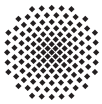

Universität Stuttgart

SimTech
Cluster of Excellence

M. Dihlmann[a] · **M. Drohmann**[b] · **B. Haasdonk**[a]

Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning

Stuttgart, May 2011

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

[b] Institute of Computational and Applied Mathematics, University of Münster,
Einsteinstr. 62, 48149 Münster, Germany
martin.drohmann@math.uni-muenster.de

Abstract odern 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 for high dimensional dynamical systems or partial differential equations. This demand for fast simulation models is even more critical for parametrized problems. There exist several snapshot-based methods for model order reduction of parametrized problems, e.g. proper orthogonal decomposition (POD) or reduced basis (RB) methods. An often faced problem is that the produced reduced models for a given accuracy tolerance are still of too high dimension. This is especially the case for evolution problems where the model shows high variability during time evolution. We will present an approach to gain control over the online complexity of a reduced model by an adaptive time domain partitioning. Thereby we can prescribe simultaneously a desired error tolerance and a limiting size of the dimension of the reduced model. This leads to *fast* and *accurate* reduced models. The method will be applied to an advection problem.

Keywords model order reduction; reduced basis method; evolution problem; parametrized partial differential equation; adaptive time partitioning

Preprint Series **Issue No. 2011-13**
Stuttgart Research Centre for Simulation Technology (SRC SimTech)
SimTech – Cluster of Excellence
Pfaffenwaldring 7a
70569 Stuttgart
publications@simtech.uni-stuttgart.de
www.simtech.uni-stuttgart.de

1 INTRODUCTION

Simulations of complex parametrized evolution problems 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 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 application and the development of model reduction techniques for parametrized models.

Projection based model reduction techniques are widely used, such as proper orthogonal decomposition [11], Krylov-subspace [1] or reduced basis methods [9]. In these methods the discrete operators are projected onto a reduced space so that the problem can be solved rapidly in this lower dimensional space.

However, if the problem depends on many parameters or if the solution shows a high variability with the parameters, a relatively high dimensional reduced space is needed in order to be able to represent all possible solution variations well, which leads to long online simulation times. This effect is even considerably increased when treating evolution problems with significant solution variations in time. These difficulties play a role particularly in case of real time applications, where full control over the online simulation time is required. Another aspect is the fact that projection based model reduction techniques generate *small but full matrices* while common discretization techniques (as FEM) lead to *large but sparse matrices*. It is even possible that calculating a solution with the reduced model is more time consuming than the simulation of the original model.

Consequently, the goal is to provide methods for generating reduced models being simultaneously accurate (concerning the approximation error) and performant (concerning the online simulation time) independent of the complexity in parameters and the complexity in the time evolution of the original problem. There exist approaches to control the online complexity of reduced models in parameter space in [3] and [5]. However, the same approximation space is used here over the whole time domain. We propose to generate a segmentation of the time interval into several smaller intervals and to construct a reduced approximation space on each of the time intervals. By an adaptive partitioning of the time domain we can even guarantee the accuracy of the reduced model with respect to a fixed error tolerance while limiting simultaneously the dimension of the approximation space per interval. Although the method can be applied to various projection based reduction techniques, we will put the focus here on the reduced basis (RB) method. An introduction to the RB method applied to time dependent problems can be found in [4],[9] and [6]. In literature we did not find similar approaches for a partitioning of the time domain in model reduction. Yet, in [2] an adaptive approach of generating collateral reduced bases on different time domains for the use in empirical interpolation of nonlinear operators was applied. An adaptive choice of the size of the reduced space at every time step during online simulation was realized in [7]. This approach optimizes the number of basis vectors used for the approximation of the solution but it does not give full control over the online complexity by strictly limiting the size of the reduced basis.

The current presentation is structured as follows. In Section 2 we introduce the general evolution equation and some notations. In Section 3 we give a brief introduction to the reduced basis method as model reduction technique of choice. In Section 4 the time domain partitioning approach is presented and a possible algorithm for adaptively partitioning the time domain is developed. The application of the method to an advection problem can be found in Section 5 followed by conclusions and an outlook in Section 6.

2 PROBLEM SETTING

We consider the general linear parameter dependent evolution equation

$$\partial_t u(\cdot, t; \boldsymbol{\mu}) = \mathcal{L}(t; \boldsymbol{\mu}) u(\cdot, t; \boldsymbol{\mu}) + b(\cdot, t; \boldsymbol{\mu}) \qquad \text{in } \Omega \tag{1}$$

$$u(\cdot, 0; \boldsymbol{\mu}) = u^0(\cdot; \boldsymbol{\mu}) \qquad \text{in } \Omega \tag{2}$$

with solutions $u(\cdot, t)$ from a Hilbert space $\mathcal{X}$ for all $t \in [\theta, T]$ and suitable boundary conditions. The parameter vector $\boldsymbol{\mu}$ stems from a possible set of parameters $\mathcal{P} \subseteq \mathbb{R}^p$. After discretization in space (by finite element or finite volume techniques, for example) and a discretization of the time interval $[\theta, T]$

