Parameter multi-domain "*hp*" empirical interpolation[*]

Jens L. Eftang[†] Benjamin Stamm[‡]

Abstract

In this paper, we introduce two parameter multi-domain "*hp*" techniques for the empirical interpolation method (EIM). In both approaches, we construct a partition of the original parameter domain into parameter subdomains: *h*-refinement. We apply the standard EIM independently within each subdomain to yield local (in parameter) approximation spaces: *p*-refinement. Further, for a particularly simple case we introduce *a priori* convergence theory for the partition procedure. We show through two numerical examples that our approaches provide significant reduction in the EIM approximation space dimension, and thus significantly reduce the computational cost associated with EIM approximations.

Keywords: empirical interpolation method; *hp*-EIM, *hp*-convergence, reduced basis method

1 Introduction

The Empirical Interpolation Method (EIM) was first introduced in [1, 8] as a tool within the Reduced Basis (RB) framework [12] for parametrized partial differential equations (PDEs). The EIM serves to construct "affine" (more precisely, affine in functions of the parameter) approximations of non-affine or non-linear [7] parametrized differential operators. This approximation is achieved through an affine approximation of the coefficient function which separates the parameter and spatial dependence. An affine decomposition of the differential operator is necessary to enable efficient RB offline-online computational procedures. The EIM thus expands the class of PDEs amenable to RB treatment. Other applications of the EIM include rapid numerical approximation of parametrized integrals and are discussed in [10]. In [2], the related Discrete Empirical Interpolation Method is employed for model order reduction of non-linear dynamical systems.

[*] Accepted in International Journal for Numerical Methods in Engineering, September, 2011. DOI: 10.1002/nme.3327

[†] Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway. eftang@mit.edu

[‡] Department of Mathematics, University of California, Berkeley and Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Given "any" parametrized function, the EIM precomputation (henceforth *offline*) stage serves to construct an approximation space spanned by "snapshots" of this function for judiciously chosen parameter values from a predefined parameter domain *and* a set of judiciously chosen spatial interpolation nodes from the spatial domain. In the EIM online stage, given *any* new parameter value from the parameter domain, the EIM approximation to the original function is the particular linear combination of the EIM basis functions that interpolates the original function at the spatial interpolation nodes.

Under the assumption that the function under consideration depends smoothly on the parameters, the EIM typically provides exponential convergence [10]. However, for many problems in which the function exhibits large (albeit smooth) variations with the parameters, a snapshot from one region of the parameter domain contributes little to the approximation of the function associated with a parameter value from another region of the parameter domain. The global (in parameter, but of course also physical space) EIM approximation space is thus in some sense unnecessecarily large, and consequently the online computation of the EIM approximation is unnecessecarily expensive.

In this paper, we introduce two approaches that both serve to reduce the dimension of the EIM approximation space. Both approaches share the same underlying idea: an adaptive partition of the parameter domain into parameter subdomains — h-refinement — and construction of standard EIM approximation spaces and associated EIM interpolation nodes restricted to each of these parameter subdomains — p-refinement. This parameter multi-domain, or *hp*, strategy provides significant dimension reduction since the smaller local (in parameter) EIM approximation spaces are optimized with respect to the parametric variations within each subdomain; the online evaluation of the EIM interpolant is thus much faster.

Our first approach is the *anchor point* (AP) splitting scheme. This method is an adaption of the *hp*-RB method introduced in [5] to the context of the EIM. The parameter subdomains are hierarchically defined based on proximity to "anchor points" identified by the EIM Greedy sampling procedure within each subdomain at each level of h-refinement; subsequently, in the p-refinement stage, the standard EIM is applied within each subdomain. Our second approach is the *gravity center* (GC) splitting scheme. The parameter subdomains are hierarchically defined based on the "gravity center" of a cloud of points identified by the EIM Greedy sampling procedure at each level of concurrent h- and p-refinement.

We provide in the next section a general problem statement along with notation required later. We then review in Section 3 the standard EIM applied to the entire parameter domain $\mathcal{D}$. Then, we present in Section 4 and Section 5 the AP and GC splitting procedures, respectively. In Section 6 we discuss for both methods the computational decoupling into offline and online stages as well as associated computational costs. In Section 7 we compare our two approaches relative to the standard EIM through two model problems. Finally, we provide some concluding remarks in Section 8.

2 Problem Statement

We introduce a spatial domain $\Omega \subset \mathbb{R}^d$ for some integer $d > 0$; we shall denote a particular spatial point $x \in \Omega$ as $x = (x_{(1)}, \ldots, x_{(d)})$. We next introduce a parameter domain $\mathcal{D} \subset \mathbb{R}^P$; we shall denote a particular parameter value $\mu \in \mathcal{D}$ as $\mu = (\mu_{(1)}, \ldots, \mu_{(P)})$. We then introduce a (given) parametrized function $\mathcal{G} : \Omega \times \mathcal{D} \to \mathbb{R}$ such that $\mathcal{G}(\cdot; \mu) \in L^\infty(\Omega)$ for all $\mu \in \mathcal{D}$; here $L^\infty(\Omega) = \{v : \operatorname{ess\,sup}_{x \in \Omega} |v(x)| < \infty\}$. We finally introduce a triangulation $\mathcal{T}^{\mathcal{N}}(\Omega)$ with $\mathcal{N}$ vertices over which we shall in practice realize $\mathcal{G}(\cdot; \mu)$, $\mu \in \mathcal{D}$, as a piecewise linear function.[1]

For any $\mu \in \mathcal{D}$, we consider the construction of an approximation $\mathcal{G}_M(\cdot; \mu) \approx \mathcal{G}(\cdot; \mu)$, where $\mathcal{G}_M(\cdot; \mu)$ resides in a parameter-independent M-dimensional linear approximation space W_M, $M < \infty$. The problem is thus twofold: *i)* the construction of a good M-dimensional approximation space $W_M = \operatorname{span}\{q_1, \ldots, q_M\}$ and *ii)* given any $\mu \in \mathcal{D}$ and the space W_M, the computation of parameter dependent coefficients $\varphi_1(\mu), \ldots, \varphi_M(\mu)$ such that

$$\mathcal{G}_M(\cdot; \mu) = \sum_{m=1}^{M} \varphi_m(\mu) q_m \approx \mathcal{G}(\cdot; \mu) \tag{1}$$

is a good approximation for all $\mu \in \mathcal{D}$. Clearly, classical polynomial interpolation procedures may be considered for this problem; however in our context here standard polynomial approximation spaces are far too general and hence the required dimension M is too large to accomodate efficient online evaluation of $\varphi_1(\mu), \ldots, \varphi_M(\mu)$. In contrast, the EIM provides a much smaller approximation space specifically targeted at the parametrized function at hand.

The following simple problem illustrates how the EIM may be invoked in practice. Consider the integral

$$F(\mu) = \int_\Omega f(\cdot; \mu) u, \tag{2}$$

where $f : \Omega \times \mathcal{D} \to \mathbb{R}$ with $f(\cdot; \mu) \in L^\infty(\Omega)$ for any $\mu \in \mathcal{D}$ and $u : \Omega \to \mathbb{R}$ is a parameter independent function (we assume that the product $f(\cdot; \mu)u$ is integrable). In general, evaluation of $F(\mu)$ with standard quadrature rules may be expensive; in particular, the evaluation cost may be prohibitive when $F(\mu)$ has to be computed for many $\mu \in \mathcal{D}$ or in real time. The EIM serves to construct an approximation $f_M(\cdot; \mu) = \sum_{m=1}^{M} \varphi_m(\mu) q_m$ to $f(\cdot; \mu)$ such that

$$F_M(\mu) \equiv \int_\Omega f_M(\cdot; \mu) u = \sum_{m=1}^{M} \varphi_m(\mu) \int_\Omega q_m u \tag{3}$$

is a good approximation to $F(\mu)$. The key point is that the separation provided by the EIM allows precomputation of the integrals on the right in (3) (by for ex-

[1] We emphasize that the EIM is not restricted to functions that are piecewise linear; however, for the computational procedures involved, a finite-dimensional representation of $\mathcal{G}(\cdot; \mu)$, $\mu \in \mathcal{D}$, is required.

ample standard quadrature rules). Hence, assuming that the coefficients $\varphi_m(\mu)$ can be efficiently computed, subsequent evaluation $\mu \to F_M(\mu)$ is very fast.

3 The Empirical Interpolation Method

The EIM was first introduced in [1] (see also [8] for a more elaborate presentation). In this paper however we shall employ the particular version of the EIM introduced in [10], which invokes the computationally less expensive interpolation error rather than the computationally more expensive projection error as a tool in the offline construction of the EIM approximation space. We now briefly review the EIM applied to the entire parameter domain $\mathcal{D}$; in the next sections we then consider the EIM within the *hp* context of this paper.

