**Universität
Stuttgart**

A Training Set and Multiple Bases Generation Approach for Parametrized Model Reduction Based on Adaptive Grids in Parameter Space

Bernard Haasdonk, Markus Dihlmann, Mario Ohlberger

**Berichte aus dem Institut für
Angewandte Analysis und Numerische Simulation**

Preprint 2010/006

Universität Stuttgart

A Training Set and Multiple Bases Generation Approach for Parametrized Model Reduction Based on Adaptive Grids in Parameter Space

Bernard Haasdonk, Markus Dihlmann, Mario Ohlberger

Berichte aus dem Institut für Angewandte Analysis und Numerische Simulation

Preprint 2010/006

Institut für Angewandte Analysis und Numerische Simulation (IANS)
Fakultät Mathematik und Physik
Fachbereich Mathematik
Pfaffenwaldring 57
D-70 569 Stuttgart

E-Mail: `ians-preprints@mathematik.uni-stuttgart.de`
WWW: `http://preprints.ians.uni-stuttgart.de`

ISSN **1611-4176**

IANS-Logo: Andreas Klimke. LaTeX-Style: Winfried Geis, Thomas Merkle.

Abstract

Modern simulation scenarios require real-time or many-query responses from a simulation model. This is the driving force for increased efforts in Model Order Reduction (MOR) for high dimensional dynamical systems or partial differential equations (PDEs). This demand for fast simulation models is even more critical for parametrized problems. Several snapshot-based methods for basis construction exist for parametrized model order reduction, e.g. Proper Orthogonal Decomposition (POD) or Reduced Basis (RB) methods. They require the careful choice of samples for generation of the reduced model. In the current article we adress two types of grid-based adaptivity that can be beneficial in such basis generation procedures. First, we describe an approach for *training set adaptivity*. Second, we introduce an approach for multiple bases on *adaptive parameter domain partitions*. Due to the modularity, both methods also can easily be combined. They result in efficient reduction schemes with accelerated training times, improved approximation properties and control on the reduced basis size. We demonstrate the applicability of the approaches for instationary PDEs and parametrized dynamical systems.

keywords: parametrized model order reduction; reduced basis methods; adaptive parameter grids; snapshot and parameter selection

1 Introduction

Many modern numerical simulation scenarios are characterized by high dimensionality in the solution variable. Frequently, further dimensions of complexity are added by time-dependence, where the solution variable can considerably vary over time, or by parameter-dependence, where the problem and therefore the solution is changing for each newly given parameter. This is the motivation of increased efforts in research directions for model order reduction (MOR), where low dimensional substitute models are created based on high dimensional and therefore computationally expensive simulation models. In particular, Reduced Basis (RB) methods [13, 14] are increasingly popular methods for complexity reduction in problems, where parametrized partial differential equations (PDEs) or dynamical systems are to be solved repeatedly for varying parameters $\boldsymbol{\mu} \in \mathcal{P}$ from some compact parameter domain $\mathcal{P} \subset \mathbb{R}^p$. Examples for such simulation scenarios are multi-query or real-time settings, such as design, control, optimization, or inverse modelling based on PDEs. Instead of repeated computation of expensive detailed solutions $u(\boldsymbol{\mu}) \in X$ from a high dimensional space X for varying parameter vector $\boldsymbol{\mu}$, a problem-specific, typically low dimensional, subspace $X_N \subset X$ is constructed in a preprocessing step, which approximates the solution under parameter changes. Based on this reduced basis space, a reduced model is devised by Galerkin projection which inexpensively computes approximations $u_N(\boldsymbol{\mu}) \in X_N$ of the unknown solution for any (and typically many) new choices of the parameter. We refrain from further details on RB-methods for parametrized partial differential equations but refer to [14] for an overview for elliptic problems, or [10] for evolution equations.

Recently, the reduced basis approach was also formulated in the context of dynamical systems [11] of the form

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) &= \mathbf{A}(\boldsymbol{\mu})\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\mu})\mathbf{u}(t), \\ \mathbf{x}(0) &= \mathbf{x}_0(\boldsymbol{\mu}) \end{aligned}$$

with parameter dependent system matrices $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{n\times n}, \mathbf{B}(\boldsymbol{\mu}) \in \mathbb{R}^{n\times m}$ and initial values $\mathbf{x}_0(\boldsymbol{\mu}) \in \mathbb{R}^n$ on the time interval $t \in [0, T]$. The expensive solutions $u(\boldsymbol{\mu}) \in X$ are given by the continuous trajectories, e.g. $u(\boldsymbol{\mu}) := \mathbf{x}(\cdot; \boldsymbol{\mu}) \in X = C([0, T], \mathbb{R}^n)$. A reduced basis Φ can be identified with a pair of biorthogonal projection matrices $(\mathbf{V}, \mathbf{W}) \in \mathbb{R}^{n\times N}, N \ll n, \mathbf{W}^T\mathbf{V} = \mathbf{I}_N$ such that the reduced system is given as

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{x}}(t) &= \hat{\mathbf{A}}(\boldsymbol{\mu})\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}(\boldsymbol{\mu})\mathbf{u}(t), \\ \hat{\mathbf{x}}(0) &= \hat{\mathbf{x}}_0(\boldsymbol{\mu}) \end{aligned}$$

with reduced matrices $\hat{\mathbf{A}}(\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}$, $\hat{\mathbf{B}}(\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{B}(\boldsymbol{\mu})$ and initial data $\hat{\mathbf{x}}_0(\boldsymbol{\mu}) = \mathbf{W}^T x_0(\boldsymbol{\mu})$. The inexpensive approximations $u_N(\boldsymbol{\mu}) \in X_N$ are thus given as the reduced trajectories $u_N(\boldsymbol{\mu}) := \mathbf{V}\hat{\mathbf{x}}(\cdot;\boldsymbol{\mu}) \in X_N \subset X$. See [11] for an extension by an output equation. Given an arbitrary positive definite symmetric matrix $\mathbf{G} \in \mathbb{R}^{n\times n}$, which defines an inner product on state space by $\langle \mathbf{x}_1, \mathbf{x}_2\rangle_{\mathbf{G}} = \mathbf{x}_1^T\mathbf{G}\mathbf{x}_2$, and a computable upper bound $C_1(\boldsymbol{\mu}) \geq \sup_t \|\exp(\mathbf{A}(\boldsymbol{\mu})t)\|_{\mathbf{G}}$ an a-posteriori error bound is derived as $\sup_{t\in[0,T]} \|\mathbf{x}(t) - \mathbf{V}\hat{\mathbf{x}}(t)\|_{\mathbf{G}} \leq \Delta(\boldsymbol{\mu}, \Phi)$ with a-posteriori error estimator

$$\Delta(\boldsymbol{\mu}, \Phi) := C_1(\boldsymbol{\mu}) \left(\|\mathbf{e}(0;\boldsymbol{\mu})\|_{\mathbf{G}} + \int_0^T \|\mathbf{R}(\tau;\boldsymbol{\mu})\|_{\mathbf{G}}\, d\tau \right).$$

The initial error is thereby defined as $\mathbf{e}(0;\boldsymbol{\mu}) := \mathbf{x}(0;\boldsymbol{\mu}) - \mathbf{V}\hat{\mathbf{x}}(0,\boldsymbol{\mu})$ and the residual is given as $\mathbf{R}(t;\boldsymbol{\mu}) := \mathbf{A}(t;\boldsymbol{\mu})\mathbf{V}\hat{\mathbf{x}}(t;\boldsymbol{\mu}) + \mathbf{B}(t;\boldsymbol{\mu})\mathbf{u}(t) - \mathbf{V}\frac{d}{dt}\hat{\mathbf{x}}(t;\boldsymbol{\mu})$. All of these parameter dependent quantities can be computed efficiently during the reduced simulation by a proper offline-online decomposition as described in [11].

