Universität Stuttgart

SimTech
Cluster of Excellence

M. Dihlmann[a] · **S. Kaulmann**[a] · **B. Haasdonk**[a]

Online Reduced Basis Construction Procedure for Model Reduction of Parametrized Evolution Systems

Stuttgart, January 2012

[a]Institute of Applied Analysis and Numerical Simulation,
University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
{dihlmann, sven.kaulmann, haasdonk}@mathematik.uni-stuttgart.de.

Abstract The reduced basis method is a model reduction technique for parametrized PDEs. It approximates the parameter-dependent solutions in a low dimensional space constructed by "snapshots" of high dimensional discrete solutions for selected parameters. Usually this reduced space is built up in advance in a costly offline phase and is suited to approximate well all solutions for parameters stemming from a given parameter domain. For the case of time dependent problems we propose a new method which rapidly builds up a locally adapted approximation space for a given parameter by selection from and combination of a set of POD bases. Instead of generating a globally valid reduced space for the whole parameter domain we will show how to construct online a low dimensional approximation space which is more performant with respect to the online cost.

Online Reduced Basis Construction Procedure for Model Reduction of Parametrized Evolution Systems

Markus Dihlmann * **Sven Kaulmann** * **Bernard Haasdonk** *

* *Institute of Applied Analysis and Numerical Simulation,*
University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
{dihlmann, sven.kaulmann, haasdonk}@mathematik.uni-stuttgart.de.

Abstract: The reduced basis method is a model reduction technique for parametrized PDEs. It approximates the parameter-dependent solutions in a low dimensional space constructed by "snapshots" of high dimensional discrete solutions for selected parameters. Usually this reduced space is built up in advance in a costly offline phase and is suited to approximate well all solutions for parameters stemming from a given parameter domain. For the case of time dependent problems we propose a new method which rapidly builds up a locally adapted approximation space for a given parameter by selection from and combination of a set of POD bases. Instead of generating a globally valid reduced space for the whole parameter domain we will show how to construct online a low dimensional approximation space which is more performant with respect to the online cost.

1. INTRODUCTION

Simulations of complex parametrized PDEs often require high dimensional discrete models due to the need of a high space resolution of the discretization. As a consequence these models are not suited for real-time applications or multi-query tasks like parameter optimization, statistical analysis or inverse problems because the calculation of solutions for many different parameters can take an excessive amount of time. This is the motivation for the development and the application of model reduction techniques for parametrized models.

The reduced basis method is such a model order reduction technique (see e.g. Rozza et al. (2008)). For any given parameter the parametrized problem is solved in a reduced space of low dimension. This low dimensional reduced basis space is constructed by "snapshots" of true solutions stemming from the original high dimensional space for selected parameters. Thereby, the reduced space is adapted to represent well all parameter-dependent solutions. However, if the problem depends on a wide range of parameters or if the variability of the solution with respect to the parameter is very high, it is possible that the reduced space requires a high number of basis functions in order to provide good global approximations. This is the case especially in evolution problems where an entire trajectory of solutions has to be approximated. There exist approaches to treat high parameter complexity in reduced basis methods as in Eftang et al. (2010, 2011); Haasdonk et al. (2011); Dihlmann et al. (2011). Yet, the offline time for basis generation can be very high in these approaches. Another difficulty is the fact that the control of the approximation error is not flexible during the online

phase. In cases where the approximation is not satisfactory one can not easily extend the reduced basis online.

We propose a new approach for the online construction of a reduced space in reduced basis methods for evolution problems. The approach is able to generate online an adapted low dimensional reduced space from a database of precomputed small reduced basis entities. In the online phase, for any given parameter, we enrich the reduced basis step by step via a proximity search in the parameter space until the desired approximation accuracy is fulfilled. This results in a locally adapted small reduced space and hence in a more efficient online simulation phase. Furthermore we are flexible in the control of the approximation error and can assure the fulfillment of an approximation accuracy requirement. Similar methods have been proposed in Amsallem and Farhat (2011) and Baur et al. (2011) in the setting of dynamical systems using a POD model order reduction. An approach that also makes use of single precomputed bases for individual parameters in the framework of Krylov-subspace methods is to be found in Lohmann and Eid (2009).

The paper is structured as follows: In Section 2 we begin with introducing the problem setting and briefly describing the framework of the reduced basis method. In Section 3 we present the offline and online procedure of the new approach. The method is demonstrated in a numerical experiment in Section 4 and we conclude in Section 5.