We first introduce the empirical interpolation of a function $\mathcal{G} : \Omega \times \mathcal{D} \to \mathbb{R}$; we require that $\mathcal{G}(\cdot;\mu) \in L^\infty(\Omega)$ for all $\mu \in \mathcal{D}$. We introduce the EIM space $W_M = \text{span}\{q_m\}_{m=1}^M$ of dimension M and the M EIM interpolation nodes $t_1, \dots, t_M \in \Omega$ (the EIM basis functions q_m, $1 \le m \le M$, and interpolation nodes will be defined shortly). We may now define, for any $\mu \in \mathcal{D}$, the empirical interpolation $\mathcal{G}_M(\cdot;\mu) \approx \mathcal{G}(\cdot;\mu)$, as the particular function in W_M that interpolates $\mathcal{G}(\cdot;\mu)$ at the M interpolation nodes:

$$\mathcal{G}_M(\cdot;\mu) = \sum_{m=1}^{M} \varphi_m(\mu) q_m, \tag{4}$$

where the coefficients $\varphi_j(\mu)$, $1 \le j \le M$, solve the linear system

$$\sum_{j=1}^{M} \varphi_j(\mu) q_j(t_i) = \mathcal{G}(t_i;\mu), \quad 1 \le i \le M. \tag{5}$$

Clearly $\mathcal{G}_M(t_i;\mu) = \mathcal{G}(t_i;\mu)$, $1 \le i \le M$, for all $\mu \in \mathcal{D}$.

We now define the EIM basis functions and the EIM interpolation nodes recursively through a Greedy sampling algorithm. To this end we require a (typically rich) training set $\Xi_{\text{train}} \subset \mathcal{D}$ of finite size $|\Xi_{\text{train}}|$ which shall serve as our computational surrogate for $\mathcal{D}$. First, for $M = 1$, we choose (randomly, say) an initial parameter value $\mu_M \in \mathcal{D}$; the first EIM interpolation node is then $t_M \equiv \arg\sup_{x\in\Omega} |\mathcal{G}(x;\mu_M)|$;[2] the first EIM basis function is $q_M \equiv \mathcal{G}(\cdot;\mu_M)/\mathcal{G}(t_M;\mu_M)$. Then, for $2 \le M \le M_{\max} < \infty$, we compute for all $\mu \in \Xi_{\text{train}} \subset \mathcal{D}$ the empirical interpolation $\mathcal{G}_{M-1}(\cdot;\mu) \approx \mathcal{G}(\cdot;\mu)$; the next parameter is then chosen as the maximizer of the $L^\infty(\Omega)$ interpolation error over Ξ_{train}:

$$\mu_M \equiv \arg \max_{\mu\in\Xi_{\text{train}}} \|\mathcal{G}_{M-1}(\cdot;\mu) - \mathcal{G}(\cdot;\mu)\|_{L^\infty}. \tag{6}$$

[2] Note that $\sup_{x\in\Omega} |\mathcal{G}(x;\mu)|$ is in practice realized as the maximum of $|\mathcal{G}(x;\mu)|$ over the $\mathcal{N}$ vertices of $\mathcal{T}^{\mathcal{N}}(\Omega)$.

We define $r_M \equiv \mathcal{G}_{M-1}(\cdot;\mu_M) - \mathcal{G}(\cdot;\mu_M)$ and choose the next EIM interpolation node as

$$t_M \equiv \arg\sup_{x\in\Omega} |r_M(x)|. \tag{7}$$

We may now finally define the next EIM basis function as

$$q_M \equiv \frac{r_M}{r_M(t_M)}. \tag{8}$$

We have thus obtained $M_{\max}$ basis functions and $M_{\max}$ interpolation nodes. We note that by construction $r_M(t_i) = 0$ for $1 \le i \le M-1$; hence $q_M(t_i) = 0$ for $1 \le i \le M-1$ and $q_M(t_M) = 1$ thanks to the normalization (8). The matrix $\{q_j(t_i)\}_{i\,j}$ in (5) is thus lower triangular with unity diagonal; as a result, for any $\mu \in \mathcal{D}$, the cost associated with the computation of the coefficients $\varphi_j(\mu)$, $1 \le j \le M$, is $\mathcal{O}(M^2)$.

We also define, for $1 \le M \le M_{\max}$, the "Lebesgue constant" [11]

$$\Lambda_M \equiv \sup_{x\in\Omega} \sum_{m=1}^{M} |V_m^M(x)|, \tag{9}$$

where $V_m^M \in W_M$, $1 \le m \le M$, are the "characteristic functions" of W_M, which satisfy $V_m^M(t_n) = \delta_{mn}$; here δ_{mn} is the Kronecker delta symbol. It can be proven [1, 8] that the EIM approximation error satisfies

$$\|\mathcal{G}(\cdot;\mu) - \mathcal{G}_M(\cdot;\mu)\|_{L^\infty} \le (1+\Lambda_M) \inf_{z\in W_M} \|\mathcal{G}(\cdot;\mu) - z\|_{L^\infty}, \quad 1 \le M \le M_{\max}, \tag{10}$$

Furthermore, it can be proven that $\Lambda_M \le 2^M - 1$; however in actual practice the behavior of Λ_M is much better [1, 8, 10].

4 An Anchor Point Splitting Scheme

4.1 Procedure

In this section we introduce the anchor point (AP) splitting procedure for the partition of the parameter domain; this procedure is an adaption of the approach introduced for the *hp* Reduced Basis method in [5, 3]. We shall require a *distance function* $d : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$, which we choose in this paper as the Euclidean distance between the two arguments; however, other distance functions may be considered.

We first describe the splitting of an arbitrary subdomain $\mathcal{V} \subseteq \mathcal{D}$ into two distinct subdomains $\mathcal{V}_0 \subset \mathcal{V}$ and $\mathcal{V}_1 \subset \mathcal{V}$; the application of this splitting step to the construction of a partition of $\mathcal{D}$ is straightforward and is discussed shortly. We assume that $\mathcal{V}$ is equipped with a sufficiently dense training set

$\Xi_{\text{train}}^{\mathcal{V}} \subset \mathcal{V}$. Given an *anchor point* $\mu_0^* \in \mathcal{V}$, we first set $\mu_1 = \mu_0^*$. We then compute the snapshot $\mathcal{G}(\cdot;\mu_1)$ and the interpolation node $t_1 = \arg\sup_{x\in\Omega} |\mathcal{G}(x;\mu_1)|$ and perform one iteration (hence $M_{\max} = 2$) of the standard EIM Greedy procedure restricted to $\Xi_{\text{train}}^{\mathcal{V}}$ in order to identify a second parameter value $\mu_2 = \arg\max_{\mu\in\Xi_{\text{train}}^{\mathcal{V}}} \|\mathcal{G}_1(\cdot;\mu) - \mathcal{G}(\cdot;\mu)\|_{L^\infty}$; we denote the maximum interpolation error over $\Xi_{\text{train}}^{\mathcal{V}}$ by

$$\epsilon^{\mathcal{V}} \equiv \max_{\mu\in\Xi_{\text{train}}^{\mathcal{V}}} \|\mathcal{G}_1(\cdot;\mu) - \mathcal{G}(\cdot;\mu)\|_{L^\infty}. \tag{11}$$

We note that μ_2 is in some sense maximally different from μ_1 since $\mathcal{G}(\cdot;\mu_2)$ is the particular function that is "worst approximated" by the current EIM scheme for $M = 1$. We define $\mu_1^* \equiv \mu_2$ and may now introduce two distinct subdomains $\mathcal{V}_0 \subset \mathcal{V}$ and $\mathcal{V}_1 \subset \mathcal{V}$ based on proximity to the two points μ_0^* and μ_1^* as

$$\mathcal{V}_0 = \{\mu \in \mathcal{V} : d(\mu,\mu_0^*) < d(\mu,\mu_1^*)\}, \tag{12}$$

$$\mathcal{V}_1 = \{\mu \in \mathcal{V} : d(\mu,\mu_1^*) \leq d(\mu,\mu_0^*)\}. \tag{13}$$

We say that μ_0^* is to the anchor point of $\mathcal{V}_0$ and that μ_1^* is the anchor point of $\mathcal{V}_1$.

We apply this "h-refinement" splitting scheme in a recursive manner in order to construct a hierarchical partition of the entire domain $\mathcal{D}$. We first choose the initial anchor point—typically a corner of $\mathcal{D}$—and split $\mathcal{D}$ into two new subdomains. We then apply the splitting scheme within each of the two generated subdomains. We continue recursively until convergence: we split a subdomain $\mathcal{V}$ as long as the maximum error $\epsilon^{\mathcal{V}}$ in (11) is larger than a prescribed tolerance $\epsilon_{\text{tol}}^h > 0$. If $\epsilon^{\mathcal{V}} < \epsilon_{\text{tol}}^h$ we stop the splitting process. We note that each subdomain (except the "root" $\mathcal{D}$) has a single "parent" and one "sibling." Thanks to this hierarchical structure we may organize the splitting procedure in a binary tree as illustrated in Figure 1.

