ESAIM: M2AN 48 (2014) 259–283
DOI: 10.1051/m2an/2013100

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

EFFICIENT GREEDY ALGORITHMS FOR HIGH-DIMENSIONAL PARAMETER SPACES WITH APPLICATIONS TO EMPIRICAL INTERPOLATION AND REDUCED BASIS METHODS*

Jan S. Hesthaven[1], Benjamin Stamm[2,3] and Shun Zhang[4]

Abstract. We propose two new algorithms to improve 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 saturation assumption of the error in the greedy algorithm, is shown to result in a significant reduction of the workload over the standard greedy algorithm. In a further improved approach, this is combined with an algorithm in which the train set for the greedy approach is adaptively sparsified and enriched. A safety check step is added at the end of the algorithm to certify the quality of the sampling. Both these techniques are applicable to high-dimensional problems and we shall demonstrate their performance on a number of numerical examples.

Mathematics Subject Classification. 41A05, 41A46, 65N15, 65N30.

Received August 4, 2011.
Published online January 10, 2014.

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. One 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 interpolating polynomial representation. Such representations, known as linear approximations, are independent of the function being approximated and have been used widely. However, as problems become complex and high-dimensional, the direct extension of such ideas quickly becomes prohibitive, if even possible.

More recently, there has been an increasing interest in the development of methods where 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,

Keywords and phrases. Greedy algorithm, reduced basis method, empirical interpolation method.

* *This work is supported in part by OSD/AFOSR FA9550-09-1-0613.*

[1] Division of Applied Mathematics, Box F, Brown University, 182 George St., Providence, RI 02912, USA. Jan.Hesthaven@Brown.edu

[2] CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France. stamm@ann.jussieu.fr

[3] UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France.

[4] Department of Mathematics, City University of Hong Kong, Kowloon Tong, Hong Kong, China. shun.zhang@cityu.edu.hk

Article published by EDP Sciences

© EDP Sciences, SMAI 2014

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

measured in some reasonable manner. This approach, known as a greedy approximation, seeks to maximize a given measure, say the maximum error, and enrich the basis to eliminate this specific error contribution, hence increasing the accuracy in an optimal manner. Such greedy approaches have proven themselves to be particularly valuable for the approximation of high-dimensional problems where simple approaches are excluded due to the curse of dimensionality. For a detailed recent overview of such ideas in a general context, we refer to [27].

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 train set $\Xi_{\text{train}} \subset \mathcal{D}$ over which a functional is evaluated to select the next element of the approximation. When the number of parameters is high, the size of this train set quickly becomes large, rendering the computational cost substantial and perhaps even prohibitive. Hence, since a fine enough train set is not realistic in practice, one is faced with the problem of ensuring the quality of the approximation under a coarse train set. It is worth noting that when dealing with certain high dimensional problems, one may encounter situations where the optimal basis set itself is of large size. This situation is, however, caused by the general complexity of the problem and we shall not discuss this further. Strategies for such cases are discussed in see [10, 11, 17, 20].

There are some other discussions on greedy algorithms in the context of reduced basis and model reduction. The standard greedy algorithm for reduced basis methods is introduced in [28]. The convergence results of the greedy algorithm in the reduced basis method can be found in [3,4]. Greedy algorithms for parametric parabolic equations are given in [14]. Reduced basis methods based on adaptive parameter grids can be found in [15, 16]. Multi-stage greedy approach for reduced basis methods is suggested in [26]. In [5,6], an optimized search greedy approach is proposed. The local adaptive greedy algorithm is discussed in [20].

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 terms, 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. With this simple and reasonable saturation assumption on the error or the error estimator, we demonstrate how to modify the greedy algorithm such that errors are only computed for those points in Ξ_{train} with a large enough predicted error, resulting in the total workload of the standard greedy algorithm being significantly reduced.

The second algorithm is an adaptively enriching greedy algorithm. In this approach, the samples in the train set are 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 the search for a new parameter, the size of the train set is maintained at a reasonable number. The saturation assumption technique is naturally included in the algorithm. This algorithm can be applied to problems with a high number of parameters with substantial savings.

We emphasize that, compared to the saturation assumption greedy algorithm, the adaptively enriching greedy algorithm is an improved version, and a more general and robust algorithm. The adaptively enriching greedy algorithm is recommended in the real computation.

What remains of this paper is organized as follows. In Section 2, we discuss the role of greedy sampling 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 the two enhanced greedy techniques. This is followed in Sections 4 and 5 by a number of detailed numerical examples for the empirical interpolation method and reduced basis techniques, respectively, to illustrate the advantages of 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 provide a brief background on two different computational techniques, both of which rely on greedy approximation techniques, serving as a motivation for the subsequent discussion of the modified greedy techniques.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

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 several suitable methods for this, we focus here on the reduced basis method (RBM) [21, 25] which has proven itself to be a 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. Here X is a suitable function 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 $u^{\mathrm{fe}}(\boldsymbol{\mu}) \in X^{\mathrm{fe}}$ be the numerical solution of the following Galerkin problem,

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

and let $s^{\mathrm{fe}}(\boldsymbol{\mu}) = \ell(u^{\mathrm{fe}}(\boldsymbol{\mu}); \boldsymbol{\mu})$ 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 assumed to be satisfied for $\boldsymbol{\mu}$-dependent positive constants $\beta(\boldsymbol{\mu})$ and $\beta^{\mathrm{fe}}(\boldsymbol{\mu})$, respectively: there exists a $\beta_0 > 0$,

$$\beta(\boldsymbol{\mu}) = \inf_{u \in X} \sup_{v \in X} \frac{a(u, v; \boldsymbol{\mu})}{\|u\|_X \|v\|_X} \geq \beta_0 \text{ and } \beta^{\mathrm{fe}}(\boldsymbol{\mu}) = \inf_{u \in X^{\mathrm{fe}}} \sup_{v \in X^{\mathrm{fe}}} \frac{a(u, v; \boldsymbol{\mu})}{\|u\|_{X^{\mathrm{fe}}} \|v\|_{X^{\mathrm{fe}}}} \geq \beta_0, \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{2.3}$$

where $\|\cdot\|_X$ and $\|\cdot\|_{X^{\mathrm{fe}}}$ are norms of the spaces X and X^{fe}, respectively.

For a collection of N parameters $S_N = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N\}$ in the parameter domain $\mathcal{D} \subset \mathbb{R}^p$, let $W_N = \{u^{\mathrm{fe}}(\boldsymbol{\mu}^1), \ldots, u^{\mathrm{fe}}(\boldsymbol{\mu}^N)\}$, where $u^{\mathrm{fe}}(\boldsymbol{\mu}^i)$ is the numerical solution of problem (2.2) corresponding to the parameter values $\boldsymbol{\mu}^i$, for $1 \leq i \leq N$. Define the reduced basis space as $X_N^{\mathrm{rb}} = \mathrm{span}\{W_N\}$.

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

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

with the corresponding value of the output functional

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

The inf-sup constant for reduced basis method is defined as:

$$\beta_N^{\mathrm{rb}}(\boldsymbol{\mu}) = \inf_{X_N^{\mathrm{rb}}} \sup_{X_N^{\mathrm{rb}}} \frac{a(u, v; \boldsymbol{\mu})}{\|u\|_{X^{\mathrm{fe}}} \|v\|_{X^{\mathrm{fe}}}}.$$

For coercive problems, $\beta^{\mathrm{rb}}(\boldsymbol{\mu}) \geq \beta_0 > 0$. For non-coercive problems, the positivity of $\beta_N^{\mathrm{rb}}(\boldsymbol{\mu})$ is not ensured. A modified test space Y_N^{rb} can be introduced by the co-called supremizer operator to ensure

$$\beta_N^{\mathrm{rb}}(\boldsymbol{\mu}) = \inf_{X_N^{\mathrm{rb}}} \sup_{Y_N^{\mathrm{rb}}} \frac{a(u, v; \boldsymbol{\mu})}{\|u\|_{X^{\mathrm{fe}}} \|v\|_{X^{\mathrm{fe}}}} \geq \beta^{\mathrm{fe}}(\boldsymbol{\mu}) > \beta_0 > 0, \quad \forall \boldsymbol{\mu} \in \mathcal{D},$$

see the discussion in Section 7.2.2 of [22] and [24, 29].

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

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

Define the error function $e(\boldsymbol{\mu}) = u_N^{\rm rb}(\boldsymbol{\mu}) - u^{\rm fe}(\boldsymbol{\mu}) \in X^{\rm fe}$ as the difference between the reduced basis (RB) solution $u_N^{\rm rb}(\boldsymbol{\mu})$ and the highly accurate finite element solution $u^{\rm fe}(\boldsymbol{\mu})$. The residual $r(v;\boldsymbol{\mu}) \in (X^{\rm fe})'$ is given as:

$$r(v;\boldsymbol{\mu}) := f(v;\boldsymbol{\mu}) - a(u_N^{\rm rb}, v;\boldsymbol{\mu}), \quad \forall v \in X^{\rm fe}, \tag{2.7}$$

and its dual norm as:

$$\|r(\cdot;\boldsymbol{\mu})\|_{(X^{\rm fe})'} := \sup_{v\in X^{\rm fe}} \frac{r(v;\boldsymbol{\mu})}{\|v\|_{X^{\rm fe}}}. \tag{2.8}$$

We define the relative error estimator for the output as:

$$\eta(\boldsymbol{\mu}, W_N) := \frac{\|r(\cdot;\boldsymbol{\mu})\|_{(X^{\rm fe})'}\|\ell^{\rm fe}(\cdot;\boldsymbol{\mu})\|_{(X^{\rm fe})'}}{\beta^{\rm fe}(\boldsymbol{\mu})|s_N^{\rm rb}(\boldsymbol{\mu})|}. \tag{2.9}$$

Other types of error estimators can also be used, see *e.g.*, [25].