2. PROBLEM SETTING AND STANDARD REDUCED BASIS APPROACH

We consider the discretized general linear parameter-dependent evolution equation

$$\mathcal{L}^{\mathrm{Im}}_{h,\Delta t} u_h^{k+1}(\boldsymbol{\mu}) = \mathcal{L}^{\mathrm{Ex}}_{h,\Delta t} u_h^k(\boldsymbol{\mu}) + \Delta t b_h(t^k, \boldsymbol{\mu}) \qquad (1)$$

$$u_h^0(\boldsymbol{\mu}) = P_{X \to X_h}(u(\cdot, 0; \boldsymbol{\mu}))$$

with discrete solutions $u_h^k(\boldsymbol{\mu}) := u_h(t^k, \boldsymbol{\mu})$ from a discrete high dimensional function space X_h at time steps $t^k = k\Delta t$ with $1 \le k \le K$. Here

$$\mathcal{L}^{\mathrm{Im}}_{h,\Delta t} := \mathrm{Id} - \Delta t \mathcal{L}_{\mathrm{Im},h}(t^{k+1}, \boldsymbol{\mu})$$

$$\mathcal{L}^{\mathrm{Ex}}_{h,\Delta t} := \mathrm{Id} + \Delta t \mathcal{L}_{\mathrm{Ex},h}(t^k, \boldsymbol{\mu})$$

represent the implicit and explicit discrete operators respectively, while $\mathcal{L}_{Im,h}, \mathcal{L}_{Ex,h}$ are the spatial discrete operators obtained by a spatial discretization (e.g. Finite Volumes or Finite Elements). The parameter stems from a set of parameters $\boldsymbol{\mu} \in \mathcal{P} \subseteq \mathbb{R}^p$.

In order to realize an efficient offline/online splitting in the procedure, we assume parameter separability of the operators $\mathcal{L}_{\mathrm{Im},h}(t^k, \boldsymbol{\mu})$, $\mathcal{L}_{\mathrm{Ex},h}(t^k, \boldsymbol{\mu})$ and the right hand side $b_h(t^k, \boldsymbol{\mu})$ so that both can be expressed as linear combinations

$$\mathcal{L}_{\mathrm{Im},h}(t^k, \boldsymbol{\mu}) = \sum_{q=1}^{Q_{\mathcal{L}_{\mathrm{Im}}}} \Theta^q_{\mathcal{L}_{\mathrm{Im}}}(t^k, \boldsymbol{\mu}) \mathcal{L}^q_{\mathrm{Im},h} \qquad (2)$$

$$\mathcal{L}_{\mathrm{Ex},h}(t^k, \boldsymbol{\mu}) = \sum_{q=1}^{Q_{\mathcal{L}_{\mathrm{Ex}}}} \Theta^q_{\mathcal{L}_{\mathrm{Ex}}}(t^k, \boldsymbol{\mu}) \mathcal{L}^q_{\mathrm{Ex},h} \qquad (3)$$

$$b_h(t^k, \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta^q_b(t^k, \boldsymbol{\mu}) b_h^q \qquad (4)$$

of parameter- and time-dependent coefficients $\Theta^q_{\mathcal{L}_{\mathrm{Im}}}(t^k, \boldsymbol{\mu})$, $\Theta^q_{\mathcal{L}_{\mathrm{Ex}}}(t^k, \boldsymbol{\mu})$, $\Theta^q_b(t^k, \boldsymbol{\mu})$ and parameter- and time-independent components $\mathcal{L}^q_{Im,h}, \mathcal{L}^q_{Ex,h}, b_h^q$.

The reduced basis method for model reduction of evolution equations consists of four main ingredients:

(1) Galerkin projection: The high dimensional evolution scheme (1) is projected onto a reduced space $X_N = \mathrm{span}\Phi_N = \mathrm{span}\{\varphi_1, \ldots, \varphi_N\}$ which is spanned by basis vectors φ_i, $1 \le i \le N$ and which is of low dimension $N \ll \dim(X_h)$. Consequently the obtained reduced evolution scheme is of low dimension and can be solved rapidly to find the reduced solution $u_N(\boldsymbol{\mu}) = \{u_N^k(\boldsymbol{\mu}), 0 \le k \le K\} \in (X_N)^{K+1}$.

(2) A-posteriori error estimation: RB-methods provide a rigorous upper bound for the approximation error $\|u_h^k(\boldsymbol{\mu}) - u_N^k(\boldsymbol{\mu})\| \le \Delta^k_{\Phi_N}(\boldsymbol{\mu})$ which can be calculated rapidly during online simulations (see Grepl and Patera (2005); Haasdonk and Ohlberger (2008)).

(3) POD-Greedy sampling: The reduced space X_N is constructed offline using the POD-Greedy algorithm by successively adding POD-modes of solution trajectories for selected parameters to the global reduced basis (see e.g. Haasdonk and Ohlberger (2008); Knezevic and Patera (2009); Eftang et al. (2011)).

(4) Offline/Online decomposition: The computational procedure can be divided into a preparing offline phase and an $\mathcal{O}(\dim(X_h))$-independent online phase.

3. NEW APPROACH: RAPID ONLINE CONSTRUCTION OF A REDUCED SPACE

The POD-Greedy algorithm provides a reduced basis space which is (nearly) optimally suited to approximate the solution manifold over the whole parameter domain $\mathcal{P}$. In our approach we propose to construct online a reduced space which is suited to locally approximate the solution manifold in a small region around the parameter $\boldsymbol{\mu}^*$ for which we want to calculate $u_N(\boldsymbol{\mu}^*)$. As this is only a local approximation of the solution manifold around $\boldsymbol{\mu}^*$ the reduced space can be of very small dimension and consequently online simulations are more performant. We first show how to construct a dictionary of bases consisting of precomputed solutions. Based on this dictionary we will explain the online simulation procedure and present four different approaches to construct online a locally adapted reduced space.