When the tolerance ϵ_{tol}^h is satisfied over $\Xi_{\text{train}}^{\mathcal{V}}$, we perform "$p$-refinement" within $\mathcal{V}$: application of the standard EIM to $\mathcal{V}$ for specified $M_{\max} > 1$ and target tolerance $\epsilon_{\text{tol}}^p < \epsilon_{\text{tol}}^h$. If the target tolerance ϵ_{tol}^p is *not* satisfied after p-refinement (for $M_{\max}$ basis functions), we successively perform additional "h" and "p" refinement steps until the tolerance is satisfied for at most $M_{\max}$ EIM basis functions. Our procedure thus enables simultaneous control over the EIM error (over the training set) and the EIM space dimension. The latter directly determines the computational cost associated with online evaluation of the EIM approximation; we discuss computational costs in detail in Section 6.

This hp-EIM anchor point refinement procedure results in K parameter subdomains, which we label $\mathcal{V}_1,\ldots,\mathcal{V}_K$. Each of these subdomains has an associated nested set of EIM approximation spaces, $W_M^k = \text{span}\{q_m^k\}_{m=1}^M$, $1 \leq M \leq M_{\max}^k$, $1 \leq k \leq K$, and an associated set of nested EIM interpolation nodes $T_M^k = \{t_1^k,\ldots,t_M^k\}$, $1 \leq M \leq M_{\max}^k$, $1 \leq k \leq K$. Here, the q_m^k denote the EIM basis functions, and $M_{\max}^k$ denotes the space dimension required in order to reach the target tolerance ϵ_{tol}^p for subdomain k. Note that the $M_{\max}^k$, $1 \leq k \leq K$, are in general different but bounded by $M_{\max}$.

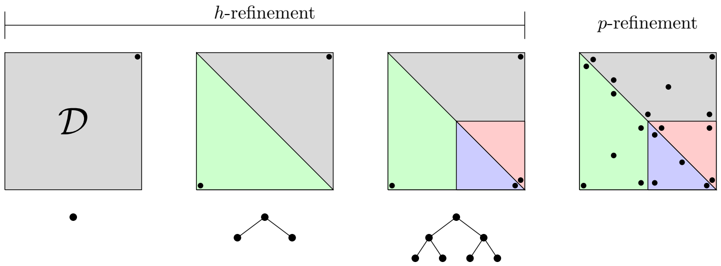


Figure 1: Two levels of h-refinement and subsequent p-refinement for the anchor point splitting procedure.

Given the partition of $\mathcal{D}$ into K subdomains with associated EIM approximation spaces and interpolation nodes, we now define the AP hp-EIM interpolant. Given any $\mu \in \mathcal{D}$, we first determine $k^* = k^*(\mu)$ such that $\mu \in \mathcal{V}_{k^*}$. Note that thanks to the binary tree structure of the partition, determination of k^* may be performed as an efficient binary search through comparison of the distances between μ and two anchor points at each level in the tree. We then compute the EIM approximation as

$$\mathcal{G}_{M,K}(\cdot;\mu) \equiv \mathcal{G}^{k^*}_{M,K}(\cdot;\mu) = \sum_{m=1}^{M} \varphi^{k^*}_{M,m}(\mu) q^{k^*}_m, \tag{14}$$

where the coefficients $\varphi^{k^*}_{M,j}(\mu)$, $1 \le j \le M$, solve the linear system

$$\sum_{j=1}^{M} \varphi^{k^*}_{M,j}(\mu) q^{k^*}_j (t^{k^*}_i) = \mathcal{G}(t^{k^*}_i;\mu), \quad 1 \le i \le M, \tag{15}$$

Remark 1. *We note that the partition of $\mathcal{D}$ can be organized in a binary tree as indicated in Figure 1 regardless of the parameter dimension P, since we always subdivide a subdomain into two new subdomains at each level of refinement. The method thus allows the partition to reflect anisotropy in the underlying parameter dependence of $\mathcal{G}(\cdot;\mu)$.*

4.2 An *a priori* convergence theory for the AP procedure

We present here an *a priori* theory for the convergence of the initial partition algorithm (h-refinement) presented above. The theory does not consider the subsequent p-refinement stage; however this step typically converges quickly and in particular exponentially fast when the underlying parameter dependence is smooth. Our interest is not in the asymptotic convergence of the partition procedure *per se*, since in practice we will always invoke an hp-type approximation rather than a pure h-type approximation. However, the theory suggests that our procedure generates a meaningful partition, and furthermore guarantees that the partition procedure does in fact terminate for specified ϵ_{tol}^h. We consider the case with $P = 1$ parameter for simplicity.

Proposition 1. *Let $\mathcal{D} \subset \mathbb{R}$ and let $|\mathcal{D}|$ denote the length of $\mathcal{D}$. Suppose that $\mathcal{G}$ is Lipschitz-contunuous in μ with Lipschitz constant $L < \infty$: for any $\mu_1 \in \mathcal{D}$, $\mu_2 \in \mathcal{D}$,*

$$\|\mathcal{G}(\cdot;\mu_1) - \mathcal{G}(\cdot;\mu_2)\|_{L^\infty} \leq L|\mu_1 - \mu_2|. \tag{16}$$

For any specified $\epsilon_{\text{tol}}^h > 0$, the AP splitting procedure is then convergent for $K = K(\epsilon_{\text{tol}}^h)$ subdomains; moreover, the convergence is first order in the sense that

$$K(\epsilon_{\text{tol}}^h) \leq \max\left\{1, \frac{4L|\mathcal{D}|}{\epsilon_{\text{tol}}^h}\right\}, \tag{17}$$

where $|\mathcal{D}|$ is the length of $\mathcal{D}$.

Proof: We consider our splitting procedure after generation of $\tilde{K}$ subdomains. Either we obtain convergence for $\tilde{K} = 1$ (i.e., $K = 1$) — in which case the proof is complete — or $\tilde{K} > 1$. We henceforth consider the case $\tilde{K} > 1$.

We consider the splitting of an arbitrary subdomain $\mathcal{V} \subset \mathcal{D}$ into distinct subdomains $\mathcal{V}_0 \subset \mathcal{V}$ and $\mathcal{V}_1 \subset \mathcal{V}$ as discussed in the previous subsection. We assume that the error tolerance ϵ_{tol}^h is not satisfied, hence $\epsilon^{\mathcal{V}} > \epsilon_{\text{tol}}^h$. Let $\mu_0^* \in \mathcal{V}$ denote the anchor point associated with $\mathcal{V}$. We then consider the empirical interpolation $\tilde{\mathcal{G}}(\cdot;\mu) = \tilde{\varphi}(\mu)\mathcal{G}(\cdot;\mu_0^*) \approx \mathcal{G}(\cdot;\mu)$ for any $\mu \in \mathcal{V}$. For the error in this approximation we obtain

$$\begin{aligned}\|\mathcal{G}(\cdot;\mu) - \tilde{\mathcal{G}}(\cdot;\mu)\|_{L^\infty} &\leq (1 + \Lambda_{(M=1)}) \inf_{z \in \text{span}\{G(\cdot;\mu_0^*)\}} \|\mathcal{G}(\cdot;\mu) - z\|_{L^\infty} \\ &\leq 2\|\mathcal{G}(\cdot;\mu) - \mathcal{G}(\cdot;\mu_0^*)\|_{L^\infty} \\ &\leq 2L|\mu - \mu_0^*|, \end{aligned} \tag{18}$$

where we first invoke (10), then choose $z = \mathcal{G}(\cdot;\mu_0^*)$, and finally invoke (16). Note that it follows from the definition of Λ_1 (Eq. (9)) and the characteristic function V_1^1 that $\Lambda_1 = 1$.