To build the parameter set S_N, the corresponding basis set W_N and the reduced basis space $X_N^{\rm rb}$, a greedy algorithm is used. Introducing a train set $\Xi_{\rm train} \subset \mathcal{D}$, consisting of a finely sampled subset of $\mathcal{D}$ of finite cardinality, we first pick a $\boldsymbol{\mu}^1 \in \Xi_{\rm train}$, and compute the corresponding basis function $u^{\rm fe}(\boldsymbol{\mu}^1)$. Let $S_1 = \{\boldsymbol{\mu}^1\}$, $W_1 = \{u^{\rm fe}(\boldsymbol{\mu}^1)\}$, and $X_1^{\rm rb} = {\rm span}\{u^{\rm fe}(\boldsymbol{\mu}^1)\}$. Now, suppose that we already have N points in $\Xi_{\rm train}$ to form S_N, the corresponding W_N and $X_N^{\rm rb}$, for some integer $N \geq 1$. Then, choose

$$\boldsymbol{\mu}^{N+1} := {\rm argmax}_{\boldsymbol{\mu}\in\Xi_{\rm train}} \eta(\boldsymbol{\mu}; W_N), \tag{2.10}$$

to select 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 $X_{N+1}^{\rm rb}$. The above procedure is repeated until N is large enough that $\max_{\boldsymbol{\mu}\in\Xi_{\rm train}} \eta(\boldsymbol{\mu}; W_N)$ is bounded by a prescribed tolerance.

For this approach to be accurate, it is essential that the training set is sufficiently fine, *i.e.*, for problems with many parameters, the size of the train set $\Xi_{\rm train}$ becomes very large. Even with a rapid approach for evaluating $\eta(\boldsymbol{\mu}; W_N)$ for all $\boldsymbol{\mu} \in \Xi_{\rm train}$ the cost of this quickly becomes a bottleneck in the construction of the reduced basis.

2.2. Empirical interpolation method

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

$$a(u,v;\boldsymbol{\mu}) = \sum_{i=1}^{Q_a} \Theta_i^a(\boldsymbol{\mu}) a_i(u,v), \quad f(v;\boldsymbol{\mu}) = \sum_{i=1}^{Q_f} \Theta_i^f(\boldsymbol{\mu}) f_i(v), \quad \text{and} \quad \ell(v;\boldsymbol{\mu}) = \sum_{i=1}^{Q_\ell} \Theta_i^\ell(\boldsymbol{\mu}) \ell_i(v), \tag{2.11}$$

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 $X_N^{\rm rb}$ with N basis functions, we can apply an offline/online strategy. In the offline step, one precomputes matrices and vectors related to forms a_i, f_i, and ℓ_i, for $i = 1, \ldots, Q_a, Q_f, Q_\ell$. The cost of this may be substantial but is done only once. In the online step, we now construct matrices and vectors in the reduced basis formulation (2.4), solve the resulting reduced basis problem, and evaluate the output functional (2.5). The amount of work of the online step is independent of the degrees of freedom of $X^{\rm fe}$, 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 of $\eta(\boldsymbol{\mu}; W_N)$ includes the solution procedure of the reduced basis problem, the evaluation of the residual (and output functional) in the dual norm, and a possible solution of a linear program to provide a lower bound of $\beta^{\rm fe}(\boldsymbol{\mu})$, see [2]. In all cases, the work does not depend on the size of $X^{\rm fe}$, but only on N and can, hence, be accomplished at low cost.

However, when the parameter dependent problem does not satisfy the affine assumption (2.11), this key benefit is lost. To circumvent this, the empirical interpolation method (EIM) [2, 12, 13] has been proposed

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

to enable the treatment of the non-affine operators and approximate them in the form (2.11) 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 [2,13,19] and serves to provide parameter values $S_N = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N\}$ such that the interpolant

$$\mathcal{I}_N(\mathcal{F})(\mathbf{x}; \boldsymbol{\mu}) := \sum_{j=1}^{N} \beta_j(\boldsymbol{\mu}) q_j(\mathbf{x}) \tag{2.12}$$

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

The sample points S_N are chosen by a greedy algorithm as discussed in the following. Again, using a train set $\Xi_{\text{train}} \subset \mathcal{D}$, comprising a fine sampling of $\mathcal{D}$ of finite cardinality, we first pick $\boldsymbol{\mu}^1 \in \Xi_{\text{train}}$, 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 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

$$q_i(\mathbf{x}^i) = 1, \qquad \text{and} \qquad q_j(\mathbf{x}^i) = 0 \qquad \text{for all } j > i, \tag{2.13}$$

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

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

The approximation of level N of $\mathcal{F}(\cdot; \boldsymbol{\mu})$ is given by the interpolant defined by (2.12). Set

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

and choose

$$\boldsymbol{\mu}^{N+1} := \text{argmax}_{\boldsymbol{\mu}\in\Xi_{\text{train}}} \eta(\boldsymbol{\mu}; W_N), \tag{2.14}$$

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:

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

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, (2.13) is satisfied since the interpolation is exact for all previous sample points in S_N. The algorithm is terminated once the error $\max_{\boldsymbol{\mu}\in\Xi_{\text{train}}} \eta(\boldsymbol{\mu}; W_N)$ is below a prescribed tolerance. One observes that the EIM also uses a greedy algorithm to choose the sample points with only slight differences to the approach of the reduced basis method. Hence, in the case of a high dimensional parameter space, the computational cost of constructing the empirical interpolation can likewise 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 cost of the greedy approach in such cases.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

3.1. A typical greedy algorithm

First, we introduce the following abstract framework to generalize the greedy algorithm. 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

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

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

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

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 scheme represents a typical greedy algorithm.

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

1: **Initialization**: Choose an initial parameter value $\boldsymbol{\mu}^1 \in \Xi_{\text{train}}$, set $S_1 = \{\boldsymbol{\mu}^1\}$, compute $v(\boldsymbol{\mu}^1)$, set $W_1 = \{v(\boldsymbol{\mu}^1)\}$, and $N = 1$;
2: **while** $\max_{\mu \in \Xi} \eta(\mu; W_N) > tol$ **do**
3: For all $\boldsymbol{\mu} \in \Xi_{\text{train}}$, compute $\eta(\boldsymbol{\mu}; W_N)$;
4: Choose $\boldsymbol{\mu}^{N+1} = \text{argmax}_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \eta(\boldsymbol{\mu}; W_N)$;
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 as a result of the above greedy algorithm, we ensure that the sets S_N and W_N are hierarchical:

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

Remark 3.1. The above algorithm is an abstraction of the greedy algorithm. For particular methods, different necessary modules, inputs, and outputs, should be added, *e.g.*, the selection of interpolation points in EIM, and the orthogonalization process in RBM.

Now, we discuss the robustness of the standard greedy algorithm.

Definition 3.2. The basis function set W is tol-robust for a sampling set Ξ of $\mathcal{D}$ if for all $\boldsymbol{\mu} \in \Xi$,

$$\eta(\boldsymbol{\mu}, W_N) \le tol.$$

Clearly, W_N generated by Algorithm 1 is tol-robust for Ξ_{train}.

3.2. An improved greedy algorithm based on a saturation assumption

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

Definition 3.3 (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

$$\eta(\boldsymbol{\mu}; W_M) \le C_{\text{sa}}\, \eta(\boldsymbol{\mu}; W_N) \quad \text{for some} \quad C_{\text{sa}} > 0 \quad \text{for all} \quad M > N > 0 \tag{3.15}$$

holds, we say that the *saturation assumption* is satisfied.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

In special cases, this assumption can be proved with an exact $C_{\rm sa}$, otherwise, it remains as an assumption. In the following $C_{\rm sa}$ is a free parameter that can be chosen in the algorithm rather than being a sound theoretical upper bound constant.

Remark 3.4. When $C_{\rm sa} = 1$, it implies that $\eta(\boldsymbol{\mu}; W_N)$ is not increasing for a fixed $\boldsymbol{\mu}$ and increasing N. Choose $C_{\rm sa} < 1$ is a more aggressive assumption, ensuring that $\eta(\boldsymbol{\mu}; W_N)$ is strictly decreasing. This assumption with $C_{\rm sa} < 1$ is very common in the adaptive finite element method community, see [1]. The assumption $C_{\rm sa} > 1$ is a more relaxed assumption, allowing that $\eta(\boldsymbol{\mu}; W_N)$ might not be monotonically decreasing, but can be 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 intermitted non-monotonic behavior as N is increasing, overall it is decreasing.

Utilizing this assumption, we can design an improved greedy algorithm. First, for each parameter value $\boldsymbol{\mu} \in \Xi_{\rm train}$, we create an error value profile $\eta_{\rm saved}(\boldsymbol{\mu})$ that can be accessed at any time and that will be updated when more accurate data is available. Initially, we can set $\eta_{\rm saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}; W_0) = \infty$. Now suppose that S_N and W_N are determined and that we aim to find the next sample point $\boldsymbol{\mu}^{N+1} = \operatorname{argmax}_{\boldsymbol{\mu}\in\Xi_{\rm train}} \eta(\boldsymbol{\mu}; W_N)$. When $\boldsymbol{\mu}$ runs through the train set $\Xi_{\rm train}$, we update a temporary maximum (over $\Xi_{\rm train}$), until we have searched the whole set $\Xi_{\rm train}$. In this loop, we require that, for each $\boldsymbol{\mu}$, $\eta_{\rm saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}; W_L)$ for some $L < N$. Now, since the saturation assumption ensures $\eta(\boldsymbol{\mu}; W_N) \le C_{\rm sa}\, \eta(\boldsymbol{\mu}; W_L)$ for $L < N$ and provided $C_{\rm sa}\, \eta_{\rm saved}(\boldsymbol{\mu})$ is less than the current temporary maximum, $\eta(\boldsymbol{\mu}, W_N)$ cannot be greater than the current temporary maximum. Hence, we may skip the computation of $\eta(\boldsymbol{\mu}, W_N)$, and leave $\eta_{\rm saved}(\boldsymbol{\mu})$ untouched. On the other hand, if $C_{\rm sa}\, \eta_{\rm saved}(\boldsymbol{\mu})$ exceeds the current temporary maximum, it is potentially the maximizer. And, we compute $\eta(\boldsymbol{\mu}, W_N)$, update $\eta_{\rm saved}(\boldsymbol{\mu})$, and compare it with the current maximum to see if the current temporary maximum must be updated. We notice that if we proceed in this manner, the requirement that for each $\boldsymbol{\mu}$, $\eta_{\rm saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}; W_L)$ for some $L < N$ holds.