3.1 Offline phase: Construction of the dictionary

We construct a "dictionary" of reduced bases

$$\mathcal{D} = \{b_e | 1 \le e \le E\} \qquad (5)$$

consisting of entities $b_e := (\Phi_e, \boldsymbol{\mu}_e)$ where the $\boldsymbol{\mu}_e \in \mathcal{M}_{\mathcal{D}}$ stem from a given set of parameters $\mathcal{M}_{\mathcal{D}}$ with $|\mathcal{M}_{\mathcal{D}}| = E$. The entity bases $\Phi_e = \{\varphi_{e,1}, \ldots, \varphi_{e,N_{max,e}}\}$ consist of basis functions $\varphi_{e,i} \in X_h$ and are obtained by solving the high dimensional evolution scheme (1) for all parameters $\boldsymbol{\mu}_e \in \mathcal{M}_{\mathcal{D}}$ and performing a POD

$$\Phi_e = \mathrm{POD}(u_h(\boldsymbol{\mu}_e), \varepsilon_{POD}) \qquad (6)$$

over the obtained trajectories

$$u_h(\boldsymbol{\mu}_e) := \{u_h^k(\boldsymbol{\mu}_e) | 0 \le k \le K\}.$$

The POD returns a POD space $\mathcal{X}_{e,\varepsilon_{POD}} = \mathrm{span}\Phi_e$ in two steps. First the optimality condition

$$\mathcal{X}_e^m = \arg \inf_{\substack{Y \subset \{u_h^k(\boldsymbol{\mu}_e), 0 \le k \le K\} \\ \dim(Y)=m}} \mathcal{R}(u_h(\boldsymbol{\mu}_e), Y) \qquad (7)$$

is fulfilled for $m = 1, ..., K$ where

$$\mathcal{R}(u_h(\boldsymbol{\mu}_e), Y) = \left(\frac{1}{K} \sum_{k=0}^K \inf_{y \in Y} \|u_h^k(\boldsymbol{\mu}_e) - y\|^2 \right)^{\frac{1}{2}}. \qquad (8)$$

Then $\mathrm{POD}(u_h(\boldsymbol{\mu}_e), \varepsilon_{POD})$ returns the basis Φ_e spanning the space $\mathcal{X}_{e,\varepsilon_{POD}} := \mathcal{X}_e^{m(\varepsilon_{POD})}$ where $m(\varepsilon_{POD})$ is the smallest natural where $\mathcal{R}(u_h(\boldsymbol{\mu}_e), \mathcal{X}_e^m) \le \varepsilon_{POD}$.

As later in the online phase we will solve a Galerkin-projected evolution scheme, we precompute the Gram matrix $\boldsymbol{G}$ and the projections of the parameter-independent components of the operators and the right hand side

$$\left(\boldsymbol{G}_{(e1,e2)} \right)_{i,j} = \langle \varphi_{e1,i}, \varphi_{e2,j} \rangle \qquad (9)$$

$$\left(\boldsymbol{L}^q_{\mathrm{Im}(e1,e2)} \right)_{j,i} = \left\langle \mathcal{L}^q_{\mathrm{Im},h} \varphi_{e_1,i}, \varphi_{e_2,j} \right\rangle \qquad (10)$$

$$\left(\boldsymbol{L}^q_{\mathrm{Ex}(e1,e2)} \right)_{j,i} = \left\langle \mathcal{L}^q_{\mathrm{Ex},h} \varphi_{e_1,i}, \varphi_{e_2,j} \right\rangle \qquad (11)$$

$$(\boldsymbol{b}_e^q)_i = \langle b_h^q, \varphi_{e,i} \rangle \qquad (12)$$

for all $e1, e2 = 1, \ldots, E$, $i = 1, \ldots, |\Phi_{e1}|$ and $j = 1, \ldots, |\Phi_{e2}|$. Thereby, we cover in advance the projection of the operators for all possible combinations of basis vectors.

3.2 Online Phase 1: Online construction algorithms for the localized reduced space

Based on the dictionary of bases $\mathcal{D}$ we propose four different methods for incremental online construction of parameter-dependent bases $\Phi(\boldsymbol{\mu}^*)$.

Initialization   Assume a parameter $\boldsymbol{\mu}^*$ to be given. For all four online construction methods, we initialize the online basis with the N_{init} nearest entity bases. That is: For an empty collection of entity indices $\mathcal{I}(\boldsymbol{\mu}^*) = \emptyset$ we repeat the following N_{init} times:

$$\bar{e} = \arg \min_{1 \le e \le E, e \notin \mathcal{I}(\boldsymbol{\mu}^*)} \| \boldsymbol{\mu}^* - \boldsymbol{\mu}_e \|_{\mathbb{R}^p}, \qquad (13)$$

$$\mathcal{I}(\boldsymbol{\mu}^*) = \mathcal{I}(\boldsymbol{\mu}^*) \cup \bar{e} \qquad (14)$$

and then set the initial basis $\overline{\Phi}(\boldsymbol{\mu}^*)$ as

$$\overline{\Phi}(\boldsymbol{\mu}^*) = \bigcup_{e \in \mathcal{I}(\boldsymbol{\mu}^*)} \Phi_e. \qquad (15)$$