by $K+1$ equidistant time instants $t^k := k\Delta t + \theta$ and a first order time integration we obtain the discrete evolution scheme

$$\left(Id - \Delta t \mathcal{L}_{h,\mathrm{Im}}(t^k;\boldsymbol{\mu})\right) u_h^{k+1}(\boldsymbol{\mu}) = \left(Id + \Delta t \mathcal{L}_{h,\mathrm{Ex}}(t^k;\boldsymbol{\mu})\right) u_h^k(\boldsymbol{\mu}) + \Delta t b_h(t^k;\boldsymbol{\mu}), \tag{3}$$

$$u_h^0(\boldsymbol{\mu}) = P\left(u^0(x;\boldsymbol{\mu})\right), \tag{4}$$

producing spatial solutions $u_h^k(\mu) = u_h(t^k;\mu)$ in a discrete function space $X_h \subset \mathcal{X}$ with $\dim(X_h) = H$ at time step $k = 0,\ldots,K$, where $P : \mathcal{X} \to X_h$ denotes the L^2 orthogonal projection operator. In order to obtain a very general formulation for the discrete evolution scheme, we included operator splitting of the operator $\mathcal{L}$ into an implicit part $\mathcal{L}_{h,\mathrm{Im}}$ and an explicit part $\mathcal{L}_{h,\mathrm{Ex}}$. For details we refer to [6]. For the separation of the procedure into a preparing offline phase and a rapid online simulation phase we need the operators $\mathcal{L}_{h,\mathrm{Im}}$ and $\mathcal{L}_{h,\mathrm{Ex}}$ as well as the right hand side b and the initial conditions to be parameter separable:

$$\mathcal{L}_{h,\mathrm{Im}}(t^k,\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_{\mathrm{Im}}}} \Theta^q_{\mathcal{L}_{\mathrm{Im}}}(t^k;\boldsymbol{\mu})\mathcal{L}^q_{h,\mathrm{Im}} \qquad b_h(t^k;\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta^q_b(t^k;\boldsymbol{\mu}) b^q \tag{5}$$

$$\mathcal{L}_{h,\mathrm{Ex}}(t^k,\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_{\mathrm{Ex}}}} \Theta^q_{\mathcal{L}_{\mathrm{Ex}}}(t^k;\boldsymbol{\mu})\mathcal{L}^q_{h,\mathrm{Ex}} \qquad u^0(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{u_0}} \Theta^q_{u_0}(\boldsymbol{\mu}) u_0^q. \tag{6}$$

The coefficients $\Theta^q_{[\cdot]}(t^k;\cdot) : \mathcal{P} \to \mathbb{R}$ can be evaluated rapidly in the online phase.

3 REDUCED BASIS METHOD

Although the technique of time domain partitioning in the generation of reduced parametrized models presented here can also be applied to other model reduction methods, we will focus here on the application of the reduced basis method to illustrate and explain the procedures.

3.1 REDUCED EVOLUTION SCHEME

In RB methods the reduced basis Φ_N consisting of basis vectors φ_n is constructed by solution snapshots corresponding to several parameters. The basis vectors $\varphi_n, n = 1,\ldots,N$ span the space $X_N = \mathrm{span}(\Phi_N) = \mathrm{span}\{\varphi_1,...,\varphi_N\} \subseteq X_h$ with the inner product inherited from $\mathcal{X}$. We assume that the basis vectors φ_n are orthonormal $\langle \varphi_n, \varphi_m \rangle = \delta_{nm}$ for $n,m = 1,...,N$. For the solution in the reduced space we start with the ansatz

$$u_N^k(\boldsymbol{\mu}) = \sum_{n=1}^{N} a_n^k(\boldsymbol{\mu})\varphi_n(x). \tag{7}$$

By a Galerkin projection of (3) onto X_N using (7) we obtain the reduced evolution scheme

$$\left(Id - \Delta t \boldsymbol{L}_{\mathrm{Im}}(t^{k+1};\boldsymbol{\mu})\right)\boldsymbol{a}^{k+1} = \left(Id + \Delta t \boldsymbol{L}_{\mathrm{Ex}}(t^k;\boldsymbol{\mu})\right)\boldsymbol{a}^k + \Delta t \boldsymbol{b}(t^k;\boldsymbol{\mu}) \tag{8}$$

$$a_n^0 = \left\langle u_h^0(\boldsymbol{\mu}), \varphi_n \right\rangle \qquad \forall n = 1,...,N \tag{9}$$