The crucial ingredient for all reduced basis methods is the choice of a *reduced basis* Φ the span of which yields the reduced basis space X_N. Methods for reduced basis construction typically are based on *snapshots*, i.e. $\{u(\boldsymbol{\mu}_i)\}_{i=1}^N$ for certain selected parameter vectors $\{\boldsymbol{\mu}_i\}_{i=1}^N$. Such snapshots can be used directly as basis vectors (after recommended orthonormalization) giving a Lagrangian Reduced Basis. Another possibility is to apply Proper Orthogonal Decomposition (POD) techniques as a data compression step for large sets of snapshots (in particular for time-sequences). In certain cases, the selection of snapshots is possible by solving an optimization problem, c.f. [2] for optimal parameter selection or [17] for an optimal choice of snapshots in time. In some simple cases optimal snapshot locations are known a-priori [13]. These approaches though are not yet available for more complex scenarios, which is the reason, why general-purpose algorithmical basis generation procedures need to be devised. In case of availability of a-posteriori error estimators, a well established approach is the *greedy* procedure [16, 7, 13], which accumulatively determines snapshots based on a (typically large) set of training parameters M_{train}. We will recall details in Sec. 2. An extension of this approach for time-dependent problems is the *POD-greedy* procedure introduced in [10] and successfully used in further applications [12].

In this presentation we propose extensions of these greedy and POD-greedy procedures, which are based on adaptive grids in parameter space, steered by a-posteriori error estimators or other indicators. The first approach is based on adaptive training set extension, which results in a reduction of the basis generation time (offline). The second approach gives explicit control of the reduced simulation time (online) by generating multiple bases with limited dimensionality on adaptive partitions of the parameter domain.

We briefly mention several further existing approaches, which are related to our proposals. The idea of a multistage greedy algorithm can be found in [15], which makes use of a sequential decomposition of a large training set. We will comment on this in Sec. 3. Further, the idea of parameter domain partition was inspired by similar approaches for stationary elliptic [4] and instationary parabolic approaches [5, 3]. In contrast to these references, which use Voronoi-type bisections leading to unstructured meshes with possibly degenerated elements, we apply adaptive hexaedral meshes of arbitrary dimension.

The structure of the presentation is as follows. In Sec. 2 we introduce the basic notation and briefly recall the meanwhile standard approaches for parameter selection in reduced basis generation. Section 3 presents an approach for adaptive extension of the training set and Sec. 4 a multiple bases approach using an adaptive parameter space partitioning. Experiments in Sec. 5 investigate the benefits of the training set extension and parameter domain partitioning separately and in a combined fashion. We conclude in Sec. 6 with a summary and comments on possible future improvements.

2 Notation

We introduce the (few) required notation:

- $\mathcal{P} \subset \mathbb{R}^p$ compact parameter domain; without loss of generality a hypercube, as any compact parameter set can be covered by such.

- $\mathcal{M}$ a non-conform adaptive hexaedral grid in the p-dimensional parameter domain with sets of leaf-elements $\mathcal{E}(\mathcal{M})$ and vertices $V(\mathcal{M})$. The centroid of an element $e \in \mathcal{E}(\mathcal{M})$ will be denoted with $c(e)$.

- X space of *detailed solutions*, this can be $X = H_0^1(\Omega)$ as a solution space of elliptic PDEs, or $X = C([0,T], \mathbb{R}^n)$ as the space of continuous trajectories in state space $\mathbb{R}^n$ of a dynamical system, etc.

- A *detailed solution* scheme, which can produce a *detailed solution* $u(\boldsymbol{\mu}) \in X$ for any admissible parameter $\boldsymbol{\mu} \in \mathcal{P}$.

- $\Phi = \{\varphi_1, \ldots, \varphi_{N(\Phi)}\}$ a *reduced basis* of size $N(\Phi)$, i.e. $N(\Phi)$ being the *reduced dimension.*

- $X_N \subset X$ the *reduced solution space* based on Φ, i.e. $X_N := \text{span}(\Phi)$ for stationary problems, or $X_N := C([0,T], \text{span}(\Phi))$ for time-dependent problems.

- a *reduced solution* scheme, which can produce a *reduced solution* $u_N(\boldsymbol{\mu}) \in X_N$ for any admissible parameter $\boldsymbol{\mu} \in \mathcal{P}$.

- $\Delta_N(\boldsymbol{\mu})$ an *error indicator*, measuring the error between the detailed and reduced solutions, or between outputs of these, i.e. $\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\| \leq \Delta_N(\boldsymbol{\mu})$ for suitable norm on X.

In particular, the error indicators $\Delta_N(\boldsymbol{\mu})$ can be any *a-posteriori error estimators* used in RB-methods [14, 10]. If such estimators are not available, also the true error can be used, in which case, the evaluation of this quantity is expensive as it involves the computation of a detailed solution, but still may be feasible in view of our adaptive approach. Further, we deliberately do not differ between stationary or instationary problems. As the reduced solutions and error estimators depend on the reduced basis that is used, we occasionally emphasize this dependency by the basis as additional argument $u_N(\boldsymbol{\mu}, \Phi), \Delta_N(\boldsymbol{\mu}, \Phi)$.

The standard *greedy* procedure [16, 7, 13] for the construction of a reduced basis Φ is based on a finite training set of parameters $M_{train} \subset \mathcal{P}$, a given desired error tolerance $\varepsilon_{tol} > 0$ and optionally an initial choice of basis Φ_0, which is to be extended. It is an accumulative basis construction procedure. It consecutively determines the $\boldsymbol{\mu}^* \in M_{train}$ that is worst resolved with the current reduced basis (as measured by the chosen error indicator $\Delta(\boldsymbol{\mu}, \Phi)$), performs a detailed simulation $u(\boldsymbol{\mu}^*)$ and uses this for extension of the basis Φ, until the error over M_{train} is less than ε_{tol}. For stationary problems, the step of basis extension is a simple inclusion of the solution snapshot $u(\boldsymbol{\mu})$ into the basis. For instationary problems, this extension-step must extract a few vectors from the detailed solution trajectory. This can either be based on residual increment criteria [6] or on orthogonal projections of the trajectory combined with temporal compression techniques [10]. The latter in particular turned out to be quite successful and is meanwhile standard and denoted *POD-greedy* procedure.

Despite the wide applicability of the (POD-)greedy algorithm, some problems remain, which will be adressed in the subsequent exposition:

1. Overfitting: In case of too small training set, the error on M_{train} may be nicely decreasing or can ideally be reduced to 0 with growing basis dimension, but the error for independent test parameters remains large.