Online POD   In general we can not guarantee linear independence of $\overline{\Phi}(\boldsymbol{\mu}^*)$, let alone orthogonality. Hence, we apply an online POD on the basis to ensure linear independence and also to ensure a good condition number of the arising reduced matrices. During this POD, the given basis is replaced by a linearly independent orthogonal set of functions spanning the same space as $\overline{\Phi}(\boldsymbol{\mu}^*)$ or a subspace: Given an online POD-tolerance $\varepsilon_{\text{POD}}^{\text{Online}}$, let

$$\{\lambda_1, \ldots, \lambda_{\bar{N}}\} \subset \mathbb{R}, \quad \{v_1, \ldots, v_{\bar{N}}\} \subset \mathbb{R}^{|\overline{\Phi}(\boldsymbol{\mu}^*)|}$$

be the set of all eigenvalues and matching eigenvectors of the Gram matrix $\boldsymbol{G}_{\overline{\Phi}}$ for the basis $\overline{\Phi}(\boldsymbol{\mu}^*)$. Without loss of generality we assume $\{\lambda_1, \ldots, \lambda_{\bar{N}}\}$ to be sorted in descending order. Here we used the notation $\boldsymbol{G}_{\boldsymbol{Z}}$ for the Gram matrix $\boldsymbol{G}_{\boldsymbol{Z}} \in \mathbb{R}^{M \times M}$, $(\boldsymbol{G}_{\boldsymbol{Z}})_{i,j} = \langle \zeta_i, \zeta_j \rangle$ of a given basis $Z = \{\zeta_1, \ldots, \zeta_M\}$.

Now, let N with

$$1 \le N \le \bar{N} := |\overline{\Phi}(\boldsymbol{\mu}^*)|$$

be the smallest natural number such that

$$\sum_{i=N}^{\bar{N}} \lambda_i \ge \varepsilon_{\text{POD}}^{\text{Online}} \quad \text{but} \quad \sum_{i=N+1}^{\bar{N}} \lambda_i < \varepsilon_{\text{POD}}^{\text{Online}}.$$

Then, the transition from the above-computed basis $\overline{\Phi}(\boldsymbol{\mu}^*)$ to the POD-basis $\Phi(\boldsymbol{\mu}^*)$ is represented by the matrix $C^{\mathcal{I}(\boldsymbol{\mu}^*)} \in \mathbb{R}^{\bar{N} \times N}$ with

$$C_{ij}^{\mathcal{I}(\boldsymbol{\mu}^*)} = \frac{v_{j,i}}{\sqrt{\lambda_j}}, \quad 1 \le i \le \bar{N}, 1 \le j \le N, \qquad (16)$$

where $v_{j,i}$ denotes the i-th entry of the eigenvector $v_j \in \mathbb{R}^{\bar{N}}$. Given this matrix $C^{\mathcal{I}(\boldsymbol{\mu}^*)}$ the final (POD-)basis $\Phi(\boldsymbol{\mu}^*)$ is given as:

$$\Phi(\boldsymbol{\mu}^*) = \{\varphi_1, \ldots, \varphi_N\} \qquad (17)$$

$$\varphi_k = \sum_{j=1}^{\bar{N}} C_{jk}^{\mathcal{I}(\boldsymbol{\mu}^*)} \overline{\varphi}_j, \qquad (18)$$

where we used the notation $\overline{\Phi}(\boldsymbol{\mu}^*) = \{\overline{\varphi}_1, \ldots, \overline{\varphi}_{\bar{N}}\}$.

In the following we use the rapidly evaluable a-posteriori error estimator $\Delta_{\Phi}^K(\boldsymbol{\mu})$ fulfilling

$$\sup_{0 \le k \le K} \| u_h^k(\boldsymbol{\mu}) - u_N^k(\boldsymbol{\mu}) \| \le \Delta_{\Phi}^K(\boldsymbol{\mu}). \qquad (19)$$

Here u_N^k is the approximation using the reduced basis Φ. Given this estimator, we check if the error tolerance $\Delta_{\Phi(\boldsymbol{\mu}^*)}^K(\boldsymbol{\mu}^*) \le \varepsilon_{tol}$ is fulfilled already. If so, we end the basis construction. If the error tolerance is not fulfilled yet, we start the basis enrichment.

Basis enrichment   In this step of the basis construction, we add additional functions to the given basis $\Phi(\boldsymbol{\mu}^*)$ until the desired error tolerance is reached. As in the initialization of the basis, we continue adding the "next nearest" entity b_e until the error tolerance is fulfilled. The distance between the parameter $\boldsymbol{\mu}^*$ and an entity b_e is not necessarily measured by the euclidean distance between parameters but by a more general distance function $d : \mathbb{R}^p \times \mathbb{N} \to \mathbb{R}_{\ge 0}$. Our enrichment algorithm is:
Repeat

$$\bar{e} = \arg \min_{1 \le e \le E, e \notin \mathcal{I}(\boldsymbol{\mu}^*)} d(\boldsymbol{\mu}^*, e), \qquad (20)$$

$$\mathcal{I}(\boldsymbol{\mu}^*) = \mathcal{I}(\boldsymbol{\mu}^*) \cup \bar{e}, \qquad (21)$$

$$\overline{\Phi}(\boldsymbol{\mu}^*) = \bigcup_{e \in \mathcal{I}(\boldsymbol{\mu}^*)} \Phi_e \qquad (22)$$

$$\Phi(\boldsymbol{\mu}^*) = \text{onlinePOD}(\overline{\Phi}(\boldsymbol{\mu}^*)) \qquad (23)$$

until $\Delta_{\Phi(\boldsymbol{\mu}^*)}^K(\boldsymbol{\mu}^*) \le \varepsilon_{tol}$.

