb}) \leq J(u_N^{rb})$ and in consequence there holds

$$J(u_{N+1}^{rb}) - J(u^{fe}) \leq J(u_N^{rb}) - J(u^{fe}). \tag{5.20}$$

Observe that if the error is measured exactly as $J(u_N^{rb}) - J(u^{fe})$, the saturation assumption is satisfied with $C_{sa} = 1$.

Let us give an example of the above minimization principle based problem. Consider the symmetric coercive elliptic problem,

$$\left\{ \begin{array}{rcl} -\nabla \cdot (\alpha(\boldsymbol{\mu}) \nabla u) & = & f \quad \text{in} \quad \Omega, \\ u & = & 0 \quad \text{on} \quad \Gamma_D, \\ \alpha(\boldsymbol{\mu}) \nabla u \cdot \mathbf{n} & = & g \quad \text{on} \quad \Gamma_N, \end{array} \right. \tag{5.21}$$

where Γ_D and Γ_N are the Dirichlet and Neumann boundaries of $\partial\Omega$, and $\Gamma_D \cup \Gamma_N = \partial\Omega$. For simplicity, we assume $\Gamma_D \neq \emptyset$. The functions f and g are L^2–functions on Ω and Γ_N

respectively. The parameter dependent diffusion coefficients $\alpha(\boldsymbol{\mu})$ are always positive. Let $X = H^1_D(\Omega) := \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$. Then its variational formulation is

$$a(u, v; \boldsymbol{\mu}) = f(v), \quad \forall v \in X, \tag{5.22}$$

where

$$a(u, v; \boldsymbol{\mu}) = \int_\Omega \alpha(\boldsymbol{\mu}) \nabla u \nabla v dx, \quad f(v) = \int_\Omega f v dx + \int_{\Gamma_N} g v ds.$$

The energy functional is

$$J(v) = \tfrac{1}{2} \|(\alpha(\boldsymbol{\mu}))^{1/2} \nabla v\|^2_{0,\Omega} - f(v),$$

and we have

$$\begin{aligned} J(u^{rb}_N) - J(u^{fe}) &= \tfrac{1}{2} \left(\|\alpha^{1/2} \nabla u^{rb}_N\|^2_{0,\Omega} - \|\alpha^{1/2} \nabla u^{fe}\|^2_{0,\Omega} \right) - f(u^{rb}_N - u^{fe}) \\ &= \tfrac{1}{2} \|\alpha^{1/2} \nabla (u^{rb}_N - u^{fe})\|^2_{0,\Omega} + \int_\Omega \alpha \nabla u^{fe} \cdot \nabla (u^{rb}_N - u^{fe}) dx - f(u^{rb}_N - u^{fe}) \\ &= \tfrac{1}{2} \|\alpha^{1/2} \nabla (u^{rb}_N - u^{fe})\|^2_{0,\Omega}. \end{aligned}$$

In the last step, Galerkin orthogonality

$$\int_\Omega \alpha \nabla u^{fe} \cdot \nabla w dx = f(w) \quad \forall\, w \in X^{fe}$$

is used. The above analysis is standard, see [13].

If we measure the error by this intrinsic energy norm $\|\alpha(\boldsymbol{\mu})^{1/2} \nabla v\|_{0,\Omega}$, the error satisfies the saturation assumption with $C_{sa} = 1$,

$$\|\alpha(\boldsymbol{\mu})^{1/2} \nabla (u^{rb}_M - u^{fe})\|_{0,\Omega} \leq \|\alpha(\boldsymbol{\mu})^{1/2} \nabla (u^{rb}_N - u^{fe})\|_{0,\Omega} \quad \text{for } 0 < N < M. \tag{5.23}$$

Unfortunately, even for the above model problem, the a posteriori error estimator used in the reduced basis method is not equivalent to the energy norm of the error exactly.

For the problem (5.21), we choose the underlying norm of X and X^{fe} to be the H^1–semi-norm $\|v\|_X = \sqrt{\int_\Omega (\nabla v)^2 dx}$. Notice, due to the fact that the dual norm of the X^{fe}–norm needs to be computed, involving an inverse of the matrix associated with the X^{fe}–norm, this X^{fe}–norm cannot be parameter dependent. Otherwise, we must invert a matrix of a size comparable to the finite element space every time for a new parameter in the computation of error estimator. This would clearly result in an unacceptable cost.