2. Training times: In case of too large training set, the training time can be exorbitant, in particular in instationary scenarios without or with merely expensive error estimators.

3. Basis size: The (POD-)greedy algorithm will produce a basis satisfying the desired error threshold ε_{tol} on the training set. But the resulting basis size may be much too large, resulting in too high online simulation times.

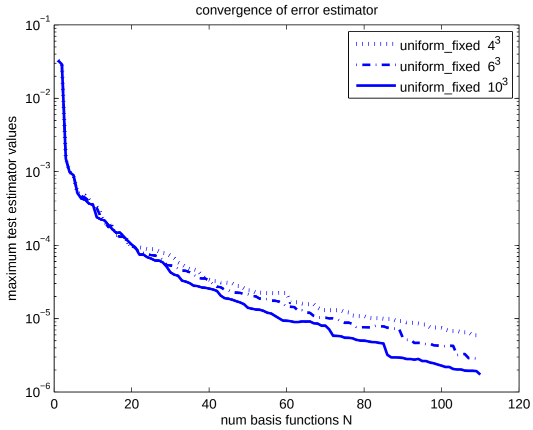

Figure 1: Illustration of overfitting for the model from [10]. The maximum error estimator on a random test set is plotted over the dimension N of the reduced basis space for the *POD-greedy* algorithm with a fixed training set corresponding to the vertices of a uniform Cartesian parameter grid with 4^3, 6^3, and 10^3 nodes.

3 Adaptive Training Set Extension

We now elaborate on a method that has been briefly explained in [9], adressing the first two issues raised above: "overfitting" and "training times". Fig. 1 gives some typical results of the (POD-)greedy basis generation procedure with the model problem from [10]. In particular, we choose 3 different sizes of (fixed) training parameter sets, generate bases with the POD-greedy procedure and monitor the maximum test error estimator over an independently drawn test parameter set. Each curve corresponds to one of the training sets. We see that in these cases the convergence rate of the test error estimator breaks down if N increases, while (not shown) the training error estimator is nicely decreasing with growing basis size. This indicates, that the (POD-)greedy algorithm tends to overfit if the training set is kept fixed and is too small.

On the other hand, if larger training sets are chosen, the breakdown of the convergence rate can be shifted towards larger values of N. Unfortunately, such a procedure usually comes with an immense increase in computational costs.

The underlying reason for both overfitting and too long training times is the unknown ideal size and distribution of the training set. Hence, the main component of our first approach is an adaptive training set extension procedure to adapt the number and location of the training parameters to the given problem.

In detail, we propose an extension of the (POD-)greedy algorithm by *early stopping*, which is a technique used in machine learning to prevent overfitting of iterative learning procedures. In particular, we enable the (POD-)greedy algorithm with additional input arguments as indicated in Fig. 2: an additional validation set M_{val} of parameters, an additional tolerance level ρ_{tol} and a maximum target basis size N_{max}. Note that by setting $\rho_{tol} = N_{max} = \infty$ the algorithm is the traditional (POD-)greedy procedure. The validation set M_{val} is used to monitor an additional validation error of the current basis. If the ratio of the validation error and the training error exceeds the tolerance $\rho_{tol} > 0$, we conclude that the basis construction process is overfitting on the training set and the early stopping of the (POD-)greedy procedure is initiated. Such cases of

```
EarlyStoppingGreedy($\Phi_0, M_{train}, \varepsilon_{tol}, M_{val}, \rho_{tol}, N_{max}$)
$\Phi := \Phi_0$
repeat
        $\boldsymbol{\mu}^* := \arg\max_{\boldsymbol{\mu} \in M_{train}} \Delta(\boldsymbol{\mu}, \Phi)$
        if $\Delta(\boldsymbol{\mu}^*) > \varepsilon_{tol}$
          then
                $\varphi :=$ ONBasisExt($u(\boldsymbol{\mu}^*), \Phi$)
                $\Phi := \Phi \cup \{\varphi\}$
        $\varepsilon := \max_{\boldsymbol{\mu} \in M_{train}} \Delta(\boldsymbol{\mu}, \Phi)$
        $\rho := \max_{\boldsymbol{\mu} \in M_{val}} \Delta(\boldsymbol{\mu}, \Phi)/\varepsilon$
  until $\varepsilon \leq \varepsilon_{tol}$ or $\rho \geq \rho_{tol}$ or $|\Phi| \geq N_{max}$
return $\Phi, \varepsilon$
```

Figure 2: The early-stopping (POD-)greedy search algorithm, for $\rho_{tol} = \infty, N_{max} = \infty$ recovering the standard (POD-)greedy procedure.

```
AdaptiveTrainExtension($\Phi_0, \mathcal{M}_0, \varepsilon_{tol}, M_{val}, \rho_{tol}, N_{max}$)
$\Phi := \Phi_0, \mathcal{M} := \mathcal{M}_0$
repeat
        $M_{train} := V(\mathcal{M})$
        $[\Phi, \varepsilon] :=$ EarlyStoppingGreedy($\Phi, M_{train}, \varepsilon_{tol}, M_{val}, \rho_{tol}, N_{max}$)
        if $\varepsilon > \varepsilon_{tol}$
          then
                $\boldsymbol{\eta} =$ ElementIndicators($\mathcal{M}, \Phi, \varepsilon$)
                $\mathcal{M} :=$ Mark($\mathcal{M}, \boldsymbol{\eta}$)
                $\mathcal{M} :=$ Refine($\mathcal{M}$)
  until $\varepsilon \leq \varepsilon_{tol}$ or $|\Phi| \geq N_{max}$
return $\Phi, \varepsilon$
```

Figure 3: The adaptive training set extension procedure based on an adaptive grid in parameter space.

detected overfitting indicate that M_{train} is too small for the desired model accuracy. For use in later approaches, the early stopping can also be induced by further criteria, in particular the basis construction procedure is stopped, if the maximum number of basis vectors N_{max} is exceeded. However, in this section we set $N_{max} = \infty$ and hence do not pose a bound on the basis size $|\Phi|$. The role of N_{max} will be crucial in the next section.

Our adaptive approach illustrated in Fig. 3 is based on an adaptive grid $\mathcal{M}$ covering the parameter set, i.e. $\mathcal{P} = \bigcup_{e \in \mathcal{E}(\mathcal{M})} e$, the vertices of which are taken as training set of the (POD-)greedy algorithm, $M_{train} := V(\mathcal{M})$. In case of detected overfitting by early stopping, we conclude a necessary refinement of the parameter grid. In the spirit of local grid adaptation in finite element methods, element indicators $\eta(e)$ are computed, which are related to the model-error on these parameter cells. A marking and refinement strategy results in uniform or local grid refinement. A subsequent restart of the greedy search over the now extended set of grid-vertices is performed, using the previously obtained basis as new initial basis, until the desired accuracy ε_{tol} is obtained. This completes the description of the *adaptive training set extension procedure* for the case of uniform global refinement. For the case of local refinement, the definition of the element indicators and the refinement rule need to be specified. We first define preliminary element indicators $\tilde{\eta}(e)$ for elements $e \in \mathcal{E}(\mathcal{M})$ of the parameter mesh by taking the maximum of the error estimator values in the vertices $\boldsymbol{\mu} \in V(e)$ and the barycenter $\boldsymbol{\mu} = c(e)$ of an element e, i.e.