It remains to state the possible choices for the distance function d. We will introduce four different distance functions: The Euclidean, Greedy-Extended, Greedy-Min and Greedy-Max functions.

Euclidean   Here we choose

$$d(\boldsymbol{\mu}^*, e) = \| \boldsymbol{\mu}^* - \boldsymbol{\mu}^e \|_{\mathbb{R}^p}. \qquad (24)$$

This distance function measures the distance of the given parameter $\boldsymbol{\mu}^*$ and a given entity b_e by the euclidean distance of $\boldsymbol{\mu}^*$ and the parameter $\boldsymbol{\mu}^e$ in b_e. The evaluation of this distance function should be very rapid.

Greedy-Extended   In this case the distance function d is given as

$$d(\boldsymbol{\mu}^*, e) = \Delta_{\Phi(\boldsymbol{\mu}^*) \cup \Phi_e}^K(\boldsymbol{\mu}^*), \qquad (25)$$

which means that we use the ability of the enlarged basis $\overline{\Phi}(\boldsymbol{\mu}^*) \cup \Phi_e$ to approximate the high dimensional solution for $\boldsymbol{\mu}^*$ as an indicator for the distance between $\boldsymbol{\mu}^*$ and b_e. The evaluation of this distance function might be a bit more costly as a reduced simulation and error estimator evaluation have to be performed.

Greedy-Min   Choosing

$$d(\boldsymbol{\mu}^*, e) = \Delta_{\Phi(\boldsymbol{\mu}^*)}^K(\boldsymbol{\mu}^e), \qquad (26)$$

the distance between $\boldsymbol{\mu}^*$ and b_e is quantified by the ability of the functions in $\overline{\Phi}(\boldsymbol{\mu}^*)$ to represent the solution to $\boldsymbol{\mu}^e$. This means that the distance between $\boldsymbol{\mu}^*$ and b_e is smaller the more the functions in $\overline{\Phi}(\boldsymbol{\mu}^*)$ resemble the high dimensional solution to $\boldsymbol{\mu}^e$.

Greedy-Max   The distance function

$$d(\boldsymbol{\mu}^*, e) = \left(\Delta_{\Phi(\boldsymbol{\mu}^*)}^K(\boldsymbol{\mu}^e) + 1 \right)^{-1}, \qquad (27)$$

performs the exact opposite of the Greedy-Min distance function: Here, an entity b_e is considered "closer" to $\boldsymbol{\mu}^*$ the less the high dimensional solution to $\boldsymbol{\mu}^e$ ressembles the functions in $\Phi(\boldsymbol{\mu}^*)$.

3.3 Online phase 2: Performing online simulations

In the online phase rapid solution approximations for the high dimensional evolution scheme (1) for a desired parameter $\boldsymbol{\mu}^*$ are performed. This is done using the reduced basis $\Phi(\boldsymbol{\mu}^*)$ constructed by one of the methods described above. We substitute the detailed solution $u_h^k(\boldsymbol{\mu})$ in (1) by its approximation $u_N^k(\boldsymbol{\mu}) = \sum_{i=1}^N a_i^k(\boldsymbol{\mu})\varphi_i$ and perform a Galerkin projection onto the reduced space $X_N = \mathrm{span}(\Phi(\boldsymbol{\mu}^*)) \subseteq X_h$. This leads to the low dimensional "reduced" evolution scheme

$$(\boldsymbol{G} - \Delta t \boldsymbol{L}_{\mathrm{Im}})\boldsymbol{a}^{k+1} = (\boldsymbol{G} + \Delta t \boldsymbol{L}_{\mathrm{Ex}})\boldsymbol{a}^k + \Delta t \boldsymbol{b}$$
$$a_n^0 = \langle u_h^0(\boldsymbol{\mu}^*), \varphi_n \rangle \quad n = 1, \dots, N. \tag{28}$$

Here, for the sake of readability, we used the abbreviations

$$\boldsymbol{a}^k := (a_1^k(\boldsymbol{\mu}), \dots, a_N^k(\boldsymbol{\mu}))^T, \tag{29}$$
$$\boldsymbol{G} := \boldsymbol{G}_{\Phi(\boldsymbol{\mu}^*)}, \tag{30}$$
$$\boldsymbol{L}_{\mathrm{Im}} := \boldsymbol{L}_{\mathrm{Im}}(t^{k+1}, \boldsymbol{\mu}), \tag{31}$$
$$\boldsymbol{L}_{\mathrm{Ex}} := \boldsymbol{L}_{\mathrm{Ex}}(t^k, \boldsymbol{\mu}), \tag{32}$$
$$\boldsymbol{b} := \boldsymbol{b}(t^k, \boldsymbol{\mu}^*). \tag{33}$$