The saturation assumption based algorithm is given in the following Algorithm 2.

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

Choose an initial parameter value $\boldsymbol{\mu}^1 \in \Xi_{\rm train}$, set $S_1 = \{\boldsymbol{\mu}^1\}$; compute $v(\boldsymbol{\mu}^1)$, set $W_1 = \{v(\boldsymbol{\mu}^1)\}$, and $N = 1$;
Set a vector $\eta_{\rm saved}$ with $\eta_{\rm saved}(\boldsymbol{\mu}) = \infty$ for all $\boldsymbol{\mu} \in \Xi_{\rm train}$;
while $\max_{\boldsymbol{\mu}\in\Xi_{\rm train}} \eta_{\rm saved}(\boldsymbol{\mu}) \ge tol$ **do**
  $error_{tmpmax} = 0$;
  for all $\boldsymbol{\mu} \in \Xi_{\rm train}$ **do**
    if $C_{sa}\eta_{\rm saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
      Compute $\eta(\boldsymbol{\mu}; W_N)$, and let $\eta_{\rm saved}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}, W_N)$;
      if $\eta_{\rm saved}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
        $error_{tmpmax} = \eta_{\rm saved}(\boldsymbol{\mu})$, and let $\boldsymbol{\mu}_{max} = \boldsymbol{\mu}$;
      end if
    end if
  end for
  Choose $\boldsymbol{\mu}^{N+1} = \boldsymbol{\mu}_{max}$, set $S_{N+1} = S_N \cup \{\boldsymbol{\mu}^{N+1}\}$;
  Compute $v(\boldsymbol{\mu}^{N+1})$, set $W_{N+1} = W_N \cup \{v(\boldsymbol{\mu}^{N+1})\}$;
  $N \leftarrow N + 1$;
end while

Algorithm 2: A greedy algorithm based on a saturation assumption.

Further we discuss several aspects on the proposed algorithm in the following remarks:

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

(2) Due to round-off errors, the error estimator may stagnate even if we add more basis functions, or the greedy algorithm will select some point already in S_N. In this case, the greedy algorithm should be terminated.

(3) The choice of the saturation assumption constant C_{sa} depends on the application of the greedy algorithm. In some particular cases, the constant can be proven to be one as in the case of the successive constraint method for the lower bounds of inf-sup constants, see [7,30]. For the general case, if the underlying method is projection based, $C_{\text{sa}} = 1$ is a reasonably good choice. If the underlying method is interpolation based, a larger C_{sa} should be used. We will discuss the choice of C_{sa} in more detail for the RBM and EIM in the upcoming sections.

It is also worth mentioning that in the early pre-asymptotic stage, we might not have to pick the parameter corresponding to the worst approximation error, since a parameter value with a bad but not worst approximation error is likewise a good choice. If a certain parameter value is truly critical, it will be chosen later. This indicates that we may not need to choose a very conservative C_{sa}.

In general, if $C_{\text{sa}} \geq 1$ the saturation assumption does not need to strictly hold in order to ensure that the algorithm works. It is more a guide to decide whether it is worth checking the accuracy for a certain parameter value or not.

(4) If the ratio between two levels are known or can be estimated, we can modify or refine the saturation assumption to be

$$\eta(\boldsymbol{\mu}; W_{N+1}) \leq C_{\text{sa},N}\, \eta(\boldsymbol{\mu}; W_N) \quad \text{for some} \quad C_{\text{sa},N} > 0 \quad \text{for all} \quad N > 0 \tag{3.16}$$

The C_{sa} in (3.15) will be M and N dependent, and is multiplications of $C_{\text{sa},i}$, $N \leq i < M$.

We now discuss the robustness of Algorithm 2. The following theorem holds.

Theorem 3.5. *If the saturation assumption is satisfied, the basis function set W_N generated by Algorithm 2 is tol-robust for Ξ_{train}.*

Proof. The algorithm ensures $\eta_{\text{saved}}(\boldsymbol{\mu}) < tol$ for all $\boldsymbol{\mu} \in \Xi_{\text{train}}$. For all $\boldsymbol{\mu}$ that $C_{\text{sa}}\eta_{\text{saved}}(\boldsymbol{\mu}) > error_{tmpmax}$ in the last searching for new basis function, $\eta_{\text{saved}}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}, W_N)$. For other points, $C_{\text{sa}}\eta_{\text{saved}}(\boldsymbol{\mu}) < tol$. By the saturation assumption, $\eta_{\boldsymbol{\mu},W_N} \leq C_{\text{sa}}\eta_{\text{saved}}(\boldsymbol{\mu}) < tol$. Thus the theorem. □

For the case that the saturation assumption is only a rudimentary guess, we do face the possibility that some points will have an error bigger than *tol*. But even in this case, if we assume the underlying method is convergent, that more basis functions will lead to a smaller error in general, the saturation assumption is valid in general. If the user is more cautious, it is easy to add a final validation step to test all points by the basis function set to ensure the basis function set is *tol*-robust.

3.3. An adaptively enriching greedy algorithm (AEGA)

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 the prescribed tolerance. Thus, for high dimensional parametrized problems, we will likely have to face the problem that the train set is not rich enough, and it is in practice hard to construct a fine enough train set. Observe that it is almost impossible to construct a tensor product based train set for $p > 10$. Even if we only put 5 points for each parameter, a large train set of $5^{11} = 48\,828\,125$ points results. For a larger p, the curse of dimensionality is inevitable.

To address this problem, we first build the set of basis functions 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,

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

we add a "safety check" step at the end, *i.e.*, we test the approximation by a larger number of test parameters, to ensure that the resulting errors are less than the prescribed tolerance on this refined parameter set too. If, for all validation points, the estimated errors are less than the tolerance, we 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".

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 parametrized problem. Fortunately, we can adaptively change the train set by removing useless points and enriching by adding 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. The gain of retaining these points in Ξ_N is very small, since for a problem with a monotonically decreasing error, these points will never be selected, and even for a non-monotonically decreasing error problem, these points have little chance to be selected. Thus we can remove these points from the train set. In the latter case, in principle, we should re-test these removed points to ensure their errors are less the tolerance with the final set of basis functions. But since these points are randomly generated, perhaps they are no better than other random validation points, we may just use new random points in to validate the basis set. If we want to be more cautious, we can add these points back to the "safety check"/validation points, to ensure the basis set is good enough for these early discarded points. If we have a pre-scribed validation set, we can expand the set by adding these early discarded points. With these points discarded, 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 work, thus combining the two ideas.

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

Remark 3.6. 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 and go along with the spirit of [5, 6].

We now discuss the robustness of Algorithm 3. The following theorem is obvious.

Theorem 3.7. *The basis function set W_N generated by Algorithm 3 is tol-robust for N_{sc} validation points.*

As discussed before, if the user has a specific validation point set, or the user is cautious about those points discarded in the earlier stage, it is easy to make the output *tol*-robust for these points by modifing line 23 or 28 from generating random points to form a new Ξ_{train} by these user-specific points.

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).

4.1. Saturation assumption

First, we present some very pessimistic estimation of the saturation assumption constant C_{sa} by some *a priori* results in EIM.

Lemma 4.1. *[13] Let Y be a Banach space of continuous functions defined on part of $\mathbb{R}^d$. For any $\mathcal{F} \in Y$, the interpolation error satisfies*

$$\|\mathcal{F} - \mathcal{I}_N(\mathcal{F})\|_{L^\infty(\Omega)} \leq (1 + \Lambda_N) \inf_{v_N \in W_N} \|\mathcal{F} - v_N\|_{L^\infty(\Omega)} \tag{4.17}$$

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

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 $\boldsymbol{\mu}^1 \in \Xi_{\text{train}}$ and set $S_1 = \{\boldsymbol{\mu}^1\}$ and $N = 1$;
Set a vector η_{saved} with $\eta_{\text{saved}}(\boldsymbol{\mu}) = \infty$ for all $\boldsymbol{\mu} \in \Xi_{\text{train}}$;
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 $\boldsymbol{\mu} \in \Xi_{\text{train}}$ **do**
    if $C_{\text{sa}}\eta_{\text{saved}}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
      Compute $\eta(\boldsymbol{\mu}; W_N)$, and let $\eta_{\text{saved}}(\boldsymbol{\mu}) = \eta(\boldsymbol{\mu}, W_N)$;
      if $\eta_{\text{saved}}(\boldsymbol{\mu}) > error_{tmpmax}$ **then**
        $error_{tmpmax} = \eta_{\text{saved}}(\boldsymbol{\mu})$, and let $\boldsymbol{\mu}_{max} = \boldsymbol{\mu}$;
      end if
      if $\eta_{\text{saved}}(\boldsymbol{\mu}) < tol$ **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 $M - \text{sizeof}(\Xi_{\text{train}})$ new samples, add them into Ξ_{train} such that $\text{sizeof}(\Xi_{\text{train}}) = M$; 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 equal to ∞ for all $\boldsymbol{\mu} \in \Xi_{\text{train}}$;
  end if
end while

Algorithm 3: An adaptively enriching greedy algorithm (AEGA).

An upper bound of the Lebesgue constant Λ_N is $2^N - 1$. In the asymptotic region, we have the conservative estimate

$$\|\mathcal{F} - \mathcal{I}_{N+1}(\mathcal{F})\|_{L^\infty(\Omega)} \approx 2\,\|\mathcal{F} - \mathcal{I}_N(\mathcal{F})\|_{L^\infty(\Omega)}.$$

Thus, a conservative bound of $C_{\text{sa},N}$ in (3.16) is 2, and C_{sa} is 2^{M-N} as discussed in remark [4] in Section 3.2.