with $\boldsymbol{a}^k = (a_1^k,...,a_N^k)^T$. The reduced operators $\boldsymbol{L}_{\mathrm{Ex}}(t^k;\boldsymbol{\mu}), \boldsymbol{L}_{\mathrm{Im}}(t^k;\boldsymbol{\mu})$ in equation (8) are Gramian-like matrices with entries $\left(\boldsymbol{L}_{\mathrm{Ex}}\right)_{n,m}(t^k;\boldsymbol{\mu}) = \left\langle \mathcal{L}_{h,\mathrm{Ex}}(t^k;\boldsymbol{\mu})\varphi_m, \varphi_n \right\rangle$ and $\left(\boldsymbol{L}_{\mathrm{Im}}\right)_{n,m}(t^k;\boldsymbol{\mu}) = \left\langle \mathcal{L}_{h,\mathrm{Im}}(t^k;\boldsymbol{\mu})\varphi_m, \varphi_n \right\rangle$ respectively for $n,m = 1,...,N$. The projected right hand side vector $\boldsymbol{b}(t^k;\boldsymbol{\mu})$ has components $\left(\boldsymbol{b}\right)_m(t^k;\boldsymbol{\mu}) = \left\langle b(t^k;\boldsymbol{\mu}), \varphi_m \right\rangle$ for $m = 1,...,N$. All quantities and operators in the reduced evolution scheme (8) are of low dimension N and are independent of the original discrete space dimension H.

In order to circumvent conducting a Galerkin projection for every new parameter we use the property of parameter separability of the operators. Thereby, we can calculate in an offline phase the operator components projection

$$\left(\boldsymbol{L}^q_{\mathrm{Ex}}\right)_{n,m} = \left\langle \mathcal{L}^q_{h,\mathrm{Ex}}\varphi_m, \varphi_n \right\rangle \qquad \left(\boldsymbol{b}^q\right)_m = \left\langle b^q, \varphi_m \right\rangle \tag{10}$$

$$\left(\boldsymbol{L}^q_{\mathrm{Im}}\right)_{n,m} = \left\langle \mathcal{L}^q_{h,\mathrm{Im}}\varphi_m, \varphi_n \right\rangle \qquad \left(\boldsymbol{a}^{0,q}\right)_m = \left\langle u^q_{0,h}, \varphi_m \right\rangle \tag{11}$$

where $\boldsymbol{L}^q_{\mathrm{Ex}}, \boldsymbol{L}^q_{\mathrm{Im}} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{b}^q \in \mathbb{R}^N$. In order to assemble assemble reduced operators of a numerical scheme for an arbitrary parameter $\boldsymbol{\mu} \in \mathcal{P}$, we only need to compute linear combinations of these small components with coefficients $\Theta^q_{[\cdot]}$:

$$\boldsymbol{L}_{\mathrm{Ex}}(t^k, \boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_{\mathrm{Ex}}}} \Theta^q_{\mathcal{L}_{\mathrm{Ex}}}(t^k, \boldsymbol{\mu}) \boldsymbol{L}^q_{\mathrm{Ex}} \qquad \boldsymbol{b}(t^k, \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \Theta^q_b(t^k, \boldsymbol{\mu}) \boldsymbol{b}^q \tag{12}$$

$$\boldsymbol{L}_{\mathrm{Im}}(t^k, \boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_{\mathrm{Im}}}} \Theta^q_{\mathcal{L}_{\mathrm{Im}}}(t^k, \boldsymbol{\mu}) \boldsymbol{L}^q_{\mathrm{Im}} \qquad \boldsymbol{a}^0(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{u0}} \Theta^q_{u0}(\boldsymbol{\mu}) \boldsymbol{a}^{0,q} \tag{13}$$

3.2 A-POSTERIORI ERROR ESTIMATION

Reduced basis methods provide a-posteriori error estimators $\left\| u^k_h(\boldsymbol{\mu}) - u^k_N(\boldsymbol{\mu}) \right\| \leq \Delta^k(\boldsymbol{\mu})$ bounding the approximation error between the reduced solution and the high-dimensional discrete solution for all $k = 0, \ldots, K$. During the online simulation such an upper bound for the approximation error can rapidly be calculated [4, 6, 10].

3.3 REDUCED BASIS GENERATION BY POD-GREEDY ALGORITHM

In reduced basis methods a common approach to build up a reduced basis space is the use of the POD-Greedy algorithm in time dependent cases [3, 6, 8]. In every loop of the POD-Greedy algorithm, we search on a training set $\mathcal{M}_{\mathrm{train}}$ of parameters the one parameter for which the reduced solution produces the highest estimated error. Next, a high dimensional detailed solution is calculated for this parameter. A POD over the time sequence of projection errors is performed and the first mode (or another fixed number of k modes) is added as a new basis vector to the existing reduced basis. This procedure is repeated until the maximum error estimator falls beneath a given tolerance.

4 TIME DOMAIN PARTITIONING

The basic idea is to construct a segmentation of the time domain into several intervals τ_i and to create reduced bases for each of these time intervals. In analogy to the parameter domain partitioning [3, 5] these specialized reduced bases on the time intervals require less basis vectors to approximate the solutions with a given error tolerance. An adaptive partitioning of the time domain allows to fix the maximum number of basis vectors per time interval $N_{\max}$ while keeping the overall approximation error below the tolerance $\varepsilon_{\mathrm{tol}}$.