We now let $\mu = \mu_1^*$ denote the anchor point associated with $\mathcal{V}_1$, identified by the single EIM Greedy iteration over $\mathcal{V}$. Eq. (18) then yields

$$\|\mathcal{G}(\cdot;\mu_1^*) - \tilde{\mathcal{G}}(\cdot;\mu_1^*)\|_{L^\infty} \leq 2L|\mu_1^* - \mu_0^*|. \tag{19}$$

Since the error tolerance is not satisfied over $\mathcal{V}$, we have $\epsilon^{\mathcal{V}} = \|\mathcal{G}(\cdot;\mu_1^*) - \mathcal{G}_1(\cdot;\mu_1^*)\|_{L^\infty} > \epsilon_{\text{tol}}^h$. Hence

$$\epsilon_{\text{tol}}^h < 2L|\mu_1^* - \mu_0^*|. \tag{20}$$

We split $\mathcal{V}$ into $\mathcal{V}_0 \subset \mathcal{V}$, $\mathcal{V}_1 \subset \mathcal{V}$ based on Euclidian distance to the two anchor points. It is clear that the length of each subdomain, $|\mathcal{V}_0|$ and $|\mathcal{V}_1|$, is at least as large as half the distance between the anchor points. We thus obtain

$$|\mathcal{V}_i| \geq \frac{|\mu_1^* - \mu_0^*|}{2} > \frac{\epsilon_{\text{tol}}^h}{4L}, \qquad i = 0, 1. \tag{21}$$

We denote the $\tilde{K}$ subdomains generated by the algorithm so far by $\mathcal{S}_k \subset \mathcal{D}$, $1 \leq k \leq \tilde{K}$; we denote the length of $\mathcal{S}_k$ by $|\mathcal{S}_k|$. Each of these subdomains results from a splitting of a subdomain $\tilde{\mathcal{S}}_k \supset \mathcal{S}_k$ one level further up in the tree. Since $\mathcal{V}$ above was arbitrary, we can for any k, $1 \leq k \leq \tilde{K}$, set $\mathcal{V} = \tilde{\mathcal{S}}_k$ and conclude that

$$|\mathcal{S}_k| > \frac{\epsilon_{\text{tol}}^h}{4L}, \qquad 1 \leq k \leq \tilde{K}. \tag{22}$$

We define the length of the smallest subdomain as $\delta_{\tilde{K}} \equiv \min_{1 \leq k \leq \tilde{K}} |\mathcal{S}_k|$, and hence in particular $\delta_{\tilde{K}} > \epsilon_{\text{tol}}^h/(4L)$.

We complete the proof by a contradiction argument. Assume that $\tilde{K} > 4L|\mathcal{D}|/\epsilon_{\text{tol}}^h$. In this case

$$\tilde{K}\delta_{\tilde{K}} > \frac{4L|\mathcal{D}|}{\epsilon_{\text{tol}}^h}\delta_{\tilde{K}} > \frac{4L|\mathcal{D}|}{\epsilon_{\text{tol}}^h} \cdot \frac{\epsilon_{\text{tol}}^h}{4L} = |\mathcal{D}|. \tag{23}$$

On the other hand, it is clear that $\tilde{K}\delta_{\tilde{K}} \leq |\mathcal{D}|$ for $\tilde{K}$ subdomains. We have thus reached a contradiction, and we conclude that the algorithm can not generate $\tilde{K} > 4L|\mathcal{D}|/\epsilon_{\text{tol}}^h$ subdomains as long as the error tolerance is not satisfied. Hence the error tolerence be satisfied for, and thus the algorithm must terminate for, $K \leq 4L|\mathcal{D}|/\epsilon_{\text{tol}}^h$ subdomains. □

5 A Gravity Center Splitting Scheme

5.1 Procedure

In this section we introduce the gravity center (GC) splitting procedure. The GC procedure is similar to the AP procedure of the previous section: both approaches invoke the standard EIM Greedy sampling procedure recursively in order to generate a hierarchical partition. However, in contrast to the AP procedure, the GC procedure splits a given subdomain in a *structured* way based on the location of the gravity center of $M > 2$ parameter values; hence the GC procedure also invokes higher order approximation terms in the partition process. Furthermore, the GC procedure leads to a tensorized partition structure.

In order to define our tensorized subdomains we shall require a relational operator $\texttt{op}(i, j)$ defined as

$$\texttt{op}(i, j) = \begin{cases} \leq & \text{if} \quad \text{bin}(i-1)_{(j)} = 0 \\ > & \text{if} \quad \text{bin}(i-1)_{(j)} = 1 \end{cases} \tag{24}$$

where $\text{bin}(i)$ is the binary representation of i as a vector in $\{0, 1\}^P$. For each i, $1 \leq i \leq 2^P$, the relational operator $\texttt{op}(i, j)$, $1 \leq j \leq P$, shall serve to

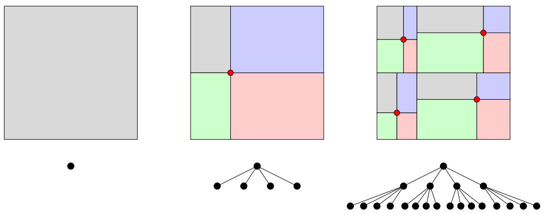


Figure 2: Two levels of parameter domain splitting for the gravity center procedure.

compare the jth components of two parameter values, and hence identify a subdomain $\mathcal{V}_i \subset \mathcal{V}$. We first describe the splitting of an "arbitrary" subdomain $\mathcal{V} = [a_{(1)}, b_{(1)}] \times \ldots \times [a_{(P)}, b_{(P)}] \subset \mathcal{D}$ into 2^P distinct subdomains, each of which may be written on tensor-product form $\mathcal{V}_i = [a^i_{(1)}, b^i_{(1)}] \times \ldots \times [a^i_{(P)}, b^i_{(P)}]$. The application of this splitting step to the construction of a partition of $\mathcal{D}$ is discussed shortly.

In order to split a subdomain $\mathcal{V}$, we first perform a standard EIM procedure within $\mathcal{V}$ for a target tolerance $\epsilon^p_{\rm tol}$ and a maximum EIM space dimension $M_{\max}$. If the target tolerance is satisfied over $\mathcal{V}$, i.e., $\epsilon^{\mathcal{V}} < \epsilon^p_{\rm tol}$, we terminate the procedure since further splitting of $\mathcal{V}$ is not required. If the target tolerance is *not* satisfied over $\mathcal{V}$, we obtain $M_{\max}$ parameter values $\{\mu_1, \ldots, \mu_{M_{\max}}\}$ from the EIM greedy procedure. The gravity point of the point cloud $\{\mu_i\}_{i=1}^{M_{\max}}$ is defined by

$$g = \frac{1}{M_{\max}} \sum_{i=1}^{M_{\max}} \mu_i. \tag{25}$$

The ith subdomain $\mathcal{V}_i$, $1 \le i \le 2^P$, in the partition of $\mathcal{V}$ is then defined by

$$\mathcal{V}_i = \{\mu \in \mathcal{V} : \mu_{(j)}\ \mathtt{op}(i,j)\ g_{(j)},\ \forall\, 1 \le j \le P\}; \tag{26}$$

where $\mu_{(j)}$ and $g_{(j)}$ represents the jth component of μ and g, respectively,

As with the AP splitting scheme, we apply the GC splitting scheme recursively in order to construct a hierarchical partition of the original domain $\mathcal{D}$. We start with a standard EIM greedy procedure within $\mathcal{D}$, and split $\mathcal{D}$ into 2^P new subdomains. We then apply the GC scheme within each of these subdomains, and continue the procedure recursively until convergence: the target tolerance is achieved over each subdomains with maximum EIM space dimension $M_{\max}$.

This *hp*-EIM refinement procedure results in K parameter subdomains, which we label $\mathcal{V}_1, \ldots, \mathcal{V}_K$. Each of these subdomains has an associated nested set of EIM approximation spaces, $W^k_M = \mathrm{span}\{q^k_m\}_{m=1}^M$, $1 \le M \le M^k_{\max}$, $1 \le k \le K$, and an associated set of nested EIM interpolation nodes $T^k_M = \{t^k_1, \ldots, t^k_M\}$, $1 \le M \le M^k_{\max}$, $1 \le k \le K$; as before $M^k_{\max}$ denotes the space dimension required in order to satisfy the target tolerance $\epsilon^p_{\rm tol}$ for subdomain

k.[3]

Given the partition of $\mathcal{D}$ into K subdomains with associated EIM approximation spaces and interpolation nodes, we now define the GC *hp*-EIM approximation. Given any $\mu \in \mathcal{D}$, we first determine $k^* = k^*(\mu)$ such that $\mu \in \mathcal{V}_{k^*}$. Note that thanks to the hierarchical structure of the partition, determination of k^* is an efficient 2^P-order search. We then compute the EIM approximation as

$$\mathcal{G}_{M,K}(\cdot;\mu) \equiv \mathcal{G}^{k^*}_{M,K}(\cdot;\mu) = \sum_{m=1}^{M} \varphi^{k^*}_{M,m}(\mu) q^{k^*}_m, \tag{27}$$

where the coefficients $\varphi^{k^*}_{M,j}(\mu)$, $1 \le j \le M$, solve the linear system

$$\sum_{j=1}^{M} \varphi^{k^*}_{M,j}(\cdot;\mu) q^{k^*}_j (t^{k^*}_i) = \mathcal{G}(t^{k^*}_i;\mu), \quad 1 \le i \le M. \tag{28}$$

Remark 2. *We note that the partition of $\mathcal{D}$ can be organized in a 2^P order tree as indicated in Figure 2, since at each level of refinement we subdivide a subdomain into 2^P new subdomains. As a result, the scheme does not take anisotropy in the underlying parameter dependence into account, and as a consequence may construct more subdomains than required. We provide some further comments on this issue in Section 7.*

5.2 An *a priori* convergence theory for the GC procedure

We present here an *a priori* theory for the convergence of the GC partition algorithm. The theory ensures that the algorithm does in fact terminate for specified ϵ^p_{tol} and $M_{\max}$. We consider the case with $P = 1$ parameter for simplicity.