Of course, as is mentioned in [13], this estimation is the worst case scenario and not observed in practice. After all, the saturation assumption is a guideline to improve the computational efficiency and does not interfere with the final accuracy of the algorithm.

We test the saturation assumption based algorithm for two test problems with low dimensional parameter spaces. The generalization to complex-valued (instead of real-valued) functions is straightforward.

Test 4.1. Consider the complex-valued function

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

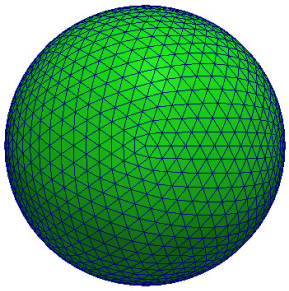


FIGURE 1. Discrete surface for the unit sphere.

where $x \in \Omega = (0, 2]$ and $k \in \mathcal{D} = [1, 25]$. The interval Ω is divided into an equidistant point set with 15 000 points to build Ω_h where the error $\|\mathcal{F}_1(x;\mu) - \mathcal{I}_N(\mathcal{F}_1)(x;\mu)\|_{L^\infty(\Omega_h)}$ is computed. For the standard greedy algorithm, the train set Ξ_{train} consists of 5000 equidistant points in $\mathcal{D}$.

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

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

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 $\boldsymbol{\mu} = (k, \theta, \varphi) \in \mathcal{D} = [1, 5] \times [0, \pi] \times [0, 2\pi]$. In computational 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 [12]. As domain Ω we take a unit sphere. For practical purposes, 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 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 Figure 2 we show the evolution of the average and the standard deviation of

$$C(N, n) = \frac{\eta(\boldsymbol{\mu}, W_N)}{\eta(\boldsymbol{\mu}, W_{N-n})}, \qquad n = 1, 2, 3, \tag{4.18}$$

over Ξ_{train} along with the standard greedy sampling process Algorithm 1. This quantity is an indication of $C_{\text{sa},N}$. We observe that $C_{\text{sa},N} \leq 2$, in agreement with the *a priori* result. But we also observe that choosing $C_{\text{sa}} = 2^{M-N}$ is too pessimistic as anticipated. $C_{\text{sa}} = 2$ seems to be a reasonable choice. 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 indicates, that at each step N, the percentage of work that needs to be done by using the saturation assumption compared to that using the standard greedy algorithm, thus comparing the different workloads, at each loop over Ξ_{train} of Algorithms 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.

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.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

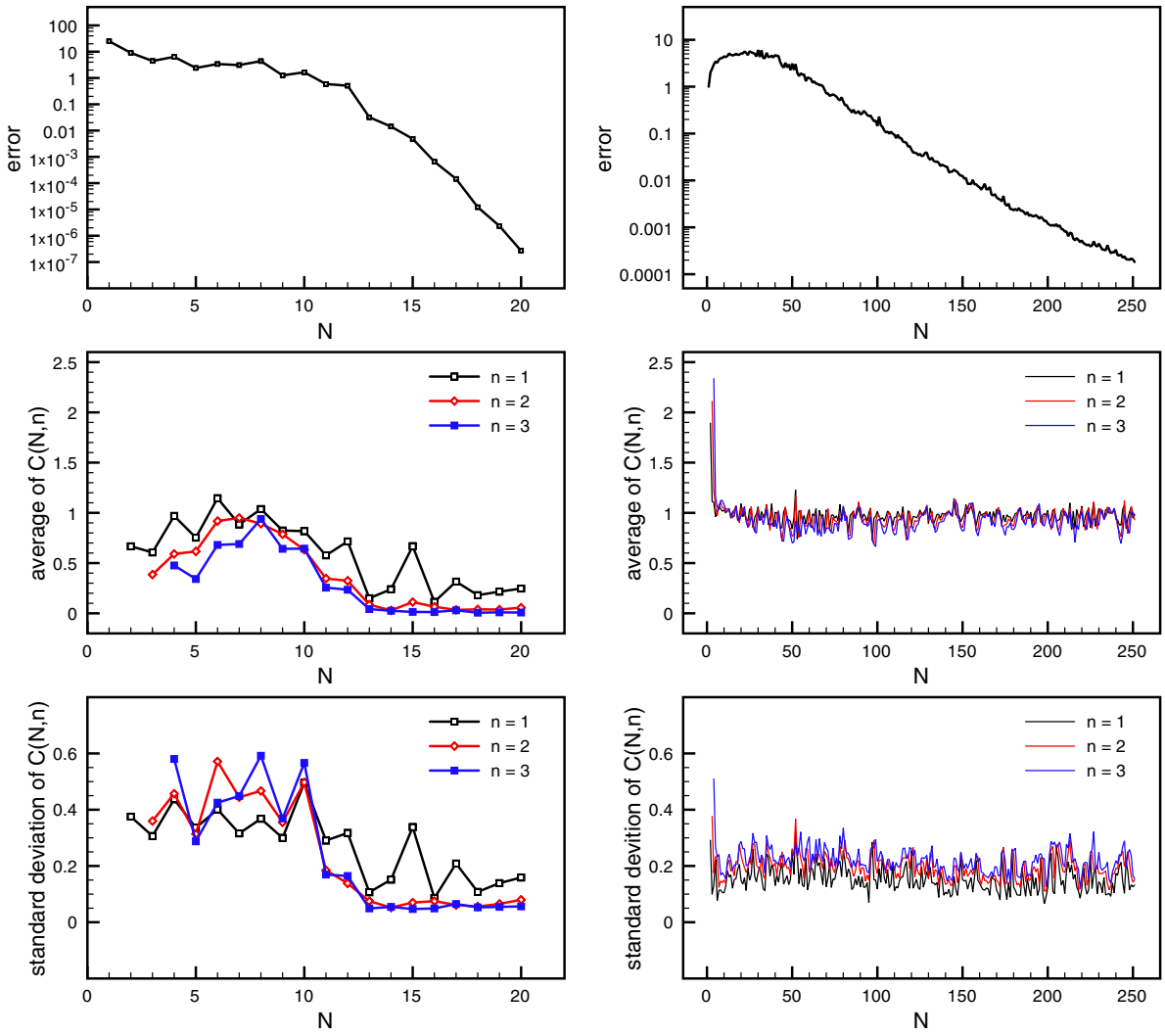


FIGURE 2. Evolution of the average and standard deviation of $C(N,n)$ over Ξ_{train} along a standard greedy sampling for $\mathcal{F}_1$ (left column) and $\mathcal{F}_2$ (right column). For illustration purposes, the accuracy with respect to N is also provided.

Considering $\mathcal{F}_2$, we set $M = 1000, 5000, 25\,000$, $N_{\text{sc}} = 125\,000$ and $C_{\text{sc}} = 2$. 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 and only for academic/illustrative purpose) 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, *i.e.*, the number of samples in the randomly sampled train set, the convergence error provided by the AEGA and the error of the AEGA over the larger control set become (not surprisingly) closer and converge to the error provided by the standard EIM using training set Ξ_{train}.

Figure 5 shows the evolution of the number of points which were part of the train set during the AEGA 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.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

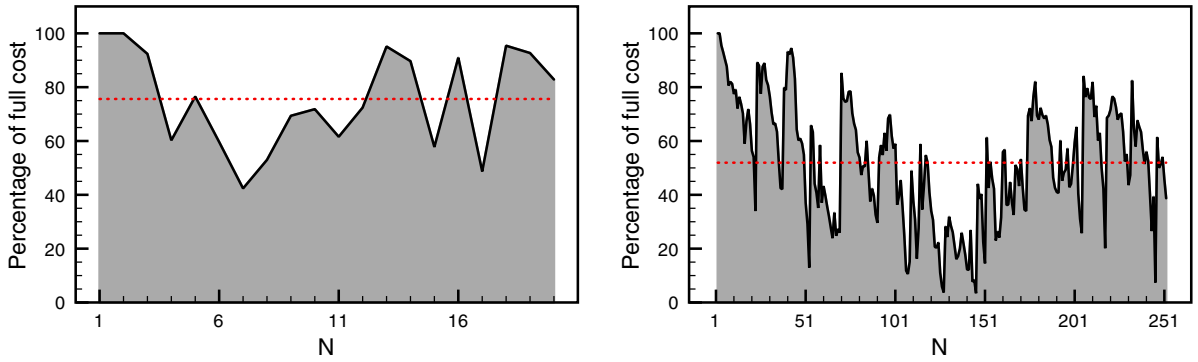


FIGURE 3. Percentage of work at each step N, using the saturation assumption based greedy algorithm, compared to the workload using the standard greedy algorithm, for $\mathcal{F}_1$ (left) and $\mathcal{F}_2$ (right) for the choice of $C_{\text{sa}} = 2$.

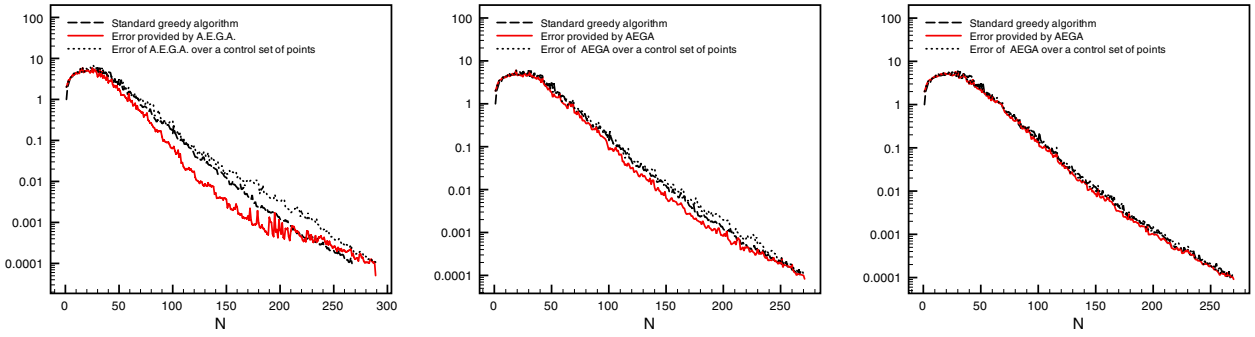


FIGURE 4. Convergence behavior of the adaptively enriching greedy algorithm for $\mathcal{F}_2$ with $M = 1000$ (left), $M = 5000$ (middle) and $M = 25\,000$ (right).