We assume that the whole time domain $[\theta, T]$ is subdivided into Υ time intervals $\tau_1, ..., \tau_\Upsilon$ with $\tau_1 := [\theta, t^{\kappa(1)}]$, $\tau_2 := [t^{\kappa(1)}, t^{\kappa(2)}]$, ... $\tau_\Upsilon := [t^{\kappa(\Upsilon-1)}, T]$ so that $[\theta, T] = \bigcup_{i=1}^{\Upsilon} \tau_i$. We define that $\kappa(i)$ is the index of the time step at the joint border between interval i and $i+1$ so that $\tau_i \cap \tau_{i+1} = t^{\kappa(i)}$. $t^{\kappa(0)}$ is defined to be θ.

For every time domain interval τ_i, $i = 1, ..., \Upsilon$ we assume to have a reduced basis $\Phi_i = \{\varphi_{i,1}, \ldots, \varphi_{i,N_i}\}$ of size N_i which spans the reduced solution space $X_{N_i} = \text{span}(\Phi_i)$ for this time interval. We approximate the solution in this time interval τ_i using the appropriate reduced basis Φ_i of the segment in the ansatz

$$u_{N_i}^k(\boldsymbol{\mu}) = \sum_{n=1}^{N_i} a_{n,i}^k(\boldsymbol{\mu})\varphi_{n,i} \in X_{N_i}. \tag{14}$$

In an offline phase the reduced bases for every time interval are generated by starting the POD-Greedy early stopping algorithm (see Algorithm 1) on every part of the time interval. As we want to generate a basis representing well the solution variability on their time interval, we only consider the error produced on the actual domain for the algorithm. The reduced operator components are calculated according to (11) for every interval. In the online simulation phase the reduced evolution scheme (8) is conducted on every time interval. In order to obtain the "initial coefficients" $\boldsymbol{a}_i^{\kappa(i-1)}$ at the first time step of a new time interval we perform an orthogonal projection of the solution at the last time step of the previous interval $u_{N_{i-1}}^{\kappa(i)}$ onto the reduced space X_{N_i} of the current interval:

$$\left\langle u_{N_{i-1}}^{\kappa(i)}(\boldsymbol{\mu}) - u_{N_i}^{\kappa(i)}(\boldsymbol{\mu}), \varphi_{m,i} \right\rangle = 0 \tag{15}$$

for all $m = 1, \ldots, N_i$. With the ansatz (14) in (15) and assuming orthonormal bases we obtain $\boldsymbol{a}_i^{\kappa(i)}(\boldsymbol{\mu}) = \boldsymbol{T}_{(i-1,i)}\boldsymbol{a}_{i-1}^{\kappa(i)}(\boldsymbol{\mu})$ with $(\boldsymbol{T}_{(i-1,i)})_{m,n} = \langle \varphi_{m,i}, \varphi_{n,i-1} \rangle$ for $n = 1, \ldots, N_{i-1}$ and $m = 1, \ldots, N_i$ and $\boldsymbol{a}_i^k = \left(a_{1,i}^k, \ldots, a_{N_i,i}^k\right)^T$. The projection error $\Delta p_{i-1,i}$ can be calculated rapidly online by

$$\Delta p_{i,i+1}(\boldsymbol{\mu}) = \left\| u_{N_i}^{\kappa(i)}(\boldsymbol{\mu}) - u_{N_{i+1}}^{\kappa(i)}(\boldsymbol{\mu}) \right\|_2 = \sqrt{\left(\boldsymbol{a}_i^{\kappa(i)}(\boldsymbol{\mu})\right)^T \boldsymbol{a}_i^{\kappa(i)}(\boldsymbol{\mu}) - \left(\boldsymbol{a}_{i+1}^{\kappa(i)}(\boldsymbol{\mu})\right)^T \boldsymbol{a}_{i+1}^{\kappa(i)}(\boldsymbol{\mu})}. \tag{16}$$

This can be used to derive a-posteriori error estimators for our enhanced scheme.

Proposition 1 *Let be $r_i^k(\boldsymbol{\mu}) = u_h^k(\boldsymbol{\mu}) - u_{N_i}^k(\boldsymbol{\mu})$ the approximation error in the time interval τ_i at time step k with $\kappa(i-1) < k \leq \kappa(i)$. If assuming that $||r_1^0(\boldsymbol{\mu})|| = 0$ and that the implicit operator $\mathcal{L}_{h,Im}$ is negative definite, then the error can be bounded by $||r_i^k(\boldsymbol{\mu})|| \leq \Delta^k(\boldsymbol{\mu})$ with*

$$\Delta^k(\boldsymbol{\mu}) = \sum_{j=1}^{k} C^{k-j}(||\text{Res}_i^j(\boldsymbol{\mu})|| + ||\text{Res}_{proj}^{(j-1)}(\boldsymbol{\mu})||). \tag{17}$$