Proposition 2. *Let $\mathcal{D} \subset \mathbb{R}$ and let $|\mathcal{D}|$ denote the length of $\mathcal{D}$. Suppose that $\mathcal{G}$ is Lipschitz-continuous in μ with Lipschitz constant $L < \infty$: for any $\mu_1 \in \mathcal{D}$, $\mu_2 \in \mathcal{D}$,*

$$\|\mathcal{G}(\cdot;\mu_1) - \mathcal{G}(\cdot;\mu_2)\|_{L^\infty} \le L|\mu_1 - \mu_2|. \tag{29}$$

For any specified $\epsilon^p_{\text{tol}} > 0$ and $M_{\max} \ge 2$, the "gravity center" splitting procedure is then convergent for $K = K(\epsilon^p_{\text{tol}})$ subdomains; moreover, the convergence is first order in the sense that

$$K(\epsilon^p_{\text{tol}}) \le \max\left\{1, \frac{2(1+\bar{\Lambda}_{M_{\max}})L|\mathcal{D}|}{\epsilon^p_{\text{tol}}}\right\}, \tag{30}$$

where $|\mathcal{D}|$ is the length of $\mathcal{D}$ and $\bar{\Lambda}_{M_{\max}} \equiv \max_{1 \le M \le M_{\max}} \Lambda_M$.

[3] Admittedly, we should here introduce separate notation for the AP and GC splitting procedures. In particular, the number of subdomains K as well as entities associated with each subdomain (such as $W^k_M, T^k_M, M^k_{\max}$) should bear subscripts AP and GC. However, we omit these subscripts for simplicity of notation. When we later compare the two approches in terms of numerical results, we introduce separate notation only as necessary.

Proof: We demonstrate here only a lower bound for the length of a subdomain (analogous to (21)). The remainder of the proof is then identical to the proof of Proposition 1.

We consider the splitting of an arbitrary subdomain $\mathcal{V} = [a, b] \subset \mathcal{D}$ into $2^{P=1} = 2$ distinct subdomains $\mathcal{V}_1$ and $\mathcal{V}_2$ as discussed in the previous subsection. We assume that the error tolerance ϵ_{tol}^p is not satisfied. In $\mathcal{V}$ we choose by virtue of the standard EIM Greedy procedure $M_{\max}$ parameter values μ_m, $1 \leq m \leq M_{\max}$, and compute the associated snapshots $\mathcal{G}(\cdot;\mu_m)$, $1 \leq m \leq M_{\max}$. For any $\mu \in \mathcal{V}$ and $1 \leq M \leq M_{\max}$ we then consider the empirical interpolation $\tilde{\mathcal{G}}_M(\cdot;\mu) = \sum_{m=1}^M \tilde{\varphi}_m(\mu)\mathcal{G}(\cdot;\mu_m) \approx \mathcal{G}(\cdot;\mu)$. Let $\tilde{W}_M = \text{span}\{\mathcal{G}(\cdot;\mu_m)\}_{m=1}^M$ denote an EIM space associated with $\mathcal{V}$. For the EIM approximation error we obtain, for any $\mu \in \mathcal{V}$,

$$\|\mathcal{G}(\cdot;\mu) - \tilde{\mathcal{G}}_M(\cdot;\mu)\|_{L^\infty} \leq (1 + \Lambda_M) \inf_{z \in \tilde{W}_M} \|\mathcal{G}(\cdot;\mu) - z\|_{L^\infty} \tag{31}$$

$$\leq (1 + \Lambda_M)\|\mathcal{G}(\cdot;\mu) - \mathcal{G}(\cdot;\mu_m)\|_{L^\infty} \tag{32}$$

$$\leq (1 + \Lambda_M)L|\mu - \mu_m|, \tag{33}$$

for $1 \leq m \leq M$ and $1 \leq M \leq M_{\max}$.

Since the error tolerance is not satisfied we have $\epsilon_{\text{tol}}^p \leq \|\mathcal{G}(\cdot;\mu_{M+1}) - \tilde{\mathcal{G}}_M(\cdot;\mu_{M+1})\|_{L^\infty(\Omega)}$ for $1 \leq M \leq M_{\max}$ (note that here, $\mu_{M_{\max}+1}$ denotes the parameter value at which the maximum error is obtained with $M_{\max}$ basis functions). With (31)–(33) we thus obtain

$$|\mu_{M+1} - \mu_m| \geq \frac{\epsilon_{\text{tol}}^p}{(1 + \Lambda_M)L} \tag{34}$$

for $1 \leq m \leq M$ and $1 \leq M \leq M_{\max}$.

The subdomain $\mathcal{V} = [a, b]$ is split at its gravity center g defined in (25) into $\mathcal{V}_1 = [a, g]$ and $\mathcal{V}_2 = (g, b]$. We now bound g away from a and b, and thus obtain a lower bound for $|\mathcal{V}_1| = |g - a|$ and $|\mathcal{V}_2| = |b - g|$. In fact we consider only a bound for $|\mathcal{V}_1|$ since the argument for $|\mathcal{V}_2|$ is analogous.

It is clear that there exist i, j, $1 \leq i, j \leq M_{\max}$, such that $\mu_i < g < \mu_j$ since there must be at least one greedily chosen parameter value on each side of g (more generally the gravity center must reside inside the convex hull of the $M_{\max} \geq 2$ greedily chosen parameter values). We consider the situation in which only one of these $M_{\max}$ parameter values resides to the left of g; if more than one of these parameter values reside to the left of g, we will for some $1 \leq M \leq M_{\max} - 1$ and some $1 \leq m \leq M$ have $|\mathcal{V}_1| \geq |\mu_{M+1} - \mu_m|$ ($|\mathcal{V}_1|$ must be larger than the distance between two greedily chosen parameter values) and hence in this case (34) applies directly as a lower bound for $|\mathcal{V}_1|$.

We denote the particular parameter value that resides to the left of g by μ_i. In this case *at least* one greedily chosen parameter value μ_j resides to the right of g; hence $|g - \mu_i| \geq |\mu_i - \mu_j|/2$ since g can not be closer to μ_i than to μ_j. Next, assume that μ_i was chosen by the Greedy algorithm at iteration $M + 1$, $1 \leq M \leq M_{\max} - 1$, and that μ_j was chosen at iteration m, $1 \leq m \leq M$. We thus set $\mu_{M+1} = \mu_i$ and $\mu_m = \mu_j$ in (34). If μ_i was chosen by the Greedy algorithm before μ_j, we set $\mu_{M+1} = \mu_j$ and $\mu_m = \mu_i$ in (34). In any case we obtain $|g - \mu_i| \geq |\mu_{M+1} - \mu_m|/2 \geq \epsilon_{\text{tol}}^p/(2(1+\Lambda_M)L)$ by (34).

We finally note that either, $a = \mu_i$ or a resides to the left of μ_i. We thus obtain

$$|g - a| \geq |g - \mu_i| \geq \frac{\epsilon_{\text{tol}}^p}{2(1+\Lambda_M)L} \geq \frac{\epsilon_{\text{tol}}^p}{2(1+\bar{\Lambda}_{M_{\max}})L}. \qquad (35)$$

The bound for $|b - g|$ is identical and invokes analogous arguments.

Eq. (35) is analogous to (21) in the proof of Proposition 1. The arguments in the remainders of the proofs are identical. □

6 Computational Cost

We now discuss the computational cost associated with the *hp*-EIM approaches presented above. We discuss the cost for the two methods concurrently since the separations of the computations into offline stages (expensive but performed once) and online stages (inexpensive and performed many times) are very similar.

In the *hp*-EIM offline stage, we perform h- and p-refinement: parameter domain partition and construction of EIM spaces and EIM interpolation nodes restricted to each parameter subdomain. The offline stage is expensive, since the cost depends on the (typically large) number $\mathcal{N}$ of vertices in the triangulation $\mathcal{T}^{\mathcal{N}}(\Omega)$. In particular, if we assume that the generated partition has K subdomains, we must perform $KM_{\max}\mathcal{N}$ function evaluations in order to construct an EIM space of dimension $M_{\max}$ associated with each subdomain. For the GC approach, we must also perform $M_{\max}\mathcal{N}$ function evaluations for each intermediate space associated with an intermediate subdomain; for this reason we expect the GC approach to be more expensive than the AP approach in the offline stage.