In Figure 6, we present, for all three 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 observes that at the end, almost all points in the train set are withdrawn (and thus the corresponding errors need to be computed). However, during a long time, the algorithm works with the initial train set until the error tolerance is satisfied on those points before they are withdrawn. As a 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 the saturation assumption for each new sample point in the train set cannot be utilized.

Remark 4.2. Algorithm 3 is subject to randomness. However, we plot only one realization of the adaptively enriching greedy algorithm. We observe in the numerical tests that different realizations do not differ from each other significantly. This can be explained by the fact that the algorithm depends on a large enough number of realizations of random points. Therefore these illustrations are still meaningful.

Test 4.3. Introduce the following real-valued function

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

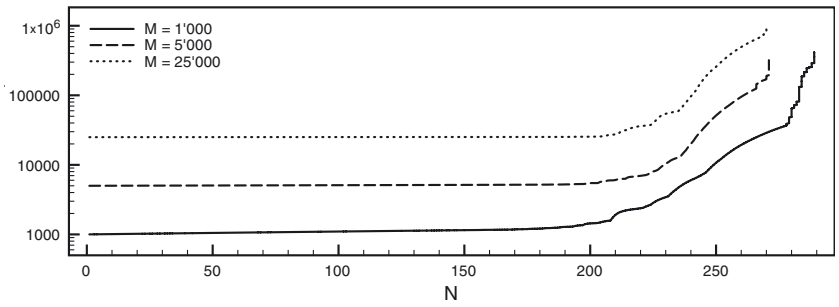


FIGURE 5. Evolution of the number of points where the accuracy is checked of the adaptively enriching greedy algorithm for different values of M (Safety check not included).

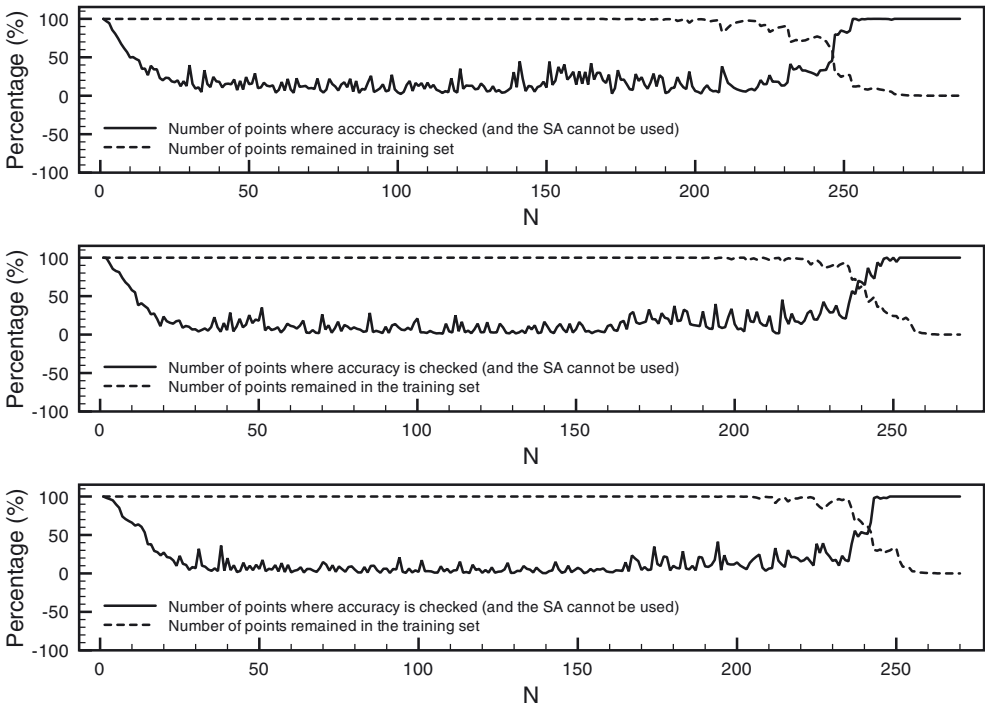


FIGURE 6. Percentage of work (effected at each step N) w.r.t. the total number of points M and of the number of points remained in the train set (at each step N) of the adaptively enriching greedy algorithm for $\mathcal{F}_2$ and different values of $M = 1000, 5000, 25\,000$ (from top to bottom).

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 Figures 7 and 8 we plot the convergence behaviour of the adaptive enriching greedy algorithm for $\mathcal{F}_3$ and $tol = 10^{-4}$. The value of N_{sc} for all different choices of $M = 100$, $M = 1000$ and $M = 10\,000$ is set equal to $N_{\text{sc}} = 100\,000$. Figure 8 is a zoom of Figure 7 towards the end of the convergence to highlight the action of the safety check. We observe that in the case of $M = 10\,000$ the safety check is passed after only a few attempts whereas for $M = 100$ the safety check fails 34 times until finally successful. This means that during

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

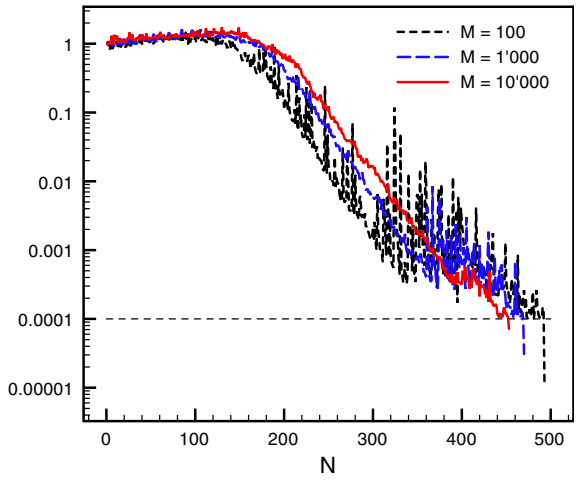


FIGURE 7. Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_3$.

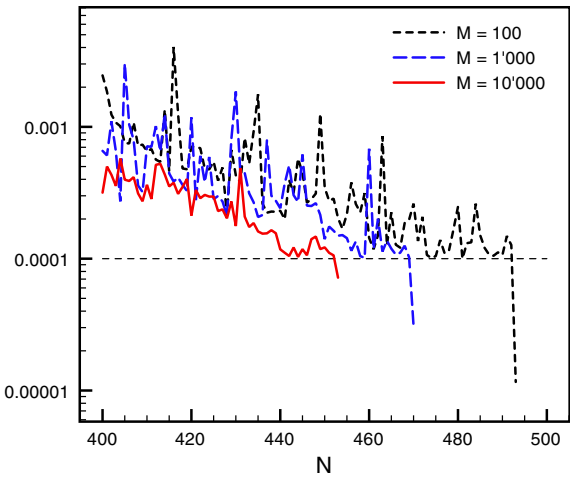


FIGURE 8. Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_3$ zoomed in towards the end where the safety check is active.

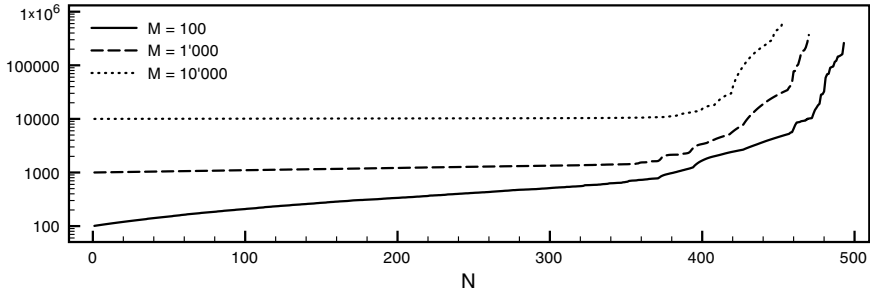


FIGURE 9. Evolution of the number of points where the accuracy is checked of the adaptively enriching greedy algorithm for different values of M (Safety check included).

each safety check there is 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 is below the prescribed tolerance.

In Figure 9, the accumulated number of points belonging to the train set Ξ_{train} 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

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

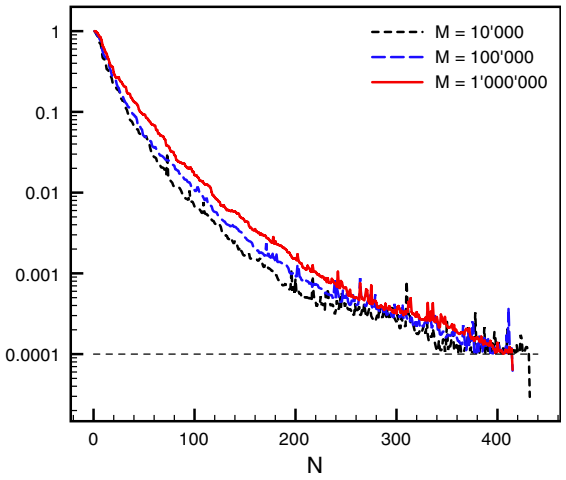


FIGURE 10. Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_4$.

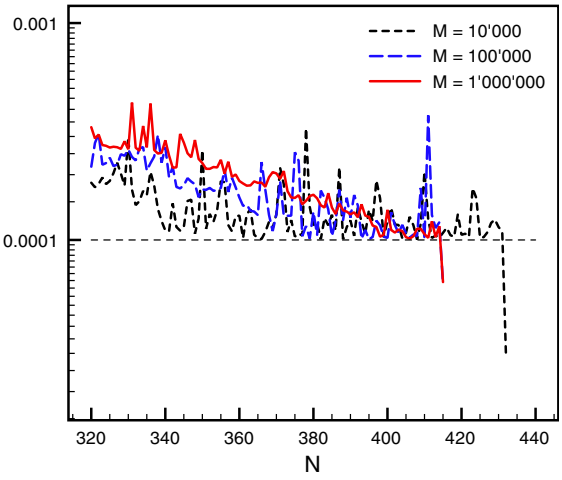


FIGURE 11. Convergence behavior of the adaptive enriching greedy algorithm for $\mathcal{F}_4$ zoomed in towards the end where the safety check is active.