$$\tilde{\eta}(e) := \left(\max_{\boldsymbol{\mu} \in V(e) \cup \{c(e)\}} \Delta(\boldsymbol{\mu}, \Phi) \right).$$

As it is well known for adaptive methods, such an indicator may have problems to detect local maxima of the error in cases where the starting parameter mesh is too coarse to resolve some detailed structures. It may happen, that the error indicator of a coarse cell is small or zero, and hence, the element is never refined in the refinement process. In order to circumvent such stalling problems we add some penalization term, which penalizes elements with long non-refinement history. We therefore define the final element indicators $\eta(e)$ as

$$\eta(e) := \gamma(e)s(e) + \tilde{\eta}(e)/\varepsilon,$$

where $\gamma(e) > 0$ denotes a weighting parameter depending on the local mesh size, $s(e)$ counts the number of precedent refinement steps that did not lead to a refinement of element e and ε is the maximum error estimator on $V(\mathcal{M})$. Thus, elements that are not detected by the point evaluation of the estimator, are penalized from one refinement step to the next, which asymptotically leads to a refinement of all elements. In each refinement step a fixed fraction $\Theta \in (0, 1]$ of the elements are refined, where the elements with the highest estimator value $\eta(e)$ are chosen. Experiments with this adaptive training set extension algorithm are presented in Section 5.

We conclude this section with a comment on the relation to the multistage greedy used in [15]. Similar to our approach the training times are reduced, as a basis on a coarser training set can already be sufficient for a finer set or ultimatively the complete parameter space. The approach, however, does not offer a local adaptation of the training set or means for overfitting prevention.

4 Adaptive Parameter Domain Partition

We now turn to point (3), "basis size", listed in Sec. 2. The following problem is frequently observed: For high dimensional or extensive parameter domains – even more expressed in instationary problems with large time-intervals – the solution variations can be very large. Of course, the (POD-)greedy algorithm will produce a global basis with the desired accuracy ε_{tol}. But this may only be obtained by a large reduced basis, e.g. size $N = 1000 - 10000$. This basis size may be prohibitive, as the online simulation of the reduced model requires operations with *full* matrices of this size. This may infer higher online computational cost than the original detailed model based on possibly much larger but usually *sparse* matrices. Hence, in addition to a guaranteed accuracy target ε_{tol}, a basis generation algorithm also should allow a limitation of the online-complexity. This can be obtained by directly limiting the reduced basis size by an upper bound N_{max}.

The key for realizing simultaneously accuracy and limited online-complexity, is a partition of the parameter space into several subdomains with small reduced bases for each of the subdomains. The method is a slight modification of similar approaches [4, 5, 3], now using structured meshes.

Being agnoscent of the future parameters, multiple reduced bases for all subdomains must be generated in the offline phase. Then, in the online phase the correct basis is selected for any given new parameter, and the reduced simulation can be performed. The partition of the parameter space can simply be fixed a-priori and the standard (POD-)greedy algorithm be run on each part. This may give smaller bases per subdomain, but by re-applying the above argumentation, the bases still can happen to be too large as we do not have a size control. Therefore, we now propose an adaptive partition approach for sub-dividing the parameter domain based on adaptive grids in the parameter domain.

The *adaptive parameter domain partition* procedure is now as follows, c.f. Fig. 4: Given a target accuracy ε_{tol} and maximum basis size N_{max}, we start with a coarse grid $\mathcal{M}$, which defines an initial parameter domain partition by its leaf elements. For each leaf element of the grid we initiate a standard (POD-)greedy (or any other) basis construction process. Now, an early stopping on the current element/subdomain is induced if N_{max} is exceeded. If this happens, it indicates that the corresponding parameter subdomain inhibits a too complex solution variety, which cannot be

```
AdaptiveParamPartition($\mathcal{M}_0, \varepsilon_{tol}, N_{max}$)
$\mathcal{M} := \mathcal{M}_0, \Phi(e) := \emptyset$ for $e \in \mathcal{E}(\mathcal{M})$
repeat
        for $e \in \mathcal{E}(\mathcal{M})$ with $\Phi(e) = \emptyset$
        do $\Phi_0 :=$ InitBasis($e$)
            $M_{train} :=$ Mtrain($e$)
            $\eta(e) := 0$
            $[\Phi(e), \varepsilon(e)] :=$ EarlyStoppingGreedy($\Phi, M_{train}, \varepsilon_{tol}, \emptyset, \infty, N_{max}$)
            if $\varepsilon(e) > \varepsilon_{tol}$
              then $\eta(e) := 1, \Phi(e) := \emptyset$
        $\eta_{max} := \max_{e \in \mathcal{E}(\mathcal{M})} \eta(e)$
        if $\eta_{max} > 0$
          then $\mathcal{M} :=$ Mark($\mathcal{M}, \boldsymbol{\eta}$)
               $\mathcal{M} :=$ Refine($\mathcal{M}$)
  until $\eta_{max} = 0$
return $\mathcal{M}, \{\Phi(e), \varepsilon(e)\}_{e \in \mathcal{E}(\mathcal{M})}$
```

Figure 4: The adaptive parameter domain decomposition procedure producing an adaptive grid in parameter domain and corresponding sequence of reduced bases.

covered by a small basis. Consequently the element is refined into several parameter subdomains and the basis generation per subdomain is restarted. The algorithm ends with possibly many subdomains, but each subdomain basis satisfying both the *prescribed accuracy and the basis size constraints*. Hence, we obtain simultaneous control over the online accuracy and online complexity by accepting higher offline computation cost and offline data storage.

We emphasize that this parameter domain partition approach is not related to the training set extension approach of the previous section, although both can be beneficially combined, as will be demonstrated later.

Overall, compared to a (POD-)greedy procedure on the complete parameter domain, the approach is characterized by increased offline computation and storage requirements, as now many bases must be computed and stored and several computations are discarded during the refinement process. However, we obtain control over the online complexity. Note that, as also remarked in [5, 3], for time-dependent problems ε_{tol} and N_{max} cannot be independently chosen arbitrary small. For a given $\varepsilon_{tol} > 0$ a POD of a solution trajectory for any fixed $\boldsymbol{\mu}^* \in \mathcal{P}$ will result in a required basis size of $N_{POD}(\boldsymbol{\mu}^*, \varepsilon_{tol}) \in \mathbb{N}$ basis vectors for approximating the trajectory with accurary ε_{tol}. Then, the requirement of choosing $N_{max} > \sup_{\boldsymbol{\mu} \in \mathcal{P}} N_{POD}(\boldsymbol{\mu}, \varepsilon_{tol})$ will imply that a sufficiently refined mesh $\mathcal{M}$ will result in a partition with bases satisfying the accuracy requirement. On the contrary, if N_{max} is fixed and too small, the refinement process will in general not terminate with a set of bases satisfying the accuracy constraint.