The functional based error estimator for (5.21) is defined as

$$\eta(\boldsymbol{\mu}; W_N) = \frac{\|f\|_{X'} \|r(\cdot; \boldsymbol{\mu})\|_{X'}}{\beta(\boldsymbol{\mu}) |f(u^{rb}_N(\boldsymbol{\mu}))|}. \tag{5.24}$$

For this simple problem and this specific choice of norm, the stability constant is $\beta(\boldsymbol{\mu}) = \min_{\mathbf{x} \in \Omega} \{\alpha(\boldsymbol{\mu})\}$. Note that

$$f(v) = a(v, v; \boldsymbol{\mu}) = \|\alpha(\boldsymbol{\mu})^{1/2} \nabla v\|^2_{0,\Omega}, \quad \forall v \in X.$$

The error estimator $\eta(\boldsymbol{\mu}; W_N)$ we used here is in fact an estimation of the relative error measured by the square of the intrinsic energy norm. In principle, since we already proved the error measured in energy norm should be monotonically decreasing (or more precisely, non-increasing), the constant C_{sa} can be chosen to be 1. However, if we examine the error estimator

$\eta(\boldsymbol{\mu}; W_N)$ defined in (5.24) carefully, we find that for a fixed parameter $\boldsymbol{\mu}$, the values of $\|f\|_{X'}$ and $\beta(\boldsymbol{\mu})$ are fixed, the change of the value of $|f(u_N^{rb}(\boldsymbol{\mu}))|$ is quite small if the approximation $u_N^{rb}(\boldsymbol{\mu})$) is already in the asymptotically region. The only problematic term is $\|r(\cdot;\boldsymbol{\mu})\|_{X'}$. This term is measured in a dual norm of a parameter independent norm (the H^1–semi-norm), not in the dual norm of the intrinsic energy norm $\|\alpha(\boldsymbol{\mu})^{1/2}\nabla v\|_{0,\Omega}$. It is easy to see that

$$\|\alpha^{1/2}\nabla e\|_{0,\Omega}^2 = f(e) \le \|f\|_{X'}\|e\|_X \le \|f\|_{X'}\frac{\|r(\cdot,\boldsymbol{\mu})\|_{X'}}{\beta(\boldsymbol{\mu})}.$$

Thus, the error estimator $\eta(\boldsymbol{\mu}; W_N)$ is only an upper bound of the the relative error measured by the square of the intrinsic energy norm. When the variation of α with respect to $\boldsymbol{\mu}$ is large, the difference between the H^1–semi-norm and the energy norm $\|\alpha^{1/2}\nabla v\|_{0,\Omega}$ may be large and we may find the error estimator is not monotonically decreasing as the number of basis functions increasing.

In Test 5.1 below, we use a moderate variation of $\alpha \in [1/10, 10]$ and we observe that the saturation assumption is satisfied with $C_{sa} = 1$. In Tests 5.2 and 5.3, a wider range of $\alpha \in [1/00, 100]$ is used. For the corresponding error estimator, the saturation assumption is not satisfied with $C_{sa} = 1$ but for a larger C_{sa}.

Test 5.1. In this example, we will show that for a simple coercive elliptic problem, the saturation assumption is satisfied numerically with $C_{sa} = 1$ for some type of error estimators.

Consider the following thermal block problem, which is a special case of (5.21), see also [14]. Let $\Omega = (0,1)^2$, and

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

where $\alpha = 10^{2\boldsymbol{\mu}-1}$ in $R_1 = (0, 0.5)^2$ and $\alpha = 1$ in $R_{rest} = \Omega\backslash(0, 0.5)^2$. We choose the one-dimensional parameter domain $\mathcal{D}$ of $\boldsymbol{\mu}$ to be $[0, 1]$, which corresponds to $\alpha \in [1/10, 10]$ in R_1. Note that in this particular case the vector of parameters $\boldsymbol{\mu}$ is indeed a scaler parameter μ. The variational problem is given in (5.22). The output functional is chosen to be $s(u) = f(u)$.