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 Figure 10, the convergence behavior of the adaptively enriching greedy algorithm for the function $\mathcal{F}_4$ and $tol = 10^{-4}$ is plotted. The value of N_{sc} for all different choices of $M = 10\,000$, $M = 100\,000$ and $M = 1\,000\,000$ is set equal to $N_{\text{sc}} = 10\,000\,000$. Figure 11 is again a zoom of Figure 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 discuss that for some class of problems, the saturation assumption is satisfied with $C_{\text{sa}} = 1$ if the error is measured in the intrinsic energy norm. Then, we show that even for non-coercive problems, $C_{\text{sa}} = 1$ is the most reasonable choice. The distinction of error in some norms and its error estimators will also be discussed.

First, suppose that the variational problem is based on a minimization principle,

$$u = \text{argmin}_{v \in X} J(v), \tag{5.19}$$

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

$$u^{\text{fe}} = \text{argmin}_{v \in X^{\text{fe}}} J(v). \tag{5.20}$$

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

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

$$u_N^{\rm rb} = \text{argmin}_{v \in X_N^{\rm rb}} J(v). \tag{5.21}$$

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

$$J(u_N^{\rm rb}) - J(u^{\rm fe}). \tag{5.22}$$

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

$$J(u_{N+1}^{\rm rb}) - J(u^{\rm fe}) \leq J(u_N^{\rm rb}) - J(u^{\rm fe}). \tag{5.23}$$

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

On the other hand, if the problem is not based on a minimization principle, we will still have the following generalized Cea's Lemma, see *e.g.* [9]:

$$\|u^{\rm fe}(\boldsymbol{\mu}) - u_N^{\rm rb}(\boldsymbol{\mu})\|_{X^{\rm fe}} \leq C(\boldsymbol{\mu}) \inf_{v \in X_N^{\rm rb}} \|u^{\rm fe}(\boldsymbol{\mu}) - v\|_{X^{\rm fe}}. \tag{5.24}$$

With the modification reduced basis approach (2.6), we can ensure that the constant $C(\boldsymbol{\mu})$ is independent of N. For some cases, when the inf-sup constant $\beta^{\rm fe}(\mu)$ is very small, $C(\boldsymbol{\mu})$ can be very large.

Note, however, that we always discuss the saturation assumption for the same $\boldsymbol{\mu}$, thus, even though $C(\boldsymbol{\mu})$ may be very large, in the asymptotic region,

$$\|u^{\rm fe}(\boldsymbol{\mu}) - u_N^{\rm rb}(\boldsymbol{\mu})\|_{X^{\rm fe}} \approx C(\boldsymbol{\mu}) \inf_{v \in X_N^{\rm rb}} \|u^{\rm fe}(\boldsymbol{\mu}) - v\|_{X^{\rm fe}}. \tag{5.25}$$

Therefore, in the asymptotic region, we have

$$\begin{aligned} \|u^{\rm fe}(\boldsymbol{\mu}) - u_{N+1}^{\rm rb}(\boldsymbol{\mu})\|_{X^{\rm fe}} &\approx C(\boldsymbol{\mu}) \inf_{v \in X_{N+1}^{\rm rb}} \|u^{\rm fe}(\boldsymbol{\mu}) - v\|_{X^{\rm fe}} \\ &\leq C(\boldsymbol{\mu}) \inf_{v \in X_N^{\rm rb}} \|u^{\rm fe}(\boldsymbol{\mu}) - v\|_{X^{\rm fe}} \approx \|u^{\rm fe}(\boldsymbol{\mu}) - u_N^{\rm rb}(\boldsymbol{\mu})\|_{X^{\rm fe}}. \end{aligned}$$

Thus, $C_{\rm sa} = 1$ is a good choice even for the case where $C(\boldsymbol{\mu})$ is large ($\beta^{\rm fe}(\mu)$ is very small). For the pre-asymptotic stage, the above discussion still provides a good guideline, and it can be observed in practice that 1 is a robust choice.

We also emphasize that in real computations, instead of the exact error, we use an error estimator to guide the greedy algorithm (referred to as the *weak greedy algorithm*). Error estimators are upper bounds for the true error measured in some norm. The upper bounds may be quite pessimistic and thus $C_{\rm sa}$ with respect to the error estimator may be larger than 1. But what we are really interested in are the true errors, so that $C_{\rm sa} = 1$ is always reasonable.

The above discussion is for the case where the saturation assumption based greedy algorithm is used alone. For the adaptively enriching greedy algorithm, the problem is less serious, and we can always use $C_{\rm sa} = 1$, see the discussion in Section 5.2 later.

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

$$\begin{cases} -\nabla \cdot (\alpha(\boldsymbol{\mu}) \nabla u) = f \text{ in } \Omega, \\ u = 0 \text{ on } \Gamma_D, \\ \alpha(\boldsymbol{\mu}) \nabla u \cdot \mathbf{n} = g \text{ on } \Gamma_N, \end{cases} \tag{5.26}$$

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

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

$$a(u, v; \boldsymbol{\mu}) = f(v), \quad \forall v \in X, \tag{5.27}$$

where

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

The energy functional is

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

and we have

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

In the last step, Galerkin orthogonality

$$\int_\Omega \alpha(\boldsymbol{\mu}) \nabla u^{\mathrm{fe}} \cdot \nabla w \mathrm{d}x = f(w) \quad \forall\, w \in X^{\mathrm{fe}}$$

is used. The above analysis is standard, see [23].

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 $C_{\mathrm{sa}} = 1$,

$$\|\alpha(\boldsymbol{\mu})^{1/2} \nabla (u^{\mathrm{rb}}_M - u^{\mathrm{fe}})\|_{0,\Omega} \leq \|\alpha(\boldsymbol{\mu})^{1/2} \nabla (u^{\mathrm{rb}}_N - u^{\mathrm{fe}})\|_{0,\Omega} \quad \text{for } 0 < N < M. \tag{5.28}$$

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.

For (5.26), we choose the underlying norm of X and X^{fe} to be the H^1-semi-norm $\|v\|_X = \sqrt{\int_\Omega (\nabla v)^2 \mathrm{d}x}$. Notice that, due to the need to compute the dual norm of the X^{fe}-norm, 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 (5.26) is defined as:

$$\eta(\boldsymbol{\mu}; W_N) = \frac{\|f\|_{X'} \|r(\cdot; \boldsymbol{\mu})\|_{X'}}{\beta(\boldsymbol{\mu}) |f(u^{\mathrm{rb}}_N(\boldsymbol{\mu}))|}. \tag{5.29}$$

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 $\eta(\boldsymbol{\mu}; W_N)$ used here is in fact an estimate of the relative error measured by the square of the intrinsic energy norm. In principle, since we already proved that the error measured in the 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 (5.29) carefully, we find that for a fixed parameter $\boldsymbol{\mu}$, the values of $\|f\|_{X'}$ and $\beta(\boldsymbol{\mu})$ are fixed, and the change of the value of $|f(u^{\mathrm{rb}}_N(\boldsymbol{\mu}))|$ is quite small if the approximation $u^{\mathrm{rb}}_N(\boldsymbol{\mu})$ is already in the asymptotical regime. The only problematic term is $\|r(\cdot; \boldsymbol{\mu})\|_{X'}$.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

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

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

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

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

Test 5.1.

Consider the following thermal block problem, which is a special case of (5.26), see also [25]. Let $\Omega = (0,1)^2$, and

$$\begin{cases} -\nabla\cdot(\alpha(\boldsymbol{\mu})\nabla u) = 0 \text{ in } \Omega, \\ u = 0 \text{ on } \Gamma_{\rm top} = \{x \in (0,1), y = 1\}, \\ \alpha(\boldsymbol{\mu})\nabla u\cdot\mathbf{n} = 0 \text{ on } \Gamma_{\rm side} = \{x = 0 \text{ and } x = 1, y \in (0,1)\}, \\ \alpha(\boldsymbol{\mu})\nabla u\cdot\mathbf{n} = 1 \text{ on } \Gamma_{\rm base} = \{x \in (0,1), y = 0\}, \end{cases} \tag{5.30}$$

where $\alpha(\boldsymbol{\mu}) = 10^{2\boldsymbol{\mu}-1}$ 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 scalar parameter μ. The variational problem is given in (5.27). 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 define our finite approximation space

$$X^{\rm fe} = \{v \in C^0(\Omega) : v|_K \in P_1(K), \quad \forall K \in \mathcal{T}\}.$$

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

$$a(u^{\rm fe}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v) \quad v \in X^{\rm fe}. \tag{5.31}$$

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

$$a(u_N^{\rm rb}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v) \quad \forall v \in X_N^{\rm rb}. \tag{5.32}$$

We choose 101 equidistant sample points in $\mathcal{D} = [0, 1]$, *i.e.*, $\Xi_{\rm train} = \{\frac{i}{100}, i = 0, 1, 2, \ldots, 100\}$, and a standard reduced basis greedy algorithm with the error estimator defined in (5.29). 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 $X_N^{\rm rb}$ corresponding to S_N, $N = 1, \ldots, 5$. Then, for all points $\boldsymbol{\mu}$ in $\Xi_{\rm train}$, $\eta(\boldsymbol{\mu}; W_N)$, $N = 1, \ldots, 5$ is computed. Figure 12 shows the plots of η for each points in $\Xi_{\rm train}$ with number of reduced basis $= 1, \ldots, 5$. Clearly, we see that for each point $\boldsymbol{\mu} \in \Xi_{\rm train}$, $\eta(\boldsymbol{\mu}; W_5) \le \eta(\boldsymbol{\mu}; W_4) \le \eta(\boldsymbol{\mu}; W_3) \le \eta(\boldsymbol{\mu}; W_2) \le \eta(\boldsymbol{\mu}; W_1)$. For this problem, the saturation assumption is satisfied for $C_{\rm sa} = 1$ in the first several steps.