$C > 0$ is a constant depending on the explicit operator $\mathcal{L}_{h,Ex}$. The residual is defined as

$$\begin{aligned} \text{Res}_i^{k+1}(\boldsymbol{\mu}) = &\left(Id - \Delta t\mathcal{L}_{h,Im}(t^{k+1};\boldsymbol{\mu})\right) u_{N_i}^{k+1}(\boldsymbol{\mu}) \\ &- \left(Id + \Delta t\mathcal{L}_{h,Ex}(t^k;\boldsymbol{\mu})\right) u_{N_i}^k(\boldsymbol{\mu}) - \Delta t b_h^k(\boldsymbol{\mu}) \end{aligned} \tag{18}$$

and i is chosen appropriately to the according time step $\kappa(i-1) < k \leq \kappa(i)$. The projection residual Res_{proj}^j is defined as $\text{Res}_{proj}^k(\boldsymbol{\mu}) = \sum_{i=1}^{\Upsilon-1} \delta_{\kappa(i)k}\Delta p_{i,i+1}(\boldsymbol{\mu})$ where $\delta_{\kappa(i)k}$ is supposed to be the Kronecker delta.

Proof In general we can estimate the norm $||r_i^k(\boldsymbol{\mu})||$ of the approximation error by putting the definition of the approximation error $r^k(\boldsymbol{\mu})_i = u_h^k(\boldsymbol{\mu}) - u_{N_i}^k(\boldsymbol{\mu})$ in (3), rearranging the terms and assuming $||Id - \Delta t\mathcal{L}_{h,\text{Im}}(t^{k+1};\boldsymbol{\mu})||^{-1} \leq 1$ due to the negative definiteness and $0 < ||Id + \Delta t\mathcal{L}_{h,\text{Ex}}(t^k;\boldsymbol{\mu})|| \leq C$ to obtain

$$||r_i^{k+1}(\boldsymbol{\mu})|| \leq C||r_i^k|| + ||Res_i^{k+1}(\boldsymbol{\mu})||. \tag{19}$$

For details of this deduction we refer to [6]. If k is the first time step of an interval ($k = \kappa(i)$ for any $i = 1, ..., \Upsilon - 1$) we do not know the value for the "initial error" $||r_i^k||$, but we can estimate its value by

$$\begin{aligned} ||r_i^k(\boldsymbol{\mu})|| &= ||u_h^k(\boldsymbol{\mu}) - u_{N_i}^k(\boldsymbol{\mu})|| = ||u_h^k(\boldsymbol{\mu}) - u_{N_{i-1}}^k(\boldsymbol{\mu}) + u_{N_{i-1}}^k(\boldsymbol{\mu}) - u_{N_i}^k(\boldsymbol{\mu})|| \\ &\leq ||u_h^k(\boldsymbol{\mu}) - u_{N_{i-1}}^k(\boldsymbol{\mu})|| + ||u_{N_{i-1}}^k(\boldsymbol{\mu}) - u_{N_i}^k(\boldsymbol{\mu})|| \leq ||r_{i-1}^k(\boldsymbol{\mu})|| + \Delta p_{i,i+1}(\boldsymbol{\mu}). \end{aligned} \tag{20}$$

Calculating $||r_i^k(\boldsymbol{\mu})||$ recursively with (19) and (20) leads to (17).

4.1 ADAPTIVE TIME DOMAIN PARTITIONING

When using a fixed partitioning of the time domain we obtain a more accurate and faster model. We will now present an adaptive way for a partitioning of the time domain guaranteeing an overall approximation error lower than ε_{tol} while limiting simultaneously the basis size to $N_{\max}$. The algorithm to this adaptive approach is described in Algorithm 2. The overall goal of this algorithm is to generate a reduced model with the following properties:

- It produces a uniform error growth over the whole time during online simulations.
- It has limited online complexity. (The basis size on each interval is limited a priori.)
- The maximum approximation error stays below a given error tolerance.

To estimate the approximation error we use the error estimator for general evolution equations from Proposition 1. As it grows monotonically, the maximum error estimator value is found at the last time step and this value $\Delta^K(\boldsymbol{\mu})$ should be kept below a given global error tolerance $\varepsilon_{\text{tol,global}}$. In order to have an approximately uniform growth of the error on the whole time domain, we fix the error tolerance for an interval τ_i to $\varepsilon_{\text{tol},i} = \varepsilon_{\text{tol,global}}(t^{\kappa(i)} - t^{\kappa(i-1)})/T$. We start the basis generation using the POD-Greedy algorithm on an interval. As soon as the maximum size $N_{\max}$ of the reduced basis is reached, the POD-Greedy algorithm is stopped and a refinement of the time domain is triggered. In the present work, each interval marked for refinement is divided into two intervals of equal size. After a segmentation of the time interval we restart the POD-Greedy algorithm on every interval while fixing the error tolerance on the new time intervals to $\varepsilon_{\text{tol},i}$, fixing the maximum basis size to $N_{\max}$ and adapting the indices of the bases and intervals. This procedure is conducted until a segmentation of the time domain is obtained such that on every interval the reduced basis has less then $N_{\max}$ basis vectors and a training error lower than $\varepsilon_{\text{tol},i}$.