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 $K \in \mathcal{T}$. We then define our finite 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 is seeking $u^{fe}(\boldsymbol{\mu}) \in X^{fe}$, such that

$$a(u^{fe}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v) \quad v \in X^{fe}. \tag{5.26}$$

With a set of N reduced basis elements W_N and the corresponding X_N^{rb}, and for a given parameter $\boldsymbol{\mu}$, we solve the following reduced basis approximation problem: Seeking $u_N^{rb}(\boldsymbol{\mu}) \in X_N^{rb}$, such that

$$a(u_N^{rb}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v) \quad \forall v \in X_N^{rb}. \tag{5.27}$$

We choose 101 equidistance sample points in $\mathcal{D} = [0, 1]$, i.e., $\Xi_{train} = \{\frac{i}{100}, i = 0, 1, 2, \cdots, 100\}$, a standard reduced basis greedy algorithm with the error estimator defined in (5.24) being used. The first parameter $\boldsymbol{\mu}^1$ is chosen to be 0.5, that is, $S_1 = \{0.5\}$. The greedy algorithm chooses the 2nd, 3rd, 4th, and 5th parameters as $\{0, 1.0, 0.15, 0.81\}$, so $S_5 = \{0.5, 0, 1, 0.15, 0.81\}$. We compute the reduced basis set W_N and the reduced basis spaces X_N^{rb} corresponding to S_N,

$N = 1, \cdots, 5$. Then, for all points $\boldsymbol{\mu}$ in Ξ_{train}, $\eta(\boldsymbol{\mu}; W_N)$, $N = 1, \cdots, 5$ is computed. Figure 12 shows the plots of η for each points in Ξ_{train} with number of reduced basis $= 1, \cdots, 5$. Clearly, we see that for each point $\boldsymbol{\mu} \in \Xi_{train}$, $\eta(\boldsymbol{\mu}; W_5) \leq \eta(\boldsymbol{\mu}; W_4) \leq \eta(\boldsymbol{\mu}; W_3) \leq \eta(\boldsymbol{\mu}; W_2) \leq \eta(\boldsymbol{\mu}; W_1)$. For this problem, the saturation assumption is clearly satisfied for $C_{sa} = 1$ in the first several steps.

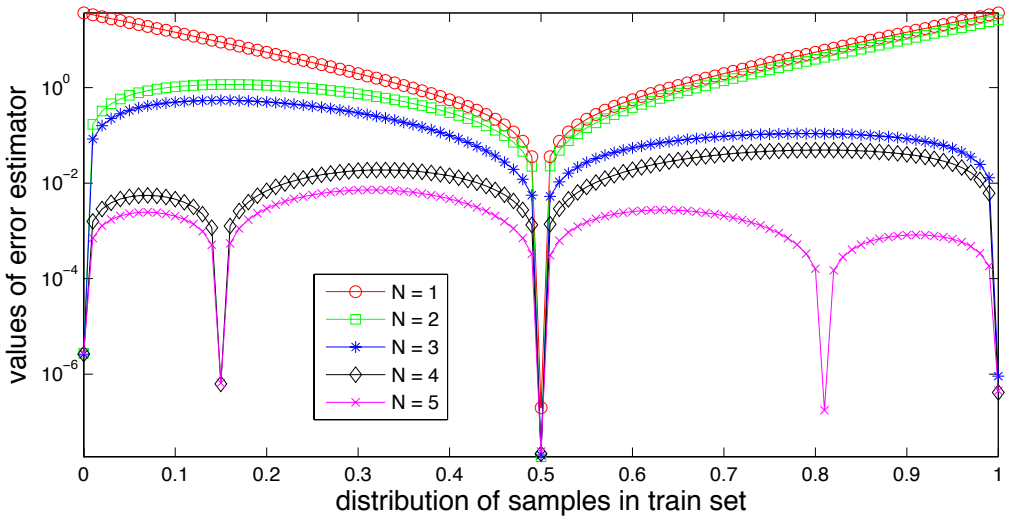

Figure 12: A verification of Saturation Assumption for a symmetric positive definite problem with 1 parameter.

Test 5.2. We now test the potential for cost savings when the saturation assumption based algorithm is used in connection with the reduced basis method.