We use the parameter separability of the operators and the right hand side to assemble $\boldsymbol{L}_{\mathrm{Im}}, \boldsymbol{L}_{\mathrm{Ex}}$ and $\boldsymbol{b}$ for all $k = 1, \dots, K$ in (28) rapidly by

$$\boldsymbol{L}_{\mathrm{Im}} = \sum_q^{Q_{\mathcal{L}_{\mathrm{Im}}}} \Theta_{\mathcal{L}_{\mathrm{Im}}}^q(t^k, \boldsymbol{\mu}^*) \left(C^{\mathcal{I}(\boldsymbol{\mu}^*)}\right)^T \boldsymbol{L}_{\mathrm{Im},\mathcal{I}(\boldsymbol{\mu}^*)}^q C^{\mathcal{I}(\boldsymbol{\mu}^*)}, \tag{34}$$

$$\boldsymbol{L}_{\mathrm{Ex}} = \sum_q^{Q_{\mathcal{L}_{\mathrm{Ex}}}} \Theta_{\mathcal{L}_{\mathrm{Ex}}}^q(t^k, \boldsymbol{\mu}^*) \left(C^{\mathcal{I}(\boldsymbol{\mu}^*)}\right)^T \boldsymbol{L}_{\mathrm{Ex},\mathcal{I}(\boldsymbol{\mu}^*)}^q C^{\mathcal{I}(\boldsymbol{\mu}^*)}, \tag{35}$$

$$\boldsymbol{b} = \sum_q^{Q_b} \Theta_b^q(t^k, \boldsymbol{\mu}^*) \boldsymbol{b}_{\mathcal{I}(\boldsymbol{\mu}^*)}^q C^{\mathcal{I}(\boldsymbol{\mu}^*)}, \tag{36}$$

where $(\boldsymbol{L}_{\mathcal{I}(\boldsymbol{\mu}^*)}^q), (\boldsymbol{L}_{\mathcal{I}(\boldsymbol{\mu}^*)}^q)$ and $\boldsymbol{b}_{\mathcal{I}(\boldsymbol{\mu}^*)}^q$ are the parts actually needed from the projected components calculated in (10)-(12). They are defined blockwise by

$$(\boldsymbol{L}_{\mathrm{Im},\mathcal{I}(\boldsymbol{\mu}^*)}^q)^{k,l} = \boldsymbol{L}_{\mathrm{Im},(\iota_k,\iota_l)}^q \tag{37}$$
$$(\boldsymbol{L}_{\mathrm{Ex},\mathcal{I}(\boldsymbol{\mu}^*)}^q)^{k,l} = \boldsymbol{L}_{\mathrm{Ex},(\iota_k,\iota_l)}^q \tag{38}$$
$$(\boldsymbol{b}_{\mathcal{I}(\boldsymbol{\mu}^*)}^q)^k = \boldsymbol{b}_{\iota_k}^q. \tag{39}$$

with $\iota_k \in \mathcal{I}(\boldsymbol{\mu}^*)$ and $\forall k, l = 1, \dots, |\mathcal{I}(\boldsymbol{\mu}^*)|$. Furthermore, $\mathcal{I}(\boldsymbol{\mu}^*) = \{\iota_1, \dots, \iota_{N_\mathcal{I}}\}$ is the index set of entities chosen for the online basis construction and $C^{\mathcal{I}(\boldsymbol{\mu}^*)}$ is the transition matrix defined in (16).

All the matrices and vectors in (28) are of dimension $\mathbb{R}^{N \times N}$ or $\mathbb{R}^N$. As $N \ll dim(X_h)$ is small, this evolution scheme can be solved rapidly.

4. EXPERIMENTS

4.1 Setting

In the numerical experiments we treat the parametrized advection-diffusion problem

$$\partial_t u(\cdot, t, \boldsymbol{\mu}) = \Delta(k u(\cdot, t, \boldsymbol{\mu})) - \nabla \cdot (\boldsymbol{v}(\boldsymbol{\mu}) u(\cdot, t, \boldsymbol{\mu}))) \tag{40}$$
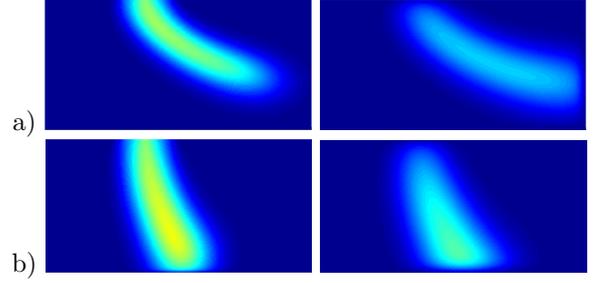$$u(\cdot, 0) = u_0 \tag{41}$$

Fig. 1. Solutions to the advection diffusion problem in a) with $\boldsymbol{\mu} = [0.75, 0.5]$ and in b) with $\boldsymbol{\mu} = [0.25, 0.5]$ at time instants $t = 0.5$ and $t = 1$.