We remark, that there are some computational methods for considerably speeding up the offline basis construction process. First, the calculation of the detailed solutions, can be beneficially cached during the basis-generation of the different subdomains. Due to many shared edges/faces, basis generation on different subdomains requires computation of many identical trajectories. These can be computed once and stored for later use. A second aspect concerns the early stopping: The required computation of N_{max} basis vectors for deciding the failed accuracy and required refinement of a given element is very expensive and superfluous, as this basis is discarded in the refinement process. Hence, methods are desired, which allow to decide earlier, whether a refinement will be required. A simple approach consists of extrapolation of the training error convergence in the (POD-)greedy algorithm: Frequently an exponential decrease of the training error can be observed in the (POD-)greedy procedure. Hence, a suitable extrapolation of the error convergence curve towards N_{max} can be peformed during basis construction. By this, at an early stage $N << N_{max}$ the decision of failing accuracy and required refinement could be taken.

We conclude with some comments on comparison with the *hp-adaptive RB approach* from [4, 5, 3]. Conceptionally the approaches are very similar and we do not expect large practical differences. We only see some conceptional differences. We have a larger grid-management overhead compared to the binary bisections of the given references and cannot yet provide rigorous theoretical statements. In contrast, we maintain control over position, shape and size of the elements/subdomains, while arbitrary bisections can lead to degenerate elements. This property in particular simplifies a combination of our partition approach with the adaptive training set extension, as demonstrated in Sec. 5 below.

5 Experiments

We consider applications of the approaches on dynamic, i.e. instationary time-dependent problems based on the RB-methods for linear evolution schemes [10] and linear dynamical systems [11]. In the following two subsections we separately present results on the adaptive training set extension and adaptive parameter domain partition approaches. In the last subsection we illustrate experiments, where both approaches are combined. The experiments are based on our software-package RBmatlab. In particular the library provides an adaptive hypercube grid implementation for arbitrary dimensions $p \in \mathbb{N}$. The grid refinement is based on isotropic refinement, i.e. every hypercube is divided into 2^p children.

5.1 Adaptive Training Set Extension

We report in more detail on the results from numerical experiments of [8]. In particular, we apply the basis construction methods to a parametrized partial differential equation. We choose the model problem described in [10, Sec. 7] for our numerical experiments concerning the basis enrichment. The model represents an instationary advection-diffusion problem in the gas-diffusion layer of a fuel-cell. The velocity field is precomputed and the detailed discretization $u(\boldsymbol{\mu})$ is obtained by an implicit/explicit finite volume scheme of first order in space and time. The problem is characterized by a three-dimensional parameter space modeled by $\boldsymbol{\mu} = (c_{init}, \delta, \beta)^T \in \mathcal{P} = [0,1] \times [0, 5 \cdot 10^{-8}] \times [0,1]$. The first parameter c_{init} is the amplitude of a sinus-shaped initial-data distribution, δ is the global diffusion coefficient on the domain, and β, $1-\beta$ model the concentrations at the boundary of two gas-inlets. As error measure $\Delta(\boldsymbol{\mu}, \Phi)$ we choose an L^2 a-posteriori error estimator, for details we refer to [10].

We generate reduced bases with three approaches (not considering parameter domain partitioning). First, the vertices of a uniform fixed Cartesian grid are chosen as the training set M_{train} without any adaptive refinement (uniform-fixed). Second, a uniform Cartesian grid is used with global uniform refinement during the basis-construction (uniform-refined) and third, a Cartesian grid with local adaptive refinement (adaptive-refined) is applied. For an initial experiment we use a restricted two-dimensional parameter space $\mathcal{P} = [0,1] \times [0, 5 \cdot 10^{-8}]$ for the parameters $\boldsymbol{\mu} = (\beta, \delta)$ and fix $c_{init} = 1$. The resulting errors $\Delta(\boldsymbol{\mu}, \Phi)$ over the 2D parameter space are visualized logarithmically in Fig. 5. The fixed grid approach in a) clearly demonstrates overfitting with respect to the grid-vertices and error values varying over several orders of magnitude. In particular, it has a low error in the upper (high diffusivity) and very high errors in the lower part (low diffusivity) of the parameter domain. The uniformly refined approach in b) is slightly advantageous concerning these aspects. The locally refined grid in c) demonstrates considerable improvements concerning equal distribution of the error and the prevention of overfitting. A very notable aspect here, is the *agreement with physics*: From the problem setting, we know, that the parameters $\beta, 1-\beta$ are Dirichlet boundary values. Hence, largest error reduction is expected for low and high values of β. Similarly, for small values of the diffusivity parameter δ the solution structure is maintained over time. Hence, more difficult approximation is expected for such small diffusivities as is reflected by many required snapshots in these regions. These expectations are perfectly fullfilled by the locally refined grid.

Quantitative results are illustrated in Fig. 6 for the full 3D parameter space $\mathcal{P}$ and different initial grid sizes (vertex numbers ranging from 2^3 to 5^3). In Fig. 6 a) we illustrate the model-error (measured as maximum error estimator over a randomly generated test set) for the refined approaches and the fixed grid setting. We clearly see, that the model-error is reduced for the uniformly refined approach and even more expressed for the locally refined approach. The overfitting indeed seems to be prevented as is expected from the motivation and construction of the adaptive training set extension approach. Fig. 6 b) quantifies the improvement of the error-distribution by monitoring the ratio of maximum to minimum test error. We see, that the local refined approach guarantees a nice equi-distribution of the model-error over the parameter domain in contrast to the other approaches, where the minimum and maximum error values vary over some orders of magnitude. Finally, in Fig. 6 c) the maximum test error decrease of a) is related to the corresponding training time, i.e. the CPU time for the overall basis construction. With respect to this last criterion, the local refinement approach is consistently superior to the non-adaptive and adaptive approaches using uniform grids.

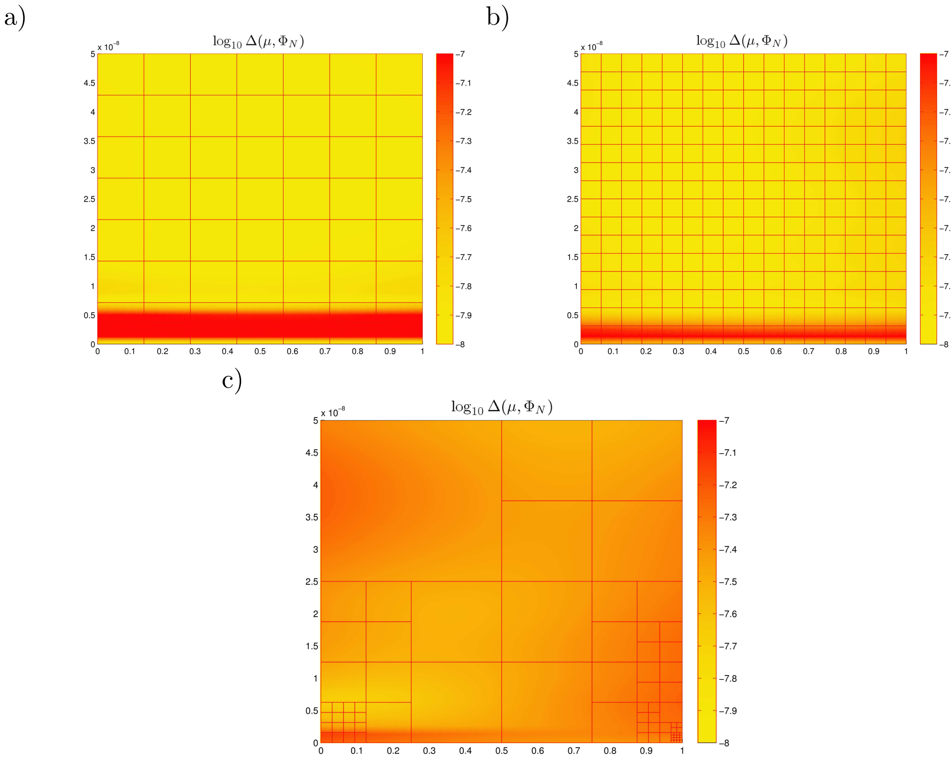

Figure 5: Demonstration of the overfitting phenomenon for a basis of size $N = 130$ in two-dimensional parameter space $\boldsymbol{\mu} = (\beta, \delta) \in \mathcal{P} = [0,1] \times [0, 5 \cdot 10^{-8}]$ with a) the uniform-fixed, b) the uniform-refined and c) the local adaptive-refined grid approach.