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.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

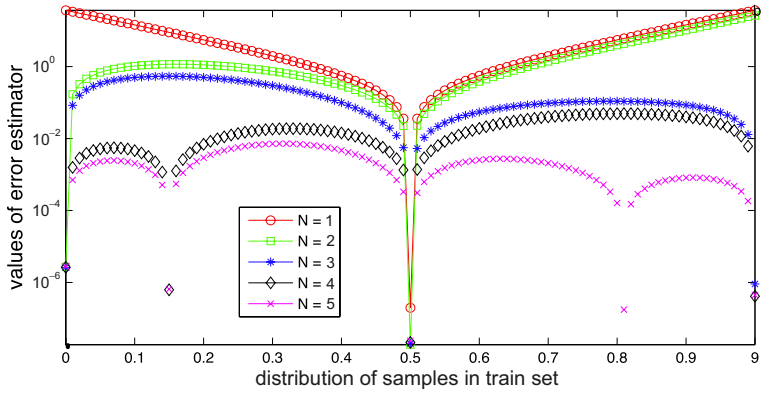


FIGURE 12. A verification of the saturation assumption for a symmetric positive definite problem with one parameter.

For (5.30), 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 becomes

$$a(u, v; \boldsymbol{\mu}) = \sum_{i=1}^{2} 100^{2\mu_i - 1} \int_{R_i} \nabla u \cdot \nabla v \mathrm{d}x + \int_{R_3} \nabla u \cdot \nabla v \mathrm{d}x. \tag{5.33}$$

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

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

We set the tolerance to be 10^{-3} and use the error estimator defined in (5.29).

Both the standard greedy algorithm and the saturation assumption based greedy algorithm requires 20 reduced basis functions to reduce the estimated error to less than the tolerance. If the error is measured in the intrinsic energy norm, we can choose $C_{\text{sa}} = 1$ 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 $C_{\text{sa}} = 1.1$, we get however the same set S_N as the standard greedy algorithm.

See Figure 13 for a comparisons of the workloads using the standard algorithm and the new approach based on the saturation assumption with $C_{\text{sa}} = 1$ and $C_{\text{sa}} = 1.1$, respectively. The mean percentage is computed as $\sum_{i=1}^{N}(\text{percentage at step } i)/N$. The mean percentages of $C_{\text{sa}} = 1$ and $C_{\text{sa}} = 1.1$ 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 Figure 14, we present the selections of the sample points S_N by the standard algorithm and the saturation assumption based greedy algorithm with $C_{\text{sa}} = 1$. Many sample points are identical.

Test 5.3. We continue and test a problem with 3 parameters. For (5.30), 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

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

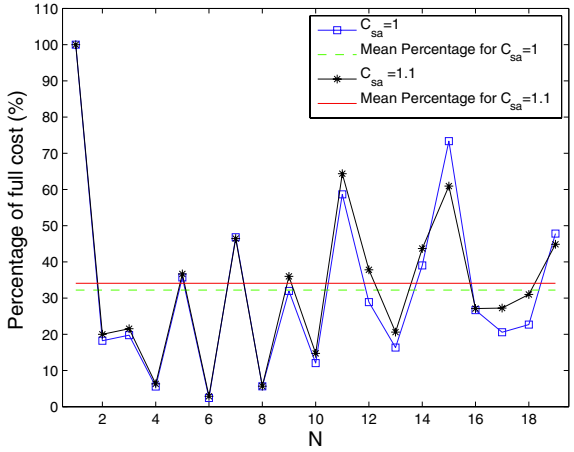


FIGURE 13. Percentage of work at each step N using the saturation assumption based greedy algorithm with $C_{\rm sa} = 1$, compared to the workload using the standard greedy algorithm for Test 5.2.

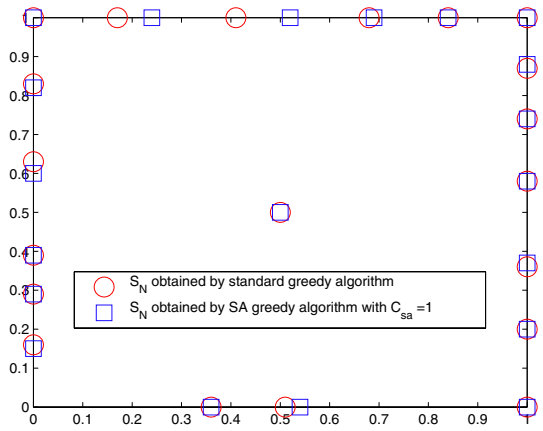


FIGURE 14. S_N obtained by the standard and the saturation assumption based greedy algorithms for Test 5.2.

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

$$a(u, v; \boldsymbol{\mu}) = \sum_{k=1}^{3} 100^{2\mu_k - 1} \int_{R_k} \nabla u \cdot \nabla v \mathrm{d}x + \int_{R_4} \nabla u \cdot \nabla v \mathrm{d}x. \tag{5.34}$$

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 $N_{50} = \{0, 1, 2, \ldots, 50\}$, the train set is given by:

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

The standard greedy algorithm needs 24 reduced basis functions to reduce the estimated error below the tolerance. If $C_{\rm sa}$ is chosen to be 1, 25 reduced basis functions are needed to reduce the estimated error below the tolerance. The set S_N obtained by the saturation assumption based algorithm with $C_{\rm sa} = 1$ is also different from the standard algorithm. As discussed before, this is mainly caused by the inaccuracy of the error estimator. If we choose $C_{\rm sa} = 3$, 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 $C_{\rm sa} = 1$ and $C_{\rm sa} = 3$, respectively. The mean percentages of workload for $C_{\rm sa} = 1$ and $C_{\rm sa} = 3$ are 21.6% and 33.7%, respectively.

Remark 5.1. For the type of compliant problem discussed in Tests 5.1, 5.2 and 5.3, other types of error estimator are suggested in [25]:

$$\eta^e(\boldsymbol{\mu}, W_N) := \frac{\|r(\cdot; \boldsymbol{\mu})\|_{(X^{\rm fe})'}}{\beta^{\rm fe}(\boldsymbol{\mu})^{1/2} \|u_N^{\rm rb}(\boldsymbol{\mu})\|_{X^{\rm fe}}} \quad \text{and} \quad \eta^s(\boldsymbol{\mu}, W_N) := \frac{\|r(\cdot; \boldsymbol{\mu})\|^2_{(X^{\rm fe})'}}{\beta^{\rm fe}(\boldsymbol{\mu}) |s_N^{\rm rb}(\boldsymbol{\mu})|}. \tag{5.35}$$

The most important term in the error estimator of the saturation assumption is the dual norm of the residual $\|r(\cdot; \boldsymbol{\mu})\|_{(X^{\rm fe})'}$. 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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

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. Numerical tests show that even if $C_{\rm sa} = 20$ 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.

Test 5.4 (A non-coercive example)**.** In this example, we show that for a non-coercive problem, the saturation assumption is satisfied numerically with $C_{\rm sa} = 1$. Let $\Omega = (0,1)^2$, and $-\Delta u - ku = 0 \in \Omega$ with same boundary conditions as (5.30). We choose $k = 1 + 1.4\mu$ and the parameter $\mu \in \mathcal{D} = [0,1]$. The finite element space $X^{\rm fe}$, the train set $\Xi_{\rm train}$ is chosen as in Test 5.1. We use the following error estimator

$$\eta(\mu, W_N) := \frac{\|r(\cdot;\mu)\|_{(X^{\rm fe})'}}{\beta^{\rm fe}_{lb}(\mu)}, \tag{5.36}$$

where $\beta^{\rm fe}_{lb}(\mu)$ is a lower bound of the inf-sup constant with respect to the corresponding norms. This lower bound is computed by a successive constraint method ([7–9, 18, 30]). A simple calculation shows that $k = 2.4674$ is an eigenvalue of the problem. Thus, when μ is close to 1, the constant $\beta^{\rm fe}(\mu)$ is small. The first parameter μ^1 is chosen to be 0, that is, $S_1 = \{0\}$. The standard greedy algorithm chooses the 2nd, 3rd, and 4th parameters as 1, 0.88, and 0.5, so $S_4 = \{0, 1, 0.88, 0.5\}$. as in Test 5.1, for all points $\mu \in \Xi_{\rm train}$, we compute $\eta(\mu, W_N)$, $N = 1, \ldots, 4$. Figure 17 shows the plots of η for each points with $N = 1, \ldots, 4$. For this non-coercive problem, the saturation assumption is clearly satisfied with $C_{\rm sa} = 1$ in the first few steps. This example agrees with our analysis that $C_{\rm sa}$ should be 1 even for non-coercive problems with an unfavorable stability constants.

5.2. Adaptively enriching greedy algorithm

Before presenting the numerical results, we discuss the choice of $C_{\rm sa}$ in the adaptively enriching greedy algorithm for reduced basis methods. As discussed earlier, in the asymptotic stage, $C_{\rm sa} = 1$ is a good choice. In the pre-asymptotic stage, the error may oscillate, and $C_{\rm sa} = 1$ may miss points with a relative large error. For the adaptively enriching greedy algorithm, this will be a lesser serious problem, since we will perform a "safety check" step using a large number of sample points. If some important parameters are missed in the earlier stage, it will eventually be captured by the algorithm, otherwise, it will not pass the "safety check". In addition, even if we miss the error computation for a particular parameter value, the error will be checked at a later stage before the algorithm terminated. Thus, for the adaptively enriching greedy algorithm for reduced basis methods, we can always use $C_{\rm sa} = 1$ confidently.

Test 5.5. We test the adaptively enriching greedy algorithm for the reduced basis method for a problem with 15 parameters.