on a two dimensional physical domain $\Omega = [0, 2] \times [0, 1]$. We assume Dirichlet boundary conditions $u(\boldsymbol{\mu}) = u_{dir}$ on $\Gamma \times [0, T]$. The velocity $\boldsymbol{v}$ is supposed to be a divergence free parameter- and time-dependent velocity field of the form

$$\boldsymbol{v}(\boldsymbol{x}, t; \boldsymbol{\mu}) = \left(\mu_1(1 - t) \cdot 5(1 - x_2^2), -\mu_2(1 - t)(4 - x_1^2)\right)^T$$

with $\boldsymbol{x} = (x_1, x_2)^T \in \Omega$. The parameters $\boldsymbol{\mu}$ stem from a two dimensional parameter domain $\boldsymbol{\mu} \in \mathcal{P} = [0, 1]^2$. This can be discretized with cell-wise constant functions and a Finite Volume scheme using an Engquist–Osher flux, which results in a corresponding discretization space X_h and discretized operators $\mathcal{L}_{\mathrm{Im},h}$ and $\mathcal{L}_{\mathrm{Ex},h}$ as well as in a discrete right hand side b_h for including the boundary conditions. We chose a space discretization into 128×64 intervals and a rectangular grid leading to 8192 degrees of freedom. In order to satisfy the CFL conditions we discretized in time into 1024 time steps. We chose an explicit/implicit discretization scheme with $\mathcal{L}_{\mathrm{Im},h}$ containing the discrete diffusion operator and $\mathcal{L}_{\mathrm{Ex},h}$ the discrete advection operator. Solutions for some chosen parameters and at different time instances are shown in Figure 1.

We constructed a reduced basis using the POD-Greedy algorithm (Haasdonk and Ohlberger (2008)) with a tolerance of $\varepsilon = 0.01$ and a dictionary following the new approach. For both methods the same training/dictionary set $\mathcal{M}_\mathcal{D}$ of 100 parameters, obtained from a 10×10 uniform mesh over the parameter domain, was used. The construction of the POD-Greedy basis took about 464 minutes while the construction of the dictionary took about 51 minutes. This difference in the offline time is due to the fact that the POD-Greedy performs a costly Greedy-search over the whole training set in every iteration during the extension algorithm. However, the dictionary is built up by just computing the solution for all $\boldsymbol{\mu}_e \in \mathcal{M}_\mathcal{D}$ and performing a POD for each of the solution trajectories.

4.2 Tests

The approaches were tested by conducting online simulations for 50 randomly chosen test parameters. In table 1 the different distance functions (Euclidean, Greedy-Extended, Greedy-Min, Greedy-Max) are compared to a standard reduced basis procedure. We set the online error tolerance for our approach to $\varepsilon_{tol} = 0.01$ and list the mean estimated error $\Delta_\varnothing$ over the test set of parameters, the mean effectivity of the error estimator $\eta_\varnothing$, the maximum value of the error estimator $\Delta_{\max}$, the mean online simulation time (including the online construction of the basis

Table 1. Mean error, estimator effectivity, maximum error, mean online simulation time and mean basis size for the different methods and a grid with 8192 elements.

Scheme	$\Delta_\varnothing$	$\eta_\varnothing$	$\Delta_{\max}$	$t_\varnothing$	$N_\varnothing$
Euclidean	$5.12e{-}3$	11.65	$9.69e{-}3$	3.24	58
Greedy-Extended	$5.36e{-}3$	11.51	$9.69e{-}3$	15.26	58
Greedy-Min	$5.68e{-}3$	11.52	$9.69e{-}3$	24.44	59
Greedy-Max	$6.05e{-}3$	12.16	$9.69e{-}3$	38.06	69
Standard RB	$8.06e{-}3$	20.12	$1.02e{-}2$	5.43	205

for our approach) $t_\varnothing$ and the mean basis size $N_\varnothing$. Here the effectivity η of an error estimator is defined as

$$\eta := \frac{\Delta_\Phi^K(\boldsymbol{\mu})}{\|u_h^K(\boldsymbol{\mu}) - u_N^K(\boldsymbol{\mu})\|}. \qquad (42)$$

We directly see the big advantages of our approach: Using the Euclidean distance function we gain a smaller mean error estimator at a smaller online simulation time and also a smaller maximum error estimator. Also, due to the more compact reduced basis, the effectivity of our error estimator is better.

While the results using the Euclidean distance function are pretty convincing, the different Greedy distance functions seem to give a rather poor performance for our approach. Regarding both approximation error and simulation time they are outperformed by the standard reduced basis approach and also by our approach using the Euclidean distance function. The poor approximation quality using the Greedy distance functions is due to the fact that we were not able to use the full dictionary while computing the minimum in (20). The repeated reduced simulations that take place in the Greedy distance functions just take too long for those approaches to be feasible with unstructured dictionaries larger than about 15 entities. Therefore, in those cases, we evaluated the distance function in only 15 entities, hence the poor performance regarding approximation quality. The high runtime of those algorithms is to be understood with the same explanation.

Another advantage of our approach is the guaranteed fulfillment of the given online error tolerance ε_{tol} if the dictionary is fine enough. The fulfillment of the tolerance cannot be guaranteed in standard reduced basis methods as is to be seen in table 1. Here, the maximum error using the standard RB approach is larger than the given tolerance of $\varepsilon_{\text{tol}} = 0.01$.