For equation (5.25), we decompose the domain Ω into 3 subdomains: $R_1 = (0, 0.5) \times (0.5, 1)$, $R_2 = (0.5, 1) \times (0, 0.5)$, and $R_3 = (0, 1)^2 \backslash (R_1 \cup R_2)$. The diffusion constant α is set to be

$$\alpha = \begin{cases} \alpha_i = 100^{2\mu_i - 1}, & x \in R_i, i = 1, 2, \\ \alpha_3 = 1, & x \in R_3, \end{cases}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2) \in [0, 1]^2$. The domain of $\alpha_i, i = 1, 2$ is set to $[1/100, 100]$. The bilinear form then becomes

$$a(u, v; \boldsymbol{\mu}) = \sum_{i=1}^{2} 100^{2\mu_i - 1} \int_{R_i} \nabla u \cdot \nabla v dx + \int_{R_3} \nabla u \cdot \nabla v dx. \tag{5.28}$$

All other forms and spaces are identical to the ones of Test 5.1. Further, let $N_{100} = \{0, 1, 2, \cdots, 100\}$, the train set is given by

$$\Xi_{train} = \{(x(i), y(j)) : x(i) = \tfrac{i}{100}, y(j) = \tfrac{j}{100}, \text{ for } i \in N_{100}, j \in N_{100}\}.$$

We set the tolerance to be 10^{-3} and use the error estimator defined in (5.24).

Both the standard greedy algorithm and the saturation assumption based greedy algorithm requires 20 reduced bases to reduce the estimated error to less than the tolerance. For this problem, if the error is measured in the intrinsic energy norm, we can choose $C_{sa} = 1$ as indicated above. Due to the inaccuracy of the error estimator, the saturation assumption based algorithm chooses a slightly different set of S_N. If we choose $C_{sa} = 1.1$ slightly larger, we get however the same set S_N as the standard greedy algorithm.

See Fig. 13 for a comparisons of the workloads using the standard algorithm and the new approach based on the saturation assumption with $C_{sa} = 1$ and $C_{sa} = 1.1$, respectively. The mean percentage is computed as $\sum_{i=1}^{N}(\text{percentage at step } i)/N$. The mean percentages of $C_{sa} = 1$ and $C_{sa} = 1.1$ is about 32% and 34% respectively. Since the computational cost for each evaluation of $\eta(\boldsymbol{\mu}; N)$ is of $O(N^3)$, the average percentages do not represent the average time saving, and only give a sense of the saving of the workloads at each step.

In Fig. 14, we present the selections of the sample points S_N by the standard algorithm and the Saturation Assumption based greedy algorithm with $C_{sa} = 1$. Many sample points are identical. In the case $C_{sa} = 1.1$, the samples S_N are identical to the ones of the standard algorithm.

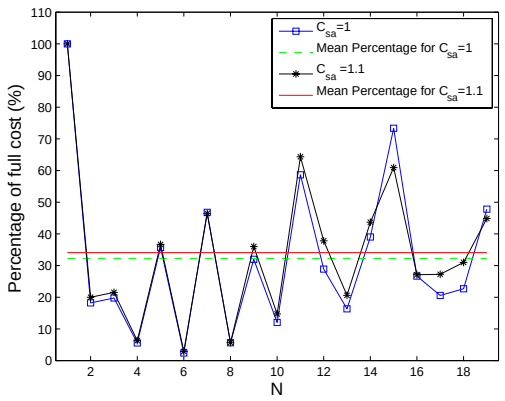

Figure 13: Percentage of work at each step N using saturation assumption based greedy algorithm with $C_{sa} = 1$, compared to the workload using the standard greedy algorithm for Test 5.2.

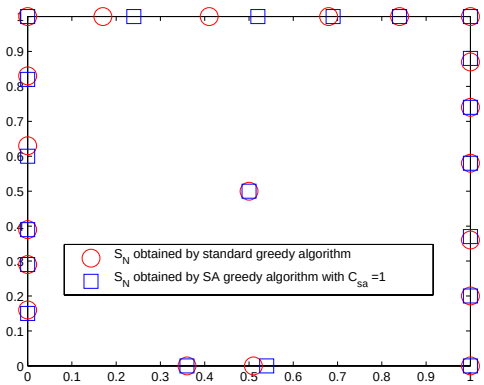

Figure 14: S_N obtained by standard and saturation assumption based greedy algorithms for Test 5.2.