In the *hp*-EIM online stage, given any new parameter value $\mu \in \mathcal{D}$, we first determine to which subdomain $\mathcal{V}_{k^*} \subset \mathcal{D}$ the new parameter value belongs. For the AP approach this search can be performed at cost $\mathcal{O}(\log_2(K))$ for K subdomains since the subdomains can be organized in a binary tree: at each level in the tree a comparison between the distances from μ to two anchor points determines whether to proceed to the left or to the right branch. For the GC approach this search can be performed at cost $\mathcal{O}(\log_{2^P}(K))$ since the subdomains can be organized in a tree of order 2^P: at each level in the tree an elementwise comparison between μ and the gravity center g determines to which of the 2^P branches to proceed at the next level. We note that for the same number of subdomains and for $P > 1$ parameters, we expect that the GC approach yields the more efficient search since the tree has fewer levels and the cost at each level is roughly the same — $\mathcal{O}(P)$ for both approaches. However the cost of this search is in any event typically negligible.

Once the correct subdomain $\mathcal{V}_{k^*}$ that contains the given parameter value μ has been determined, we perform the standard EIM online stage: we solve a system of the form (5) at cost $\mathcal{O}(M^2)$ in order to evaluate the EIM coefficients. To form this system we require evaluation of $\mathcal{G}(\cdot;\mu)$ only at the M spatial interpolation nodes in T^{k^*}. Of course, if we wish to additionally visualize $\mathcal{G}_M(x;\mu)$ over $\mathcal{T}^{\mathcal{N}}(\Omega)$ the cost becomes $\mathcal{N}$-dependent.

Both of our *hp*-EIM procedures provide a reduction in the EIM *online* computational cost through reduction in the number of EIM basis functions required for the EIM space associated with a given subdomain for a specified error tolerance. However, we should emphasize that the *hp*-EIM offline stage may be significantly more expensive than a standard EIM offline stage. The reason is that although an *hp*-EIM approach provides dimension reduction for each (parametrically local) approximation space, the *total* number of function evaluations required to construct all spaces (for the AP procedure roughly $KM_{\max}\mathcal{N}$) may be significantly larger than the total number of function evaluations required for a standard EIM offline procedure: a (user specified) decrease in $M_{\max}$ necessitates a superlinear increase in K, and hence the product $KM_{\max}$ grows with decreasing $M_{\max}$. However, in many applications (when the EIM approximation is required in real time or for a large number of different parameter values) an increase in the offline cost may be acceptable if this additional effort results in a decrease in the subsequent online cost.

For each subdomain, we must compute and store a lower-triangular matrix $\{q_j^k(t_i)\}$ of (maximum) size $M_{\max} \times M_{\max}$; hence the online storage requirement is $\mathcal{O}(KM_{\max}^2)$. The product $KM_{\max}^2$ typically increases with decreasing $M_{\max}$ and hence the *hp*-EIM procedures require larger online storage than the standard EIM procedure. However, the quadratic scaling in $M_{\max}$ helps to moderate this online storage increase.

We finally note that an alternative "discretely orthogonal" basis for W_M is $\{V_m^M, 1 \leq m \leq M\}$. This basis enables $\mathcal{O}(M)$ online computational cost and $\mathcal{O}(KM_{\max})$ online storage since $\{q_j(t_i)\}$ in (5) is in this case replaced by the identity matrix $\{\delta_{ji}\}$. However, this basis is not hierarchical since $\{V_m^{M-1}, 1 \leq m \leq M-1\} \nsubseteq \{V_m^M, 1 \leq m \leq M\}$ and hence the computation of the characteristic functions would have to be computed as an additional final step in the EIM precomputation procedure. In any event, when the EIM is applied within the Reduced Basis framework for parametrized PDEs, the computational cost of the RB online stage scales as M^2 independent of the choice of the EIM basis [8]. For this reason, and for simlicity of exposition, we consider in this paper the standard EIM basis functions $q_m, 1 \leq m \leq M$.

7 Numerical Results

We present in this section numerical results for our two *hp*-EIM approaches applied to two model problems. In all cases, the *hp*-EIM yields signifact online computational speedup compared to the standard EIM.

	$M_{\max}$	ϵ^h_{tol}	$K(M_{\max}, \epsilon^h_{\text{tol}})$
Computation 1	143	1	2
Computation 2	77	0.99	12
Computation 3	40	0.8	55
Computation 4	30	0.6	106

Table 1: Specified $M_{\max}$ and ϵ^h_{tol}, and the required number of subdomains $K(M_{\max}, \epsilon^h_{\text{tol}})$ for the anchor point procedure applied to Example 1.

7.1 Example 1: 2D Gaussian surface

We define the spatial domain $\Omega \equiv (0,1)\times(0,1) \subset \mathbb{R}^2$, and we introduce a regular triangulation $\mathcal{T}^{\mathcal{N}}(\Omega)$ with $\mathcal{N} = 2601$ vertices. We define the parameter domain $\mathcal{D} \equiv [0.3, 0.7] \times [0.3, 0.7] \subset \mathbb{R}^2$, and we introdue a "tensor-product" train sample $\Xi_{\text{train}} \subset \mathcal{D}$ of size 1600. We consider the Gaussian function

$$\mathcal{G}(x;\mu) \equiv \exp\biggl(-\frac{(x_{(1)} - \mu_{(1)})^2}{0.02} - \frac{(x_{(2)} - \mu_{(2)})^2}{0.02}\biggr), \tag{36}$$

for $x = (x_{(1)}, x_{(2)}) \in \Omega$ and $\mu = (\mu_{(1)}, \mu_{(2)}) \in \mathcal{D}$. The function $\mathcal{G}$ is thus parametrized by the location of its maximum.

Clearly, $\mathcal{G}$ is particularly well suited for *hp*-adaptivity: snapshots associated with μ in one region of $\mathcal{D}$ do not provide a good approximation for functions assocaited with μ in another region of $\mathcal{D}$. We thus expect an *hp*-EIM procedure to provide significant reduction in M for this example.

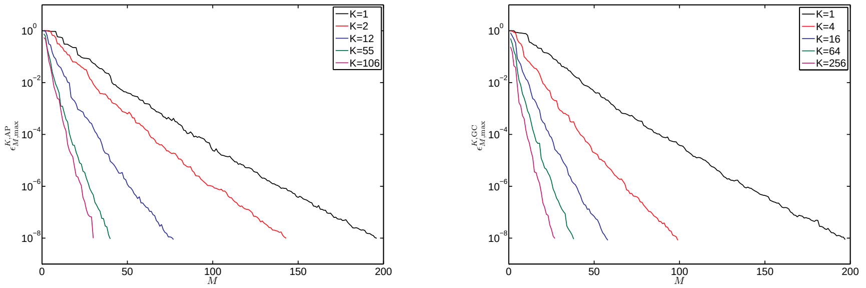
To obtain a benchmark for comparison, we first pursue the standard EIM with $M_{\max} = 196$, which corresponds to a tolerance 10^{-8} satisfied over Ξ_{train}. We note that the standard EIM is a special case both of the AP procedure and of the GC procedure for $K = 1$ subdomain.

We next pursue the AP splitting procedure. We specify $\epsilon^p_{\text{tol}} = 10^{-8}$ as the tolerance to be satisfied over the training set on each subdomain. We then perform four computations for different $M_{\max}$ and ϵ^h_{tol} and obtain partitions with $K(M_{\max}, \epsilon^h_{\text{tol}})$ subdomains as listed in Table 1. In Figure 3a we show the maximum error during each of the four computations,

$$\epsilon^{K,\text{AP}}_{M\max} = \max_{\mu \in \tilde{\Xi}^{\text{AP}}_{\text{train}}} \|\mathcal{G}^{\text{AP}}_{M,K}(\cdot;\mu) - \mathcal{G}(\cdot;\mu)\|_{L^\infty}, \tag{37}$$

for $K = 2, 12, 55, 106$ (we also show in Figure 3a the benchmark convergence for the case $K = 1$); here $\tilde{\Xi}^{\text{AP}}_{\text{train}}$ denotes the union of the train samples over each of the subdomains. In Figures 4a and 4b we show the partitions of $\mathcal{D}$ with $K = 12$ and $K = 55$ subdomains, respectively. We note that the size of the subdomains is rather uniform over $\mathcal{D}$, which reflects the uniform parameter dependence of $\mathcal{G}$, as expected.