EarlyStoppingGreedy($\Phi_0, M_{\text{train}}, \varepsilon_{\text{tol}}, M_{\text{val}}, \rho_{\text{tol}}, N_{\max}$)

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

Algorithm 1: The early-stopping (POD-)greedy search algorithm, for $\rho_{\text{tol}} = \infty$, $N_{\max} = \infty$ recovering the standard (POD-)greedy procedure.

AdaptiveTimePartition($\mathcal{T}_0, \varepsilon_{\text{tol,global}}, N_{\max}$)
1 $\mathcal{T} := \mathcal{T}_0, \Phi_i := \emptyset$ for $\tau_i \in \mathcal{T}$
2 **repeat**
3 $\Upsilon = \text{card}(\mathcal{T})$
4 **for** $i = 1, ...\Upsilon$ with $\Phi_i = \emptyset$
5 **do** $\Phi_i :=$ InitBasis(i)
6 $M_{\text{train},i} :=$ Mtrain(i)
7 $\eta_i := 0$
8 $\varepsilon_{\text{tol},i} := \varepsilon_{\text{tol,global}} \cdot (t^{\kappa(i+1)} - t^{\kappa(i)})/(T - \theta)$
9 $[\Phi_i, \varepsilon_i] :=$ EarlyStoppingGreedy($\Phi_i, M_{\text{train},i}, \varepsilon_{\text{tol},i}, \emptyset, \infty, N_{\max}$)
10 **if** $\varepsilon_i > \varepsilon_{\text{tol},i}$
11 **then** $\eta_i := 1$
12 $\eta_{\max} := \max_{i=1,...,\Upsilon} \eta_i$
13 **if** $\eta_{\max} > 0$
14 **then** $[\mathcal{T}, \boldsymbol{\Phi}] :=$ RefineTPart($\mathcal{T}, \boldsymbol{\Phi}, \boldsymbol{\eta}, \mathcal{M}_{\text{val}}$)
15 **until** $\eta_{\max} = 0$
16 **return** $\mathcal{T}, \{\Phi_i, \varepsilon_i\}_{i=1}^{\Upsilon}$

Algorithm 2: The adaptive time partition algorithm generates automatically a partitioning of the time domain and generates a reduced basis on each domain having less than $N_{\max}$ basis vectors and an approximation error on the training set $\mathcal{M}_{\text{train},i}$ lower than $\varepsilon_{\text{tol},i}$. $\mathcal{T} = \{\tau_i\}_{i=1}^{\Upsilon}$ is the set of all time intervals with the initial set $\mathcal{T}_0$ and η_i marks the intervals which have to be refined by a refinement algorithm.

5 EXPERIMENTS

5.1 THE ADVECTION MODEL

In the experiments we consider the advection problem

$$\partial_t u(\mu) = -\nabla \cdot (\boldsymbol{v}(\mu) u(\mu)) \text{ in } \Omega \times [0, T] \tag{21}$$

with $\Omega := [0, 2] \times [0, 1]$, $\theta = 0$ and $T = 1$. We assume suitable initial conditions $u(\mu) = u^0(\mu)$ for $t = 0$. Furthermore, Dirichlet boundary conditions $u(\mu) = u_{\text{dir}}$ on $\Gamma_{\text{dir}} \times [0, T]$ and Neumann boundary conditions $\nabla u(\mu) \cdot \boldsymbol{n} = u_{\text{neu}}$ on $\Gamma_{\text{neu}} \times [0, T]$ are prescribed. The velocity $\boldsymbol{v}$ is supposed to be a divergence free parameter and time dependent velocity field of the form

$$\boldsymbol{v}(\boldsymbol{x}, t; \mu) = \begin{pmatrix} \mu(1 - t) \cdot 5(1 - x_2^2) \\ -0.5(1 - t)(4 - x_1^2) \end{pmatrix}$$

with $\boldsymbol{x} = (x_1, x_2)^T \in \Omega$. 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 discretization operators $\mathcal{L}_{h,\text{Im}}$ and $\mathcal{L}_{h,\text{Ex}}$ as well as in a discrete right hand side b_h for including the boundary conditions. We chose a space discretization into 64×32 intervals and a triangular grid leading to 4096 degrees of freedom. For satisfying the CFL conditions we discretized time into 512 time steps. Here, we chose a pure explicit discretization scheme with $\mathcal{L}_{\text{Im}} = 0$. Solutions are illustrated in Figure 1. As the control of the parameter complexity is not the issue here we restrained our model to be dependent of only one parameter. (This parameter controls the strength of the velocity field in x-direction.) In case of models depending on many parameters and in case of high solution variability with the parameter changes, the adaptive methods from [5] can be applied.