Test 5.3. We continue and test a problem with 3 parameters. For (5.25), we decompose the domain Ω into 4 subdomains: $R_k = (\frac{i-1}{2}, \frac{i}{2}) \times (\frac{j-1}{2}, \frac{j}{2})$, for $i = 1, 2$, $j = 1, 2$, and $k = 2(i-1)+j$. The diffusion constant α is set to be

$$\alpha = \begin{cases} \alpha_k = 100^{2\mu_k - 1} & x \in R_k, k = 1, 2, 3, \\ \alpha_4 = 1 & x \in R_4, \end{cases}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3) \in [0, 1]^3$. The bilinear form is

$$a(u, v; \boldsymbol{\mu}) = \sum_{k=1}^{3} 100^{2\mu_k - 1} \int_{R_k} \nabla u \cdot \nabla v dx + \int_{R_4} \nabla u \cdot \nabla v dx, \tag{5.29}$$

All other forms and spaces are identical to the ones of Test 5.1. We again choose the output functional based error estimator as Test 4.1 and the tolerance is set to be 10^{-3}. Let $N_{50} = \{0, 1, 2, \cdots, 50\}$, the train set is given by

$$\Xi_{train} = \{(x(i), y(j), z(k)) : x(i) = \tfrac{i}{50}, y(j) = \tfrac{j}{50}, z(k) = \tfrac{k}{50}, i \in N_{50}, j \in N_{50}, k \in N_{50}\}.$$

The standard greedy algorithm needs 24 reduced bases to reduce the estimated error less than the tolerance. If C_{sa} is chosen to be 1, 25 reduced bases are needed to reduce the estimated error less than the tolerance. The set S_N obtained by the Saturation Assumption based algorithm

with $C_{sa} = 1$ is also different from the standard algorithm. As discussed before, this is mainly caused by the inaccuracy of the error estimator. If we choose $C_{sa} = 3$, we obtain the same sample points S_N as the standard greedy algorithm. See Figure 15 for the comparisons of the workloads using the standard algorithm and the saturation assumption based algorithm with $C_{sa} = 1$ and $C_{sa} = 3$, respectively. The mean percentages of workload for $C_{sa} = 1$ and $C_{sa} = 3$ are 21.6% and 33.7%, respectively.

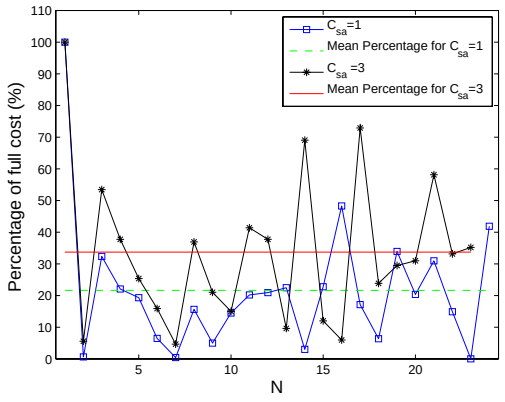

Figure 15: Percentage of work at each step N using saturation assumption based greedy algorithm with $C_{sa} = 1$ and $C_{sa} = 3$, compared to the work load using the standard greedy algorithm for Test 5.3.

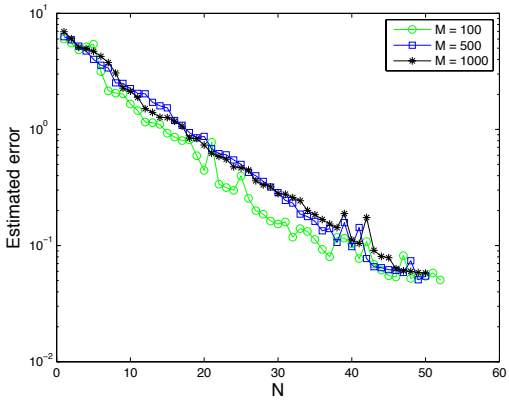

Figure 16: Convergence behavior of the adaptively enriching greedy algorithm for Test 5.4 with M = 100, 500, and 1'000.