For (5.30), 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

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

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

$$a(u, v; \boldsymbol{\mu}) = \sum_{k=1}^{15} 5^{2\mu_k - 1} \int_{R_k} \nabla u \cdot \nabla v \mathrm{d}x + \int_{R_{16}} \nabla u \cdot \nabla v \mathrm{d}x. \tag{5.37}$$

All other forms and spaces are identical to 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 $C_{\rm sa} = 1$, $tol = 0.05$, $N_{\rm sc} = 10\,000$. Since there is a "safety check" step to ensure the quality of the reduced basis functions, we do not worry that the choice of $C_{\rm sa}$ is too aggressive. We test three cases: $M = 100$, $M = 500$, and $M = 1\,000$. The convergence

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

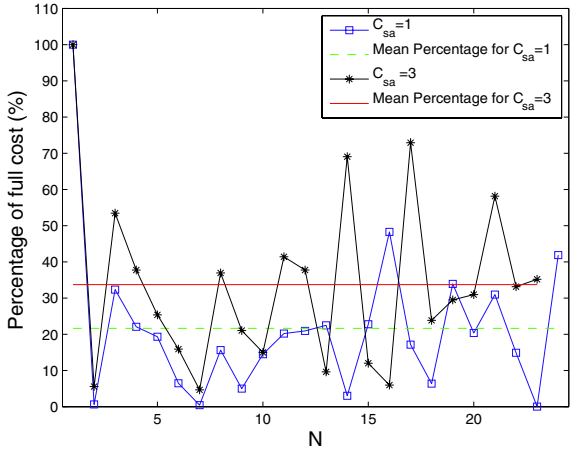


FIGURE 15. Percentage of work at each step N using the saturation assumption based greedy algorithm with $C_{\mathrm{sa}} = 1$ and $C_{\mathrm{sa}} = 3$, compared to the work load using the standard greedy algorithm for Test 5.3.

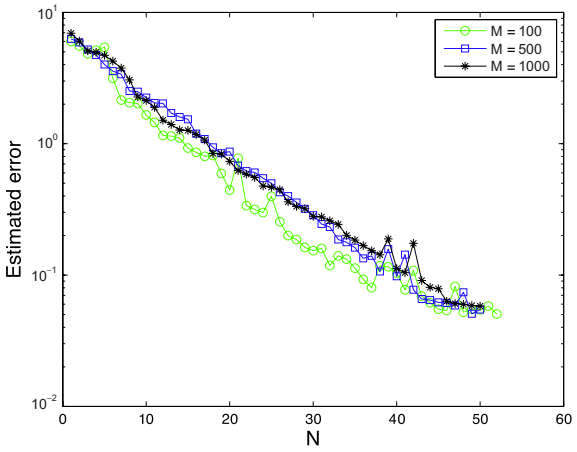


FIGURE 16. Convergence behavior of the adaptively enriching greedy algorithm for Test 5.4 with $M = 100$, 500, and 1000.

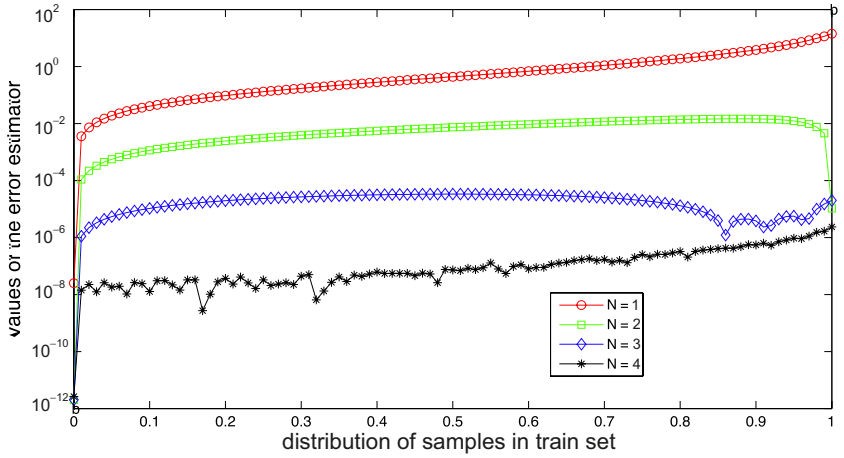


FIGURE 17. A verification of saturation assumption for a non-coercive problem with one parameter.

for one realization is plotted in Figure 16. The number of reduced basis for $M = 100$ is 52, and for the other two cases is 50. This is in agreement with the intuition that a bigger M will lead to a smaller number of the basis functions since better sample points can be chosen. However, once M is sufficiently large, we do not observe a further decrease of the number of required basis functions, similar to what can be observed in the case of the EIM in Section 4.

The percentage of work (effected at each step N) with respect to the total number of points M and 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 Figure 18. Initially, the estimated errors are larger than the tolerance for almost all points in the train set. However, when the RB space is rich enough, more and more points are removed, and eventually, almost all points in the initial train set are removed in later stages. For the

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

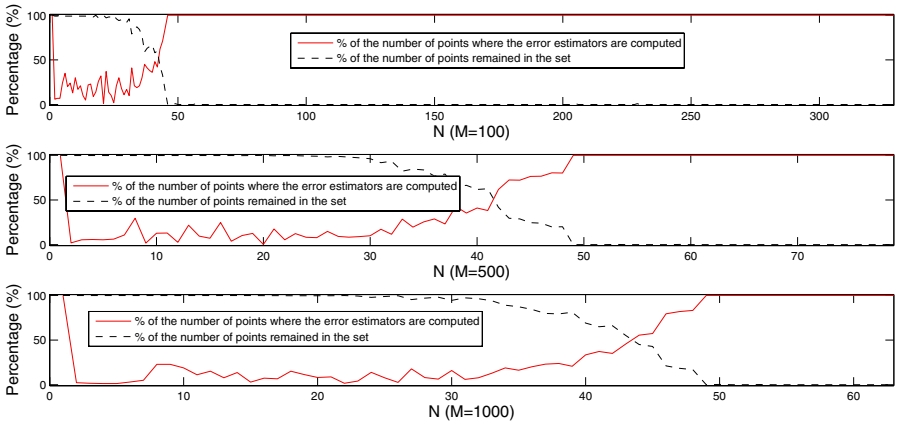


FIGURE 18. Percentage of work (effected at each step N) w.r.t. the total number of points M and of the number of points remained in the train set (at each step N) of the adaptively enriching greedy algorithm for Test 5.5 and different values of $M = 100$, 500, and 1000.

number of the points where the error estimators are computed, *i.e.*, those points cannot be skipped by using the saturation assumption, it behaves like the Algorithm 2 with a fixed train set since the train set barely changes 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%.

We should emphasize that the saving here is very substantial compared to a standard algorithm. In our algorithm, an extremely large number of train points are used (if cumulated over the iterations), and it is impractical to use such a large set in the standard greedy algorithm.

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 basis functions 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 coarse random train set, the algorithm is robust 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 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 practical 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 [30].

Acknowledgements. The first and the third author 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. Comput.* **44** (1985) 303–320.

[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) 667–672.

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press

[3] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova and P. Wojtaszczyk, Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.* **43** (2011) 1457–1472.

[4] A. Buffa, Y. Maday, A. Patera, C. Prud'homme and G. Turinici, *A priori* convergence of the greedy algorithm for the parametrized reduced basis. *M2AN* **46** (2012) 595–603. Special Issue in honor of David Gottlieb.

[5] T. Bui-Thanh, *Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems*, MIT Thesis (2007).

[6] T. Bui-Thanh, K. Willcox and O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM J. Sci. Comput.* **30** (2008) 3270–3288.

[7] 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) 1295–1300.

[8] 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) 1099–1116.

[9] 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) 970–996.

[10] 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** (2010) 3170–3200.

[11] J.L. Eftang and B. Stamm, Parameter multi-domain *hp* empirical interpolation. *Int. J. Numer. Meth. Engng.* **90** (2012) 412–428.

[12] B. Fares, J.S. Hesthaven, Y. Maday and B. Stamm, The reduced basis method for the electric field integral equation. *J. Comput. Phys.* **230** (2011) 5532–5555.

[13] 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) 575–605.

[14] M.A. Grepl and A.T. Patera, *A posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN* **39** (2005) 157–181.

[15] B. Haasdonk, M. Dihlmann and M. Ohlberger, A training set and multiple basis functions generation approach for parametrized model reduction based on adaptive grids in parameter space. *Math. Comput. Modell. Dyn. Syst.* **17** (2011) 423-442.

[16] B. Haasdonk and M. Ohlberger, Basis construction for reduced basis methods by adaptive parameter grids, in *Proc. International Conference on Adaptive Modeling and Simulation* **2007** (2007) 116–119.

[17] J.S. Hesthaven and S. Zhang, On the use of ANOVA expansions in reduced basis methods for high-dimensional parametric partial differential equations, Brown Division of Applied Math Scientific Computing Tech Report 2011-31.

[18] 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) 473–478.

[19] 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** (2009) 383–404.

[20] Y. Maday and B. Stamm, Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces, `arXiv`: math.NA, Apr 2012, *accepted in SIAM Journal on Scientific Computing*.

[21] 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.

[22] A. Quarteroni, G. Rozza and A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications. *J. Math. Ind.* **1** (2011) 3.

[23] S. Repin, *A Posteriori Estimates for Partial Differential Equations*, Walter de Gruyter, Berlin (2008).

[24] G. Rozza and K. Veroy, On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Eng.* **196** (2007) 1244–1260.

[25] 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 Comput. Methods Engrg.* **15** (2008) 229–275.

[26] S. Sen, Reduced-basis approximation and *a posteriori* error estimation for many-parameter heat conduction problems. *Numerical Heat Transfer, Part B: Fundamentals* **54** (2008) 369–389.

[27] V.N. Temlyakov, Greedy Approximation. *Acta Numerica* (2008) 235–409.

[28] K. Veroy, *Reduced-Basis Methods Applied to Problems in Elasticity: Analysis and Applications*, MIT Thesis (2003).

[29] K. Veroy, C. Prudhomme, D.V. Rovas and A. Patera, *A posteriori* error bounds for reduced basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations, in *Proc. 16th AIAA Comput. Fluid Dynamics Conf.* (2003). Paper 2003–3847.

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

https://doi.org/10.1051/m2an/2013100 Published online by Cambridge University Press