5.2 Adaptive Parameter Domain Partitioning

We apply the multiple bases approach with adaptive parameter domain partitioning (not considering training set adaptivity) to a dynamical system describing an advection problem taken from [11]. The problem is defined on a rectangular domain $\Omega = [0,2] \times [0,1]$ and time interval $[0,T], T = 1$. The initial condition $u_0(\cdot, \boldsymbol{\mu})$ is equal to zero on the whole domain Ω except for the region around the coordinate $x_C = (0.75, 1)$ on the top edge, where a cone with center x_C and amplitude $\mu_0 \in [0,1]$ is placed. The velocity field on the domain is chosen as a weighted superposition of two divergence free parabolic velocity fields in x_1- and x_2- direction. By the free weighting parameter $\mu_1 \in [0,1]$ the strength of the velocity field in x_1-direction can be set.

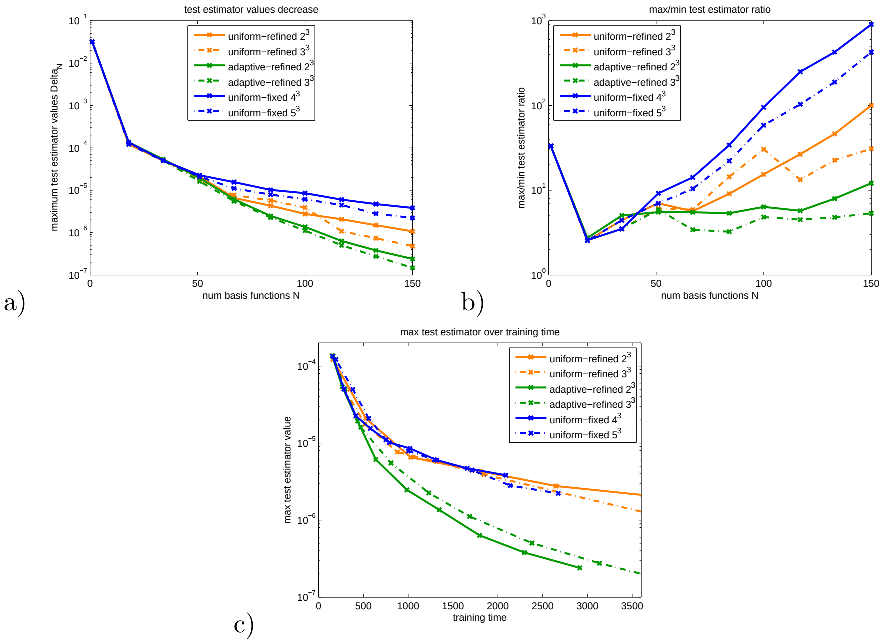

Figure 6: Quantitative comparison of the basis-generation approaches with respect to a) maximum test error, b) ratio of maximum to minimum test error and c) maximum test error over training time.

This allows to choose between a flow in x_2-direction only for $\mu_1 = 0$ and a mixture of both flow directions for $\mu_1 > 0$. For $\mu_1 = 0.5$ the velocity fields in both directions are equally strong. The overall parametrization is given by $\boldsymbol{\mu} = (\mu_0, \mu_1)^T \in \mathcal{P} = [0,1] \times [0,1]$.

For comparison, we generate reduced bases for the problem using three different approaches. The first approach is a standard (POD-)greedy approach using no parameter domain partitioning. The second approach uses a fixed parameter domain partitioning, dividing the parameter domain into four equally spaced subdomains and generating bases for each of the parts. The third approach consists of generating multiple bases on an adaptive parameter domain partitioning. For all three basis generation methods the training set M_{train} consists of the vertices of a uniform fixed Cartesian grid over the domain or subdomain. To ensure comparability, the target training error is $\varepsilon_{tol} = 1 \cdot 10^{-4}$ in all three cases.

In Fig. 7 we see the experimental results. The parameter domain $\mathcal{P} = [0,1] \times [0,1]$ is plotted with its partitions indicated by blue lines. The red dotted lines represent the grid providing the parameters in M_{train}. Blue dots mark the vertices used for basis generation. The numbers in the parameter grid subdomains show the resulting basis size of the (POD-)greedy algorithm while the colors indicate the estimated error on the domain. The standard (POD-)greedy algorithm in a) produces a rather large basis of 172 vectors to obtain a training set error smaller than ϵ_{tol}. The fixed parameter domain partition in b) reaches the same training set error with significantly smaller bases of sizes between 95 and 113 basis vectors. Furthermore, the colors in the plot show that in comparison to a) the maximal error on the domain is kept almost constant on a value of about 0.1. Finally, our adaptive parameter domain partition approach in c) produces an adaptive partitioning of the domain. The size of each of the bases in the partitions is limited by $N = 100$ while simultaneously the prescribed accuracy of the solution is reached. Furthermore, we see that the approximation error on the whole domain is smaller. In Fig. 7 we nicely see, that the physical meaning of parameters and their influence on the solution is reflected in the distribution of the basis sizes in b) and the repartition of subdomains in c). The higher the value of the parameter μ_1, the higher the velocity in x_1-direction, the more volatile is the evolution of the solution. Hence, more basis vectors are required for an approximation of the solution. The parameter μ_0 scales the