Remark 5.1. *For the type of compliance problem discussed in Tests 5.1, 5.2 and 5.3, other types of error estimator are suggested in [14]:*

$$\eta^e(\boldsymbol{\mu}, W_N) := \frac{\|r(\cdot;\boldsymbol{\mu})\|_{(X^{fe})'}}{\beta^{fe}(\boldsymbol{\mu})^{1/2}|u_N^{rb}(\boldsymbol{\mu})|} \quad \textit{and} \quad \eta^s(\boldsymbol{\mu}, W_N) := \frac{\|r(\cdot;\boldsymbol{\mu})\|^2_{(X^{fe})'}}{\beta^{fe}(\boldsymbol{\mu})|s_N^{rb}(\boldsymbol{\mu})|}. \tag{5.30}$$

As discussed above, the most important term in the error estimator of the Saturation Assumption is the dual norm of the residual $\|r(\cdot;\boldsymbol{\mu})\|_{(X^{fe})'}$. For the error estimator $\eta^e(\boldsymbol{\mu}; W_N)$, the behavior is similar to that of $\eta(\boldsymbol{\mu}; W_N)$. For the error estimator $\eta^s(\boldsymbol{\mu}; W_N)$, the dual norm of the residual is squared. The dual norm is computed with respect to a parameter independent reference norm. The square makes the difference between the dual norm based on the intrinsic energy norm and on the reference norm larger. Normally, if the error estimator $\eta^s(\boldsymbol{\mu}; W_N)$ is used, we need a more conservative C_{sa}. Numerical tests show that even if $C_{sa} = 20$ is set for Test 5.3, the workload of the saturation assumption based algorithm is still only about 45% (on average) of the standard greedy algorithm.

5.2 Adaptively enriching greedy algorithm

Test 5.4 We test the adaptively enriching greedy algorithm for the reduced basis method for a problem with 15 parameters.

For (5.25), we decompose the domain Ω 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 set to be

$$\alpha = \begin{cases} \alpha_k = 5^{2\mu_k - 1}, & x \in R_k, \quad k = 1, 2, \cdots, 15, \\ \alpha_{16} = 1, & x \in R_{16}. \end{cases}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_{15}) \in [0,1]^{15}$. The domain of $\alpha_k, k = 1, 2, \ldots, 15$, is given by $[1/5, 5]$. The bilinear form then consists of

$$a(u, v; \boldsymbol{\mu}) = \sum_{k=1}^{15} 5^{2\mu_k - 1} \int_{R_k} \nabla u \cdot \nabla v dx + \int_{R_{16}} \nabla u \cdot \nabla v dx. \tag{5.31}$$

All other forms and spaces are identical to the ones of Test 5.1. Due to the many jumps of the coefficients along the interfaces of the subdomains, the solution space of this problem is very rich. We set $C_{sa} = 1$, $tol = 0.05$, $N_{sc} = 10'000$. Since there is a "safety check" step to ensure the quality of the reduced bases, we should not worry that the choice of constant C_{sa} is too aggressive. We test three cases: $M = 100$, $M = 500$, and $M = 1'000$. The convergence for one realization is plotted in Fig. 16. The number of reduced basis for $M = 100$ is 52, and for the other two cases is 50. This suggests us that a bigger M will lead to a smaller number of the bases. The percentage of work (effected at each step N) with respect to the total number of points M and of the number of points remained in the train set (at each step N) of the adaptively enriching greedy algorithm for different values of $M = 100$, 500, and 1000 are shown in 17. At the beginning stage, the estimated errors are larger than tolerance for almost all points in the train set, when the RB space is rich enough, more and more points are removed, and eventually, almost all points in the trained set are removed in later stages. For the number of the points where the error estimators are computed, that is, the saturation assumption part, it behaves like the Algorithm 2 with a fixed train set since the train set barely changed in the beginning. In the later stage, since almost all points are new points, the percentage of the number of the points where the error estimators are computed is close to 100%.