5.2 RB MODEL REDUCTION WITH TIME DOMAIN PARTITIONING

We generated reduced basis spaces using a POD-Greedy algorithm in three different ways: without T-partitioning, on predefined equally sized subdivisions into 7, 64 and 128 intervals of the time domain and with the adaptive approach from Section 4.1 limiting the maximum number of reduced basis functions by $N_{\max} = 45$. The desired error tolerance was set to $\varepsilon_{\text{tol,global}} = 10^{-2}$. Online simulations

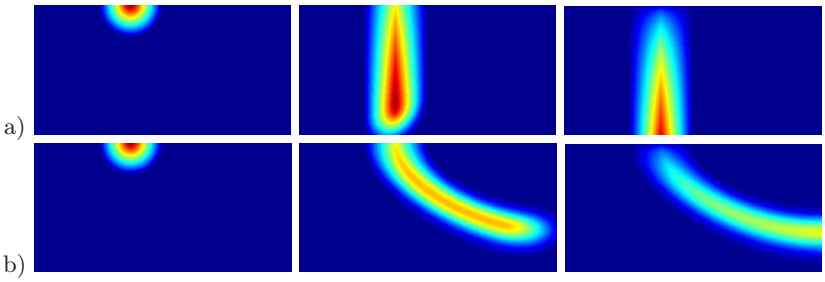

Fig. 1 Solutions to the advection problems with a) $\mu = 0$ and b) $\mu = 1$ at time instants $t = 0$, $t = 0.3$ and $t = 1$. Obviously, the solutions varies considerably with time.

adaptation	Υ	ø-dim(RB)	ø-online time[s]	max. error	offline time[h]
-	1	84.00	0.7	$9.87 \cdot 10^{-3}$	0.84
yes	7	33.63	0.61	$7.85 \cdot 10^{-3}$	2.10
no	7	34.31	0.61	$9.32 \cdot 10^{-3}$	0.70
no	64	24.61	0.61	$6.28 \cdot 10^{-3}$	5.08
no	128	23.43	0.64	$7.36 \cdot 10^{-3}$	12.13

Table 1 Comparison of average reduced basis sizes, offline time, average run-times and maximum error estimates for non-adaptive and adaptive runs with different fineness of the time interval partition. The average online run-times and maximum errors are obtained from 20 simulations with randomly selected parameters $\boldsymbol{\mu}$.

were performed for a set of 20 randomly chosen parameters using all previously generated models. Table 1 compares the reduced basis sizes averaged over the sub-intervals, the average online simulation time, the maximum estimated error during online simulations and the offline time consumed for the basis generation. We observe that a predefined subdivision into seven sub-intervals already leads to a significant reduction of the reduced basis sizes by a factor of 2.5. It is noteworthy, that even the offline time is slightly reduced in this case due to the polynomial complexity of the basis generation w.r.t. the number of basis functions.

The bases on the very fine divisions into 64 and 128 intervals (meaning respectively 8 and 4 time-steps per interval) are practically of the same average size. Consequently, these can be considered as bases of minimal possible basis size per interval representing the limit of what we are able to reach when using the T-partition approach alone. We need this minimal basis size per interval to cover the parameter variability of the solution. Table 1 also shows, that the adaptive basis generation approach produces only slightly larger reduced bases ($\text{ø}\dim(RB) = 33.63$) than the "minimal possible bases" ($\text{ø}\dim(RB) \sim 25$). The same fact is also illustrated in Figure 2b where the reduced bases dimensions on the sub-intervals are shown. The adaptively generated basis envelops closely the minimal possible basis sizes. Both models produce the largest reduced spaces near the point of highest solution variation around $t = 0.5$. The fact that the adaptively generated subdivision of the time domain is very close to the optimum is also confirmed by Figure 2a, which shows the estimated error evolution over time for different reduced models. First, we see that the error grows almost uniformly over the whole time domain as desired. The error evolution of the adaptively generated T-partition model is close to the maximal feasible error evolution by a very small refinement into 128 partitions. Furthermore, we observe that the projection error between the intervals is non-negligible but diminishes for smaller intervals.

We stated that it was possible to generate *fast* and *accurate* reduced models using the T-partition approach, meaning that the approximation error as well as the reduced basis dimension are simultaneously controllable. In order to show this we generated several reduced models with different demands on the error tolerance $\varepsilon_{\text{tol,global}}$ at a predefined basis size constraint $N_{\max} = 40$ using the adaptive T-partition approach. For comparison we created reduced models without T-partitioning. We calculated for a validation set of 25 randomly chosen parameters the average simulation time as well as the maximum estimated approximation error. The results are illustrated in Figure 3. We observe the average simulation time rises for the model without adaptive T-partitioning. This is due to the fact that we need higher dimensional reduced models to meet the accuracy requirements. Yet, when using the adaptive T-partition approach we can limit the maximum number of basis vectors per interval