To underline the flexibility of the online basis construction we conducted another test. Here we did not prescribe an error tolerance for the approximation error but we set a desired online simulation time. Again we conducted online simulations for 50 randomly chosen parameters using the standard reduced model generated with a POD-Greedy algorithm (with a variable number of basis functions) and the online basis construction procedure using the euclidean distance measure. The results are shown in Figure 2.

Here we see another nice ability of our approach: Given a desired online runtime, we are able to compute a pretty compact, parameter adapted basis that outperforms the standard approach regarding approximation error by two orders of magnitude. The estimated error in case of the
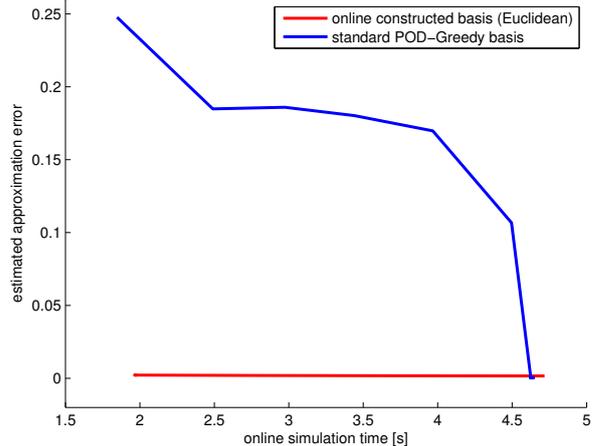

Fig. 2. The estimated approximation error plotted over the online simulation time for two reduced models generated with the standard POD-Greedy approach and with the online euclidean basis construction.

online constructed basis can be kept at an almost constant level of 0.0022 while the estimated error in the standard RB case rises to 0.247 for very short online simulation times.

5. DISCUSSION

We presented new online basis enrichment procedures for the reduced basis method in the case of evolution equations, where instead of using a reduced space suited for the whole parameter domain a local reduced space using precomputed basis entities is constructed in the online phase for a desired parameter by combining precomputed basis entities.

As the constructed reduced space is locally adapted, it can be of smaller dimension than the reduced space in standard RB-methods. Hence, the online complexity is lower and online simulations are faster. Of course, to obtain an overall better performance the online construction procedure has to be efficient and fast. Note that due to the smaller dimension of the adapted reduced space, the error estimators are much more effective.

Another advantage is the flexible control of the approximation error. In standard RB-methods the (POD-)Greedy basis generation assures that the approximation error tolerance is fulfilled on a set of training points stemming from the parameter domain. Yet, it is possible that this error tolerance is violated when performing simulations for parameters outside the training set. In our approach we adapt the reduced space so that the approximation error is lower than a given tolerance for the actually given parameter. Thereby we can guarantee the fulfillment of approximation error requirements for any parameter chosen from the parameter domain as long as the dictionary is sufficiently large. Furthermore the error tolerance for the online phase can be chosen almost freely without the need of reconducting a costly offline phase.

Although the offline time to generate the database of solutions is in general lower than the offline time to

generate a reduced basis using the POD-Greedy algorithm, the storage cost for the offline data is higher in our approaches. This is due to the fact, that we have to store a wide range of POD-compressed solution trajectories and projected operator components to allow all possible combinations of dictionary elements. A way to reduce the size of the storage requirements could be to restrain the number of possible combinations for each basis-entity and thereby omit entries in the matrices of projected operator components.

Another approach to construct online an even more compact reduced basis is the application of an "online POD-Greedy" algorithm where in the basis extension steps not the whole basis entity in a dictionary entry is added to the actual basis, but we orthogonalize the entity basis to the actual reduced basis and add the first mode of a POD over this orthogonalised trajectory as a new basis vector to our actual reduced basis. However, as even the simpler Greedy approaches in this work turned out to be too slow, one would first have to introduce a more sophisticated distance measure in order to get a competitive online POD-Greedy approach.

REFERENCES

Amsallem, D. and Farhat, C. (2011). An online method for interpolating linear parametric reduced-order models. *SIAM J. Sci Comp*, 33, No.5, 2169–2198.

Baur, U., Beattie, C., Benner, P., and Gucercin, S. (2011). Interpolatory projection methods for parameterized model reduction. *SIAM J. Sci Comp*, 33 No.5, 2489–2518.

Dihlmann, M., Drohmann, M., and Haasdonk, B. (2011). Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. In *Proc. of ADMOS 2011*.

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

Eftang, J.L., Patera, A.T., and Rønquist, E.M. (2010). An hp certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci Comp*, 32(6), 3170–3200.

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

Haasdonk, B., Dihlmann, M., and Ohlberger, M. (2011). A training set and multiple basis generation approach for parametrized model reduction based on adaptive grids in parameter space. *Mathematical and Computer Modelling of Dynamical Systems*.

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

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

Lohmann, B. and Eid, R. (2009). Efficient order reduction of parametric and nonlinear models by superposition of locally reduced models. In *Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2007 und 2008*, 27–36. Shaker Verlag, Aachen.

Rozza, G., Huynh, D., and Patera, A. (2008). 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.