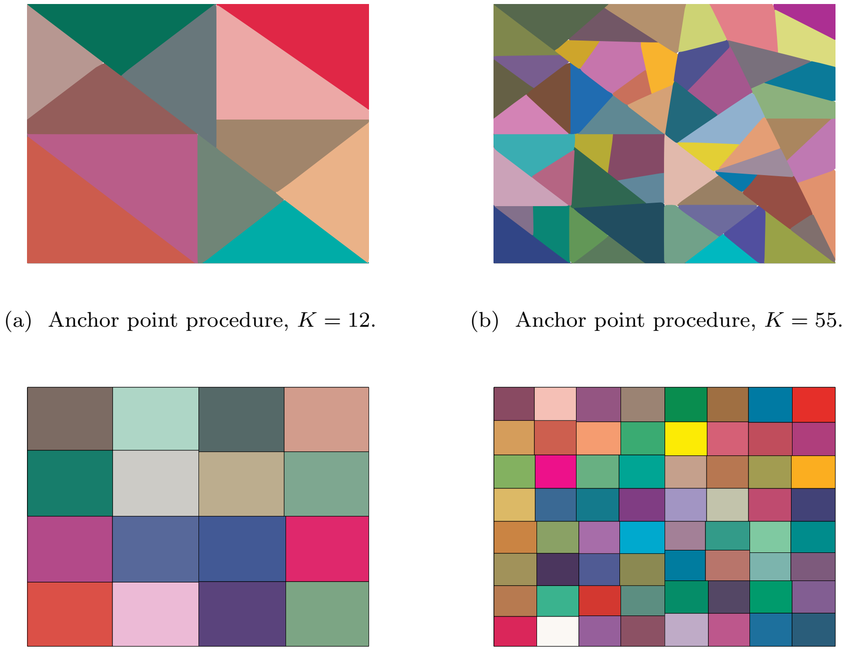
We then pursue the GC splitting procedure. We specify $\epsilon^p_{\text{tol}} = 10^{-8}$ as the tolerance to be satisfied over the training set on each subdomain. We then perform four computations for different $M_{\max}$ and obtain partitions with $K(M_{\max})$



(a) Anchor point procedure.

(b) Gravity center procedure.

Figure 3: Convergence for Example 1.



(a) Anchor point procedure, $K = 12$.

(b) Anchor point procedure, $K = 55$.

(c) Gravity center procedure, $K = 16$.

(d) Gravity center procedure, $K = 64$.

Figure 4: Parameter domain partitions for Example 1.

	$M_{\max}$	$K(M_{\max})$
Computation 1	99	4
Computation 2	58	16
Computation 3	38	64
Computation 4	27	256

Table 2: Specified $M_{\max}$ and the required number of subdomains $K(M_{\max})$ for the gravity center procedure applied to Example 1.

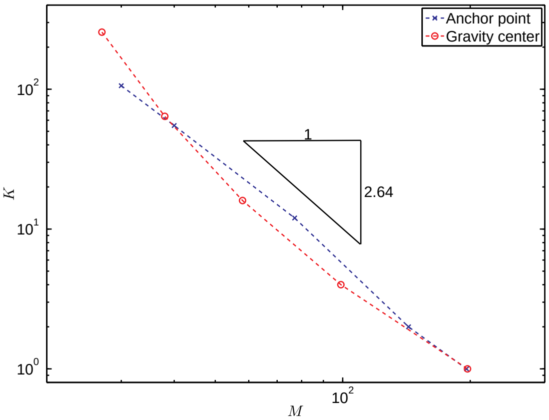


Figure 5: The number of required subdomains K as a function of specified $M_{\max}$ for a given satisfied tolerance ϵ^p_{tol}.

subdomains as listed in Table 2. In Figure 3b we show the maximum error during each of the four computations,

$$\epsilon^{K,\text{GC}}_{M,\max} = \max_{\mu \in \tilde{\Xi}^{\text{GC}}_{\text{train}}} \|\mathcal{G}^{\text{GC}}_{M,K}(\cdot;\mu) - \mathcal{G}(\cdot;\mu)\|_{L^\infty}, \tag{38}$$

for $K = 4, 16, 64, 256$; here $\tilde{\Xi}^{\text{GC}}_{\text{train}}$ denotes the union of the train samples over each of the subdomains. In Figures 4c and 4d we show the partitions of $\mathcal{D}$ with $K = 16$ and $K = 64$ subdomains, respectively. We note that the size of the subdomains is uniform.

We finally compare in Figure 5 our two approaches in terms of the number of required subdomains K for specified $M_{\max}$ such that $\epsilon^p_{\text{tol}} = 10^{-8}$ is satisfied over train samples over all subdomains — a measure of the optimality of the partition. We note that there is an algebraic relationship between K and $M_{\max}$, and that for Example 1 the two approaches perform very similarly in terms of the number of subdomains required for a specified tolerance. We further note that the product $KM_{\max}$ increases with K, and thus a smaller (used specified) $M_{\max}$ will result in larger offline computational cost. We would expect less steep curves in Figure 5 had we decreased the half-width of the Gaussian: a narrower

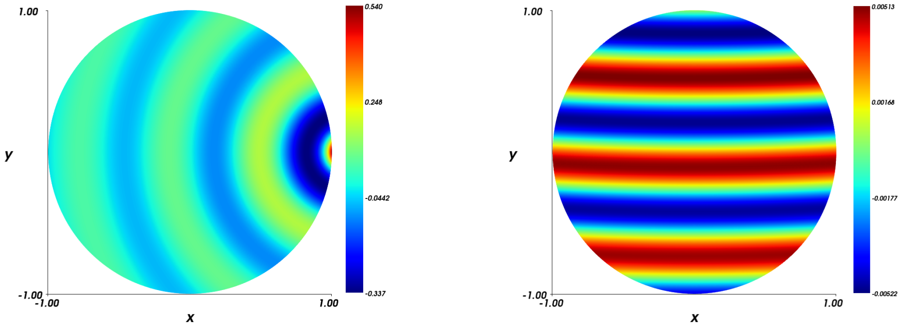


Figure 6: Example of the 3D wave function for a fixed $z = 0$ and parameter values $k = 10$, $c = (1.1, 0, 0)$ (left) and $k = 10$, $c = (0, 20, 0)$ (right).

Gaussian would have even more local parameter dependence and hence benefit more from hp-treatment.

7.2 Example 2: 3D wave function with near-singularity

Denote by $B_R(\mathcal{O})$ a ball in $\mathbb{R}^3$ with radius R centred at the origin $\mathcal{O}$. Then, consider the scalar function

$$\mathcal{G}(x;\mu) = \frac{\cos\left(k|x - c(\mu)|\right)}{k|x - c(\mu)|}, \qquad x = (x_{(1)}, x_{(2)}, x_{(3)}) \in \Omega = B_1(\mathcal{O}), \tag{39}$$

with $\mu = (k, r, \theta, \varphi) \in \mathcal{D}$ and $c(\mu) = r\left(\sin(\theta)\cos(\varphi), \sin(\theta)\sin(\varphi), \cos(\theta)\right) \in \mathbb{R}^3$. The parameter domain $\mathcal{D}$ is defined by $\mathcal{D} = [1, 10] \times [1.1, 20] \times [0, \pi/2] \times [0, \pi/2]$. The spatial domain Ω and the parameter domain $\mathcal{D}$ are discretized by $15 \times 15 \times 15$ and $8 \times 8 \times 8 \times 8$ tensorized grids, respectively, leading to the discrete versions $\mathcal{T}^{\mathcal{N}}(\Omega)$ and Ξ_{train}, respectively.

We note that $\mathcal{G}$ is particularly well suited for hp-adaptivity: the function has a very different structure for different wave numbers k and different locations of the pulse $c \in \mathbb{R}^3$. Snapshots with rapid and slow oscillations have little in common, and thus snapshots associated with k large contribute little to approximations of functions associated with k small, and vice versa. Similarly, snapshots with the singularity at c (outside but) close to Ω have high amplitude close to c and moderate amplitude elsewhere; such functions contribute little to the approximation of functions associated with c far from Ω, which have almost constant amplitude. Two examples of $\mathcal{G}$ for fixed $x_{(3)} = 0$ and parameter values $k = 10$, $c = (1.1, 0, 0)$ and $k = 10$, $c = (0, 20, 0)$ are shown in Figure 6; we note in particular the effect of c on the amplitude of the function.

To obtain a benchmark for comparison, we first pursue the standard EIM with $M_{\max} = 420$, which corresponds to a tolerance 10^{-3} satisfied over Ξ_{train}.

	$M_{\max}$	ϵ^h_{tol}	$K(M_{\max}, \epsilon^h_{\text{tol}})$
Computation 1	286	8	6
Computation 2	191	5	17
Computation 3	100	4	206

Table 3: Specified $M_{\max}$ and ϵ^h_{tol}, and the required number of subdomains $K(M_{\max}, \epsilon^h_{\text{tol}})$ for the anchor point procedure applied to Example 2.