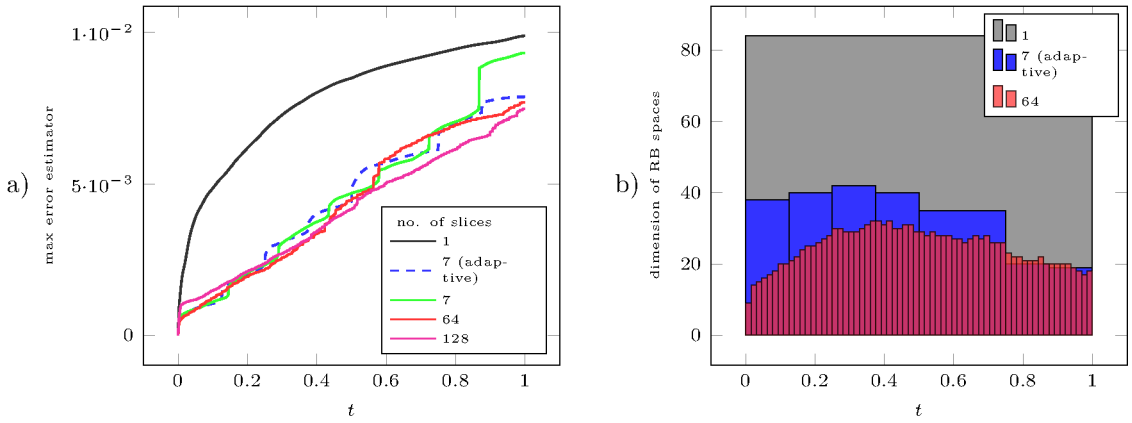

Fig. 2 Comparison of models with non-adaptively and adaptively generated T-partition bases and different fineness of the partitions: a) Illustration of the time-evolution of the maximum error estimator over a set of 20 randomly chosen parameters. b) Illustration of reduced basis sizes on time intervals.

(almost) independently of the error tolerance. Consequently, when the demand to the error tolerance is augmented the average simulation time can be kept almost constant and we obtain fast and accurate models.

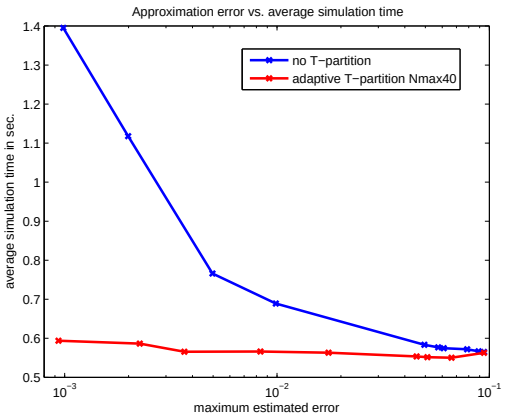

Fig. 3 The average simulation time plotted over the maximal error estimator over a randomly chosen test set of 25 parameters using reduced models with and without T-partitioning.

6 CONCLUSION AND OUTLOOK

With the time domain partitioning approach we presented a generic method for treating model reduction of evolution problems and guaranteeing simultaneously online time efficiency and accuracy. This is realized by an adaptive partitioning of the time domain into several intervals and creating specialised reduced bases with limited size on each of the intervals. We showed in experiments with an advection problem dependent on one parameter, that applying the method leads to a considerable improvement of the approximation error while the online simulation time is kept on a low level.

As this method produces a non-negligible projection error between the intervals we see room for improvement. In case of problems with complex parameter dependency, it is probable that the T-partition approach does not have enough effect for a considerable improvement of the approximation error. However, this problem should be solved by combining the T-partition approach with the P-partition approach from [5].

References

1. A.C. Antoulas. An overview of approximation methods for large-scale dynamical systems. *Annual Reviews in Control*, 29:p.181–190, 2005.
2. M. Drohmann, B. Haasdonk, and M. Ohlberger. Adaptive reduced basis methods for nonlinear convection-diffusion equations. In *Proceedings of FVCA6*, 2011.
3. J. L. Eftang, A. T. Patera, and E. M. Ronquist. An hp certified reduced basis method for parametrized parabolic partial differential equations. In *In Proceedings of ICOSAHOM 2009*, 2010.
4. 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.
5. B. Haasdonk, M. Dihlmann, and M. Ohlberger. 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*, 2011.
6. 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.
7. B. Haasdonk and M. Ohlberger. Space-adaptive reduced basis simulation for time-dependent problems. In *Proc. MATHMOD 2009, 6th Vienna International Conference on Mathematical Modelling*, 2009.
8. 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.
9. N.C. Nguyen, G. Rozza, D.B.P. Huynh, and A. Patera. Reduced basis approximation and a posteriori error estimation for parametrized parabolic pdes; application to real-time bayesian parameter estimation. *John Wiley & Sons*, 2010.
10. D.V. Rovas, L. Machiels, and Y. Maday. Reduced basis output bound methods for parabolic problems. *IMA J. Numer. Anal.*, 26(3):423–445, 2006.
11. S. Volkwein. Model reduction using proper orthogonal decomposition, 2008. Lecture notes, University of Graz.