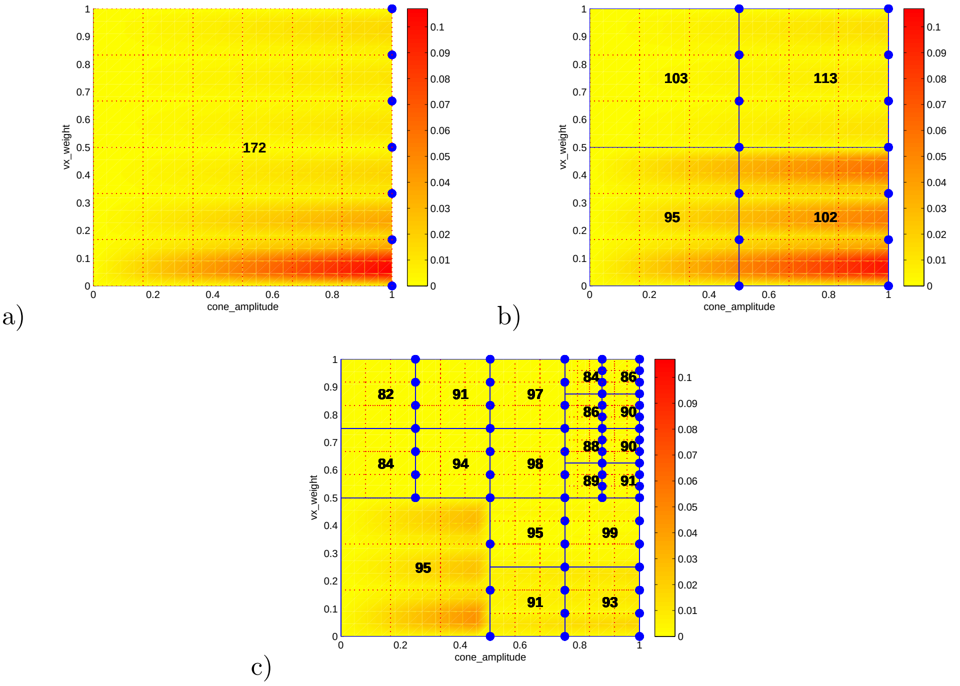

Figure 7: Demonstration of different basis generation approaches using the standard (POD-)greedy procedure on each subdomain. The colors indicate the test error over the parameter domain. a) No parameter domain partition ($\varepsilon < \varepsilon_{tol}$,$N$ very large), b) uniform parameter domain partition ($\varepsilon \leq \varepsilon_{tol}$,$N$ large), c) adaptive parameter domain partition ($\varepsilon \leq \varepsilon_{tol}$,$N \leq N_{max}$).

solution by varying the amplitude of the initial cone, hence larger maximum errors are obtained at the upper limit of the μ_0 interval. However, it is hard to compare the above reduced models, as the final training set accuracy, which is the stopping criterion for the training algorithms, is heavily depending on the training set size. This quantity is in general not predictive for the final test behaviour as argued earlier. Therefore, we now address the test error estimator as reasonable quality measure of a basis. For this, we randomly draw 500 parameter samples M_{test} from the parameter space and determine the maximum test error estimator $\varepsilon_{test,max} := \max_{\mu \in M_{test}} \Delta(\boldsymbol{\mu}, \Phi)$ and the average test error estimator $\varepsilon_{test,av} := \frac{1}{|M_{test}|} \sum_{\mu \in M_{test}} \Delta(\boldsymbol{\mu}, \Phi)$. The second relevant quality criterion is the reduced online simulation time, as this was the original motivation for the current approach. As the basis dimension now is varying for the parameters, the online simulation time similarly is non-constant. Therefore, we also compute the average online-simulation time over M_{test} as runtime measure. Fig. 8 shows the results of these measurements for various reduced bases. When demanding an accurate model over the entire parameter domain using a standard approach, we see that we have to pay the price of high online-simulation times due to large sizes of the bases. The online simulation time rises exponentially with increasing demands on the model-error. In contrast, using a multiple bases approach, the online-simulation time can be kept on a low level while providing approximations with a high accuracy.

5.3 Combination of Both Adaptive Approaches

As the building block of the parameter domain partition algorithm is a basis generation algorithm producing a basis on a subdomain of the parameter domain, we can also apply the adaptive training set extension algorithm of Sec. 3 instead of the standard (POD-)greedy. Hence, on each adaptively determined subdomain of $\mathcal{P}$ an adaptive training set extension is performed to construct a basis. The results are illustrated in Fig. 9. The adaptive training set extension algorithm successfully

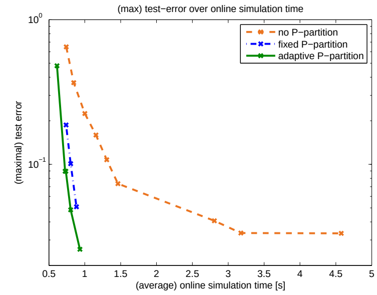

Figure 8: Comparison of different basis generation approaches regarding the maximal test error on the domain versus final (average) online-simulation time.

identifies regions of high model error and induces a refinement of the parameter domain partitions in these regions. The model error – especially in the lower part of the domain – is reduced significantly.

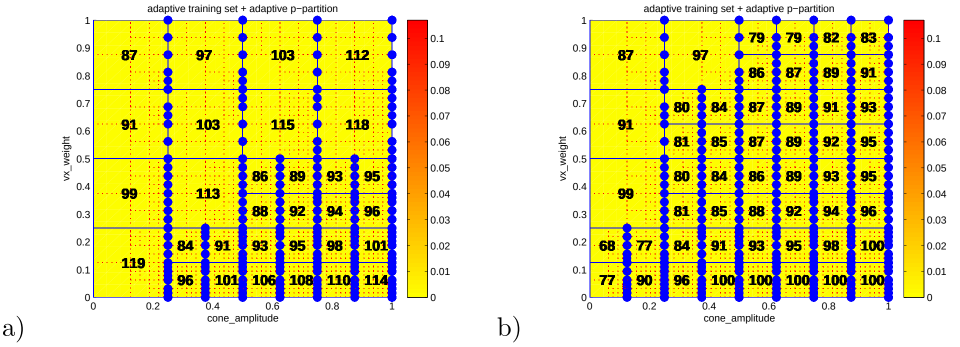

Figure 9: Demonstration of the adaptive parameter domain partition approach using the local adaptive training set extension method on each sub-domain. In a) the basis size was limited to $N_{max} = 120$ and in b) the limit was $N_{max} = 100$. We see that for the same training error of $\varepsilon_{tol} = 0.0001$ a smaller basis size limit leads to a finer partition of the parameter domain.

For this approach we conducted the same error analysis and online runtime measurement as in Sec. 5.2, which is based on 500 randomly chosen parameter samples. The results for the methods of Fig. 7 a),b),c) and 9 b) are given in Tab. 1. In the first three lines we present quantitative support for Fig. 8. We again see that even though the size of the basis is limited in the adaptive parameter domain partition approach, the maximal test error as well as the average test error on the domain are considerably smaller compared to the approach without partition. From the last line in Tab. 1 it can be seen that the combination of both adaptive approaches outperforms all other approaches by far. While the online simulation time of the combined approach is comparable to the pure parameter domain partititon approach, the error on the test set is reduced by two orders of magnitude.

Note that for this example, a non-isotropic adaptive grid would be beneficial. As the $\boldsymbol{\mu}$ samples with maximum error are always located at the right border of the parameter domain, we would only need refinement in this parameter direction. This would result in fewer number of subdomains and hence overall reduced offline computation time and storage. A further acceleration of the offline-phase could be achieved by applying the early stopping algorithm, which extrapolates the error convergence curve of the (POD-)greedy algorithm during basis-generation, as explained in Sec. 4.