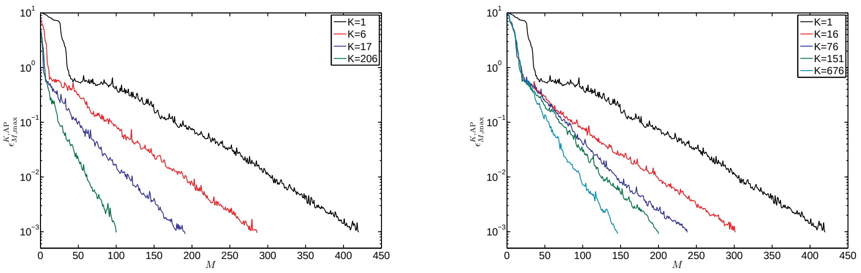
	$M_{\max}$	$K(M_{\max})$
Computation 1	301	16
Computation 2	238	76
Computation 3	200	151
Computation 4	146	676

Table 4: Specified $M_{\max}$ and the required number of subdomains $K(M_{\max})$ for the gravity center procedure applied to Example 2.

We next pursue the AP splitting procedure. We specify $\epsilon^p_{\text{tol}} = 10^{-3}$ as the tolerance to be satisfied over the training set on each subdomain. We then perform three computations for three different $M_{\max}$ and ϵ^h_{tol} and obtain partitions with $K(M_{\max}, \epsilon^h_{\text{tol}})$ subdomains as listed in Table 3. In Figure 7a we show the maximum error during each of the three computations, $\epsilon^{K,\text{AP}}_{M,\max}$ for $K = 6, 17, 206$.

We then pursue the GC splitting procedure. We specify $\epsilon^p_{\text{tol}} = 10^{-3}$ as the tolerance to be satisfied over the training set on each subdomain. We then perform three computations for four different $M_{\max}$ and obtain partitions with $K(M_{\max})$ subdomains as listed in Table 4. In Figure 7b we show the maximum error during each of the four computations, $\epsilon^{K,\text{GC}}_{M,\max}$ for $K = 16, 76, 151, 676$.

We finally compare in Figure 8 our two approaches in terms of the number of required subdomains K for specified $M_{\max}$ such that $\epsilon^p_{\text{tol}} = 10^{-3}$ is satisfied over the training sample on each subdomain. We note that there is an alge-



(a) Anchor point procedure.

(b) Gravity center procedure.

Figure 7: Convergence for Example 2.

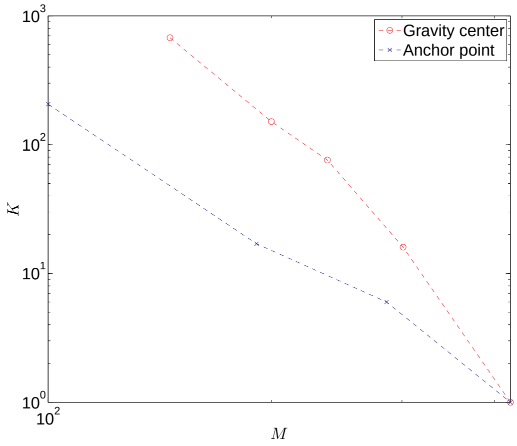


Figure 8: The number of required subdomains K as a function of specified $M_{\max}$ for a given satisfied tolerance ϵ_{tol}^p.

braic relationship between K and M, and that for Example 2 the AP approach seems to provide the somewhat more optimal partition. Presumably, this is a consequence of the ability of the AP procedure to reflect some anisotropy in the parameter dependence of $\mathcal{G}$ with respect to the number of subdomains generated.

8 Closing Remarks

The *hp*-EIM procedures derived in this paper provide a partition of the full parameter domain into parameter subdomains; a standard EIM approximation is pursued on each subdomain in order to satisfy a specified tolerance ϵ_{tol}^p for a specified maximum number $M_{\max}$ of EIM basis functions. Two different approaches are discussed. The first approach — the anchor point splitting procedure — is based on the first two modes associated with a standard EIM approximation: a given parameter (sub)domain is split into two new subdomains by a hyperplane equidistant from the the first two parameter values identified by the standard EIM Greedy sampling procedure. The second approach — the gravity center splitting procedure — is based on all $M_{\max}$ modes associated with a standard EIM approximation: a given parameter (sub)domain is split into 2^P tensorized new subdomains at the gravity center of the $M_{\max}$ parameter values identified by the EIM Greedy sampling procedure (recall that P is the dimension of the parameter domain). For both approaches, *a priori* convergence theory guarantees successful termination of the partition procedures.

Through two numerical examples we demonstrate that both the AP and GC approaches provide significant computational speedup in the EIM online stage through reduction in the required EIM space dimension for a given error tolerance. Admittedly, our two examples are particularly well suited for *hp*-EIM treatment. Functions with very smooth parameter dependence will be less suited for *hp*-EIM treatment. In such cases we expect that the required number of subdomains increases more rapidly with decreasing EIM space dimension, and hence the impact on the offline computational cost may be large. On the other hand, in applications where the EIM approximation is required many times or in real time, an increase in the offline cost may be acceptable if the result is a decrease in the online cost.

The AP approach seems to be somewhat better suited for higher dimensional parameter domains in particular when the parameter dependence of the function under consideration is anisotropic: only two new subdomains are introduced for each splitting. The GC approach is on the other hand arguably simpler in terms of implementation; in particular, the tensorized subdomain structure enables explicit construction of the parameter training sets associated with each subdomain.

A straightforward application of the *hp*-EIM procedures is within the Reduced Basis (RB) framework for order reduction of non-affine parametrized partial differential equations. In this context, the (*hp*-EIM or) EIM accomodates efficient offline-online computational procedures through affine approximations of the non-affine differential operator [1, 8, 7, 2, 13, 9]. The cost of the RB online stage grows quadratically with the number of terms in the affine approximation of the operator (M) and hence the *hp*-EIM approach will serve to reduce the RB online cost. Similarly, the *hp*-EIM approach may be applied within the related *hp*-RB framework [5] in order to provide a reduction both in the RB space dimension and in the EIM space dimension. In [6] the gravity center approach discussed in this paper is applied within an RB framework for the electric field integral equation.

The *hp*-EIM method may also provide an improvement of the rigorous *a posteriori* error bounds recently introduced for the EIM [4]. Currently these bounds are global in the parameters, and the *hp*-EIM thus provides a natural way of localizing, and hence in effect sharpen, the bounds.

Acknowledgements

We are grateful for many fruitful discussions with Prof. Anthony T. Patera and Prof. Martin A. Grepl. This work has been supported by the Norwegian University of Science and Technology, University of California, Berkeley, Lawrence Berkeley National Laboratory, and OSD/AFOSR Grant Number FA9550-09-1-0613.

References

[1] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization

of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.

[2] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[3] J. L. Eftang, D. J. Knezevic, and A. T Patera. An *hp* certified reduced basis method for parametrized parabolic partial differential equations. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):395–422, 2011.

[4] Jens L. Eftang, Martin A. Grepl, and Anthony T. Patera. A posteriori error bounds for the empirical interpolation method. *C. R. Math. Acad. Sci. Paris*, 348(9-10):575–579, 2010.

[5] Jens L. Eftang, Anthony T. Patera, and Einar M. Rønquist. An "*hp*" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM Journal on Scientific Computing*, 32(6):3170–3200, 2010.

[6] M. Fares, J. Hesthaven, Y. Maday, and B. Stamm. The reduced basis method for the parametrized electric field integral equation. *Journal of Computational Physics*, 230(14):5532–5555, 2011.

[7] M. Grepl. *A Posteriori* Error Bounds for Reduced-Basis Approximations of Nonaffine and Nonlinear Parabolic Partial Differential Equations. *Mathematical Models and Methods in Applied Sciences*, 22(3), 2012.

[8] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[9] A. E. Lvgren, Y. Maday, and E. M. Rnquist. The reduced basis element method: Offline-online decomposition in the nonconforming, nonaffine case. In Jan S. Hesthaven and Einar M. Rnquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 247–254. Springer Berlin Heidelberg, 2011.

[10] Yvon Maday, Ngoc Cuong Nguyen, Anthony T. Patera, and George S. H. Pau. A general multipurpose interpolation procedure: The magic points. *Communications in pure and applied mathematics*, 8:383–404, 2009.

[11] A. Quarteroni, R. Sacco, and F Saleri. *Numerical Mathematics*, volume 37. Springer, New York, 1991.

[12] 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. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[13] Gianluigi Rozza. Reduced basis methods for stokes equations in domains with non-affine parameter dependence. *Computing and Visualization in Science*, 12:23–35, 2009.