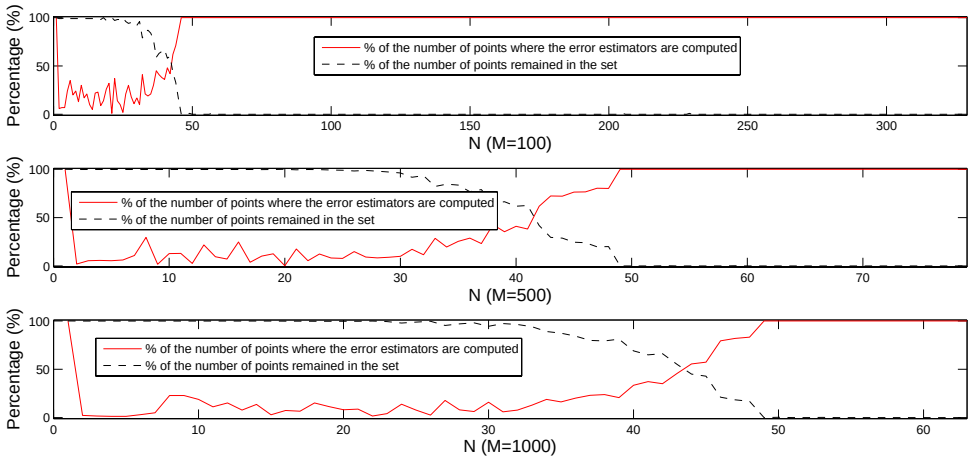

Figure 17: Percentage of work (effected at each step N) w.r.t. the total number of points M and of the number of points remained in the train set (at each step N) of the adaptively enriching greedy algorithm for Test 5.4 and different values of M = 100, 500, and 1'000.

Remark 5.2. *For a fixed M and provided the algorithm is performed several times, we observe that even though the train set is generated randomly each time, the numbers of the reduced bases needed to reduce the estimated error to the prescribed tolerance are very similar. This means that even if we start with a different and very corse random train set, the algorithm is quite stable in the sense of capturing the dimensionality of the reduced basis space.*

6 Conclusions

In this paper, we propose two enhanced greedy algorithms designed to improve sampling approaches for high dimensional parameters spaces. We have demonstrated the efficiency of these new techniques on the empirical interpolation method (EIM) and the reduced basis method (RBM). Among other key observations, we have documented the potential for substantial savings over standard greedy algorithm by utilization of a simple saturation assumption. Combining this with a "safety check" step guaranteed adaptively enriching greedy algorithm, the EIM and RBM for problems with a high number of parameters are now workable and more robust.

With some possible modifications, the algorithms developed here can be applied to other scenarios where a greedy algorithm is needed, for example, the successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup constants [3, 4, 5, 10].

Acknowledgement

The first and third authors acknowledge partial support by OSD/AFOSR FA9550-09-1- 0613.

References

[1] R. E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Math. Comp., 44(1985), pp.303-320. 3.2

[2] M. Barrault, N. C. Nguyen, Y. Maday, and A. T. Patera, *An "empirical interpolation" method: Application to efficient reduced-basis discretization of partial differential equations*, C. R. Acad. Sci. Paris, Ser. I, 339 (2004), pp. 667–672. 2.2

[3] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodriguez, *A monotonic evaluation of lower bounds for inf-sup stability constants in the frame of reduced basis approximations*, C. R. Acad. Sci. Paris, Ser. I, 346 (2008), pp. 1295–1300. 6

[4] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodriguez, *Improved successive constraint method based a posteriori error estimate for reduced basis approximation of 2d Maxwells problem*, ESAIM: M2AN, 43 (2009), pp. 1099–1116. 6

[5] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodriguez, *Certified reduced basis methods and output bounds for the harmonic maxwell equations*, SIAM J. Sci. Comput., 32(2010), pp. 970-996. 6

[6] 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. 1

[7] J. L. Eftang and B. Stamm, *Parameter multi-domain hp empirical interpolation*, NTNU preprint, Numerics no. 3/2011, 2011. 1

[8] 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. 2.2, 4.1

[9] 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. 2.2

[10] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera, *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*, C. R. Acad. Sci. Paris, Ser. I, 345 (2007), pp. 473–478. 6

[11] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. H. Pau, *A general multipurpose interpolation procedure: the magic points*, Commun. Pure Appl. Anal., 8(1):383–404, 2009. 2.2

[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. 2.1

[13] S. Repin, *A Posteriori Estimates for Partial Differential Equations*, Walter de Gruyter, Berlin, 2008. 5.1

[14] 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. 2.1, 2.1, 5.1, 5.1

[15] V.N. Temlyakov, *Greedy Approximation*, Acta Numerica (2008), 235-409. 1