	$\varepsilon_{test,max}$	$\varepsilon_{test,av}$	$\bar{t}_{sim,reduced}$
no parameter domain partition	$1.08 \cdot 10^{-1}$	$1.17 \cdot 10^{-2}$	1.31 s
fixed parameter domain partition	$1.01 \cdot 10^{-1}$	$1.44 \cdot 10^{-2}$	0.804 s
adaptive parameter domain partition	$4.85 \cdot 10^{-2}$	$4.62 \cdot 10^{-3}$	0.804 s
adaptive p-domain partition + training set	$2.06 \cdot 10^{-4}$	$7.90 \cdot 10^{-5}$	0.756 s

Table 1: Results of the test error calculation for the bases generated during the experiments. The table shows the maximal error on the parameter domain, the average error over the domain and the average time for an online simulation of the reduced model. The errors were calculated using a sample of 500 randomly chosen parameter vectors. The bases were generated with the same training error $\varepsilon_{tol} = 0.0001$ and the adaptive bases had a limited size of $N_{max} = 100$.

6 Conclusions and Outlook

We addressed the task of reduced basis construction based on snapshots and presented two different approaches that use adaptive grids in the parameter domain. The approaches are applicable to time-dependent and stationary problems, for PDEs or state-space dynamical systems, as long as an error indicator is available. In particular, as error indicator both a-posteriori error estimators or true errors can be chosen. Furthermore, the error can be considered for the field-variable, the state-vector or even for output functionals.

The main aspect of the first approach, the adaptive training set extension procedure, is the “right guess” of the training set size and the location of its points. This is obtained by applying overfitting control and adaptive training set extension based on adaptive grids in the parameter space. The procedure prevents phenomena of standard methods, where overfitting for too small training sets and high training times for too large training sets can be observed. In comparison to fixed training set approaches, the adaptive training set extension produces reduced bases with better model accuracy and more uniform distribution of the model-error over the parameter space. The basis computation time for equal accuracy is reduced.

The second approach, the multiple bases approach on adaptive parameter domain partitions, is a method to handle large parameter domains, where single-basis approaches for the reduced model would not be feasible. The partition and computation of many separate bases clearly comes with increased offline computation and storage costs. However, by the rigorous control of the maximum basis size, an explicit mean for online simulation time control is obtained. In particular, the numerical experiments demonstrate that by comparing models with equal test error, the adaptive parameter domain partition approach indeed results in models with smaller online-runtime.

Overall, the adaptive training set extension procedure is a method yielding efficient *offline* economization of the basis generation, while the adaptive parameter domain partition approach guarantees an efficient *online* phase. Both methods can be combined to obtain the benefits of both approaches.

There are several perspectives for improvement of the proposed algorithms. First, the application to the choice of interpolation points in interpolatory MOR [1] approaches seems straightforward. Also there, the location and number of interpolation points is an open question, which can very likely be addressed and solved by adaptive approaches. The parameter space dimensions were still quite decent. Higher parameter dimensions may be solved by similar approaches using adaptive random point sets or sparse grids. The parameter domain partition approach allows several aspects for further development. One point would be to realize adaptive non-isotropic refinement of a parameter domain partition grid. By respecting different parameter ranges, different parameter influences, etc. problem specific directions for refinement can be identified. A further open issue is the redundancy in different subdomain bases: The single bases of different subdomains are developed independently. They may contain identical or similar basis vectors. This seems a clear possibility for memory savings by suitable shared basis-vectors or sub-bases. A further aspect of improvement is the reuse of old bases. In its generality, the parameter domain partition approach performs a restart of the basis generation procedure on refined elements. Thus, previously computed bases of coarser parameter subdomains are dismissed. Here we see room for improvement by reusing information throughout the refinement process.

7 Acknowledgement

We acknowledge funding by the Landesstiftung Baden Württemberg gGmbH and thank the German Research Foundation (DFG) for financial support within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

References

[1] U. Baur and P. Benner. Parametrische Modellreduktion mit dünnen Gittern. In *GMA-Fachausschuss 1.30, Modellbildung, Identifizierung und Simulation in der Automatisierungstechnik, Salzburg ISBN 978-3-9502451-3-4*, 2008.

[2] T. Bui-Thanh. *Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems*. PhD thesis, Massachusetts Institute of Technology, 2007.

[3] J. L. Eftang, D. Knezevic, A. T. Patera, and E. M. Ronquist. An hp certified reduced basis method for parametrized parabolic partial differential equations. Technical report, MIT, Cambridge, 2010. Submitted to MCMDS.

[4] J. L. Eftang, A. T. Patera, and E. M. Ronquist. An âhpâ certified reduced basis method for parametrized elliptic partial differential equations. Technical report, MIT, Cambridge, 2009. submitted to SISC.

[5] J. L. Eftang, A. T. Patera, and E. M. Ronquist. An âhpâ certified reduced basis method for parametrized parabolic partial differential equations. In *In Proceedings of ICOSAHOM 2009*, 2010.

[6] M.A. Grepl. *Reduced-basis Approximations and a Posteriori Error Estimation for Parabolic Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, May 2005.

[7] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN, Math. Model. Numer. Anal.*, 39(1):157–181, 2005.

[8] B. Haasdonk and M. Ohlberger. Basis construction for reduced basis methods by adaptive parameter grids. In P. Díez and K. Runesson, editors, *Proc. International Conference on Adaptive Modeling and Simulation, ADMOS 2007*. CIMNE, Barcelona, 2007.

[9] B. Haasdonk and M. Ohlberger. Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. In *Proc. 5th International Symposium on Finite Volumes for Complex Applications*, pages 471–478, 2008.

[10] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *M2AN, Math. Model. Numer. Anal.*, 42(2):277–302, 2008.

[11] B. Haasdonk and M. Ohlberger. Efficient reduced models for parametrized dynamical systems by offline/online decomposition. In *Proc. MATHMOD 2009, 6th Vienna International Conference on Mathematical Modelling*, 2009.

[12] D.J. Knezevic and A.T. Patera. A certified reduced basis method for the Fokker-Planck equation of dilute polymeric fluids: FENE dumbbells in extensional flow. Technical report, MIT, Cambridge, MA, 2009.

[13] A.T. Patera and G. Rozza. *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations.* MIT, 2007. Version 1.0, Copyright MIT 2006-2007, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering.

[14] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Meth. Eng.*, 15(3):229–275, 2008.

[15] S. Sen. Reduced basis approximations and a posteriori error estimation for many-parameter heat conduction problems. *Numerical Heat Transfer, Part B: Fundamentals*, 54(5):369–389, 2008.

[16] K. Veroy, C. Prud'homme, D. V. Rovas, and A. T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *In Proceedings of 16th AIAA computational fluid dynamics conference*, 2003. Paper 2003-3847.

[17] S. Volkwein and K. Kunisch. Optimal snapshot location for computing POD basis functions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 2010. to appear.

Bernard Haasdonk
Institute of Applied Analysis and Numerical Simulation, University of Stuttgart,
Pfaffenwaldring 57
70569 Stuttgart
Germany
E-Mail: `haasdonk@mathematik.uni-stuttgart.de`

Markus Dihlmann
Institute of Applied Analysis and Numerical Simulation, University of Stuttgart,
Pfaffenwaldring 57
70569 Stuttgart
Germany
E-Mail: `dihlmann@mathematik.uni-stuttgart.de`

Mario Ohlberger
Institute of Computational and Applied Mathematics, University of Münster,
Einsteinstr. 62,
48149 Münster,
Germany
E-Mail: `mario.ohlberger@math.uni-muenster.de`