# Adaptive Reduced Basis Methods for Nonlinear Convection–Diffusion Equations

Martin Drohmann, Bernard Haasdonk, and Mario Ohlberger

**Abstract** Many applications from science and engineering are based on parametrized evolution equations and depend on time-consuming parameter studies or need to ensure critical constraints on the simulation time. For both settings, model order reduction by the reduced basis methods is a suitable means to reduce computational time. In this proceedings, we show the applicability of the reduced basis framework to a finite volume scheme of a parametrized and highly nonlinear convection-diffusion problem with discontinuous solutions. The complexity of the problem setting requires the use of several new techniques like parametrized empirical operator interpolation, efficient a posteriori error estimation and adaptive generation of reduced data. The latter is usually realized by an adaptive search for base functions in the parameter space. Common methods and effects are shortly revised in this presentation and supplemented by the analysis of a new strategy to adaptively search in the time domain for empirical interpolation data.

**Key words:** finite volume methods, model reduction, reduced basis methods, empirical interpolation
**MSC2010:** 65M08, 65J15, 65Y20

## 1 Introduction

Reduced basis (RB) methods are popular methods for model order reduction of problems with parametrized partial differential equations that need to be solved for

M. Drohmann and M. Ohlberger
Institute of Computational and Applied Mathematics, University of Muenster, Einsteinstr. 62, 48149 Muenster, e-mail: mdrohmann, ohlberger@uni-muenster.de

B. Haasdonk
Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, 70569 Stuttgart e-mail: haasdonk@mathematik.uni-stuttgart.de

many parameters. Such scenarios might occur in parameter studies, optimization, control, inverse problems or statistical analysis for a given parametrized problem. Such problems deal with different solutions $u_h(\mu) \in \mathscr{W}_h$ from a high dimensional discrete function space $\mathscr{W}_h \subset L^2(\Omega)$ which are characterized by a parameter $\mu \in \mathscr{P} \subset \mathbb{R}^p$. For evolution problems, a discrete solution forms a series of what we call "snapshot solutions" $u_h^k(\mu)$ indexed by a time-step number $k = 0, \ldots, K$.

By applying the reduced basis method, these solution trajectories need to be computed for a few parameters only and can then be used to span a problem-specific subspace $\mathscr{W}_{\text{red}} \subset \mathscr{W}_h$. If this subspace captures a broad solution variety, a numerical scheme based on this reduced basis space $\mathscr{W}_{\text{red}}$ can produce reduced solutions $u_{\text{red}}(\mu) \in \mathscr{W}_{\text{red}}$ very inexpensively for every parameter $\mu \in \mathscr{P}$. In case of nonlinear discretizations or complex dependencies of the equations on the parameter, the reduced scheme requires an empirical interpolation method [1] to efficiently interpolate operator evaluations in a low-dimensional discrete function space.

The applicability of the reduced scheme has been successfully demonstrated for stationary, instationary, linear and nonlinear problems mainly based on finite element schemes (cf. [7] and the references therein). In this presentation, we focus on a scalar, but highly nonlinear convection–diffusion problem: For a given parameter $\mu \in \mathscr{P}$ determine solutions $u = u(x,t;\mu)$ fulfilling

$$\partial_t u + \nabla \cdot (\mathbf{v}(u;\mu)u) - \nabla \cdot (d(u;\mu)\nabla u) = 0 \qquad \text{in } \Omega \times [0, T_{\max}] \tag{1}$$

$$u(0;\mu) = u_0(\mu) \qquad \text{in } \Omega \times \{0\} \tag{2}$$

plus Dirichlet boundary conditions $u(\mu) = u_{\text{dir}}(\mu)$ on $\Gamma_{\text{dir}} \times [0, T_{\max}]$, Neumann boundary conditions $(\mathbf{v}(u;\mu)u - d(u;\mu)\nabla u) \cdot \mathbf{n} = u_{\text{neu}}(\mu)$ on $\Gamma_{\text{neu}} \times [0, T_{\max}]$ with suitable parametrized functions $\mathbf{v}(\cdot;\mu) \in C(\mathbb{R}, \mathbb{R}^d)$ and $d(\cdot;\mu) \in C(\mathbb{R}, \mathbb{R}^+)$.

For complex data functions, solutions of this problem can depend on the parameter in a highly nonlinear way, and the convection term can lead to a variety of solution snapshots which is difficult to capture by a linear subspace $\mathscr{W}_{\text{red}}$. This makes the construction of the reduced basis space $\mathscr{W}_{\text{red}}$ difficult and therefore requires sophisticated construction algorithms for the reduced data. After elaborating on the empirical operator interpolation and the reduced basis scheme for problem (1)-(2) in Section 2, we provide an overview of such algorithms in Section 3 with a focus on the time-adaptive construction of interpolation for the empirical interpolation. In Section 4, we numerically discuss the effects and costs of the introduced algorithms based on a finite volume discretization of a Buckley–Leverett type problem.

## 2 Reduced basis method

In this section, we present a reduced basis method for general operator based discretizations of equations (1), (2). We show that the reduced scheme depends both in memory and computational complexity on the low dimensions of suitable reduced spaces only and can therefore be efficiently evaluated. We first introduce the ba-

sic approach, and discuss the main ingredients to efficiently compute the reduced solutions at the end of this section. For a more detailed presentation, we refer to [2].

As a starting point for the reduced basis scheme, we assume a high dimensional discretization scheme producing for each parameter $\mu \in \mathscr{P}$ a sequence of solution snapshots $u_h^k(\mu)$ stemming from an $H$-dimensional discrete function space $\mathscr{W}_h$. The sequence indices $k = 0, \dots, K$ correspond to strictly increasing time steps $t^k := k\Delta t$ from the interval $[0, T_{\max}]$, where $\Delta t > 0$ is a global time step size. For the high-dimensional scheme, first, the initial data is projected on the discrete function space yielding a discrete solution $u_h^0(\mu) = \mathscr{P}_h[u_0(\mu)]$, where $\mathscr{P}_h : L^2(\Omega) \to \mathscr{W}_h$ is a projection operator. Subsequently, equations of the form

$$(\mathrm{Id} + \Delta t \mathscr{L}_I(\mu)) \left[u_h^{k+1}(\mu)\right] - (\mathrm{Id} + \Delta t \mathscr{L}_E(\mu)) \left[u_h^k(\mu)\right] = 0 \tag{3}$$

are solved with the Newton–Raphson method. The operators $\mathscr{L}_I(\mu), \mathscr{L}_E(\mu) : \mathscr{W}_h \to \mathscr{W}_h$ describe the explicit and implicit discretization terms of a first order Runge–Kutta scheme. For our numerical experiments presented in Section 4, the operators implement finite volume fluxes for the diffusive respectively convective terms.

For the reduced basis scheme, we first assume a given reduced basis space $\mathscr{W}_{\mathrm{red}} \subset \mathscr{W}_h$ of dimension $N \ll H$. This space is spanned by selected solution snapshots and its construction implies a computationally expensive preprocessing step. This allows to solve for reduced solutions $u_{\mathrm{red}}^k(\mu) \in \mathscr{W}_{\mathrm{red}}$. These are computed by projection of the initial data on the reduced basis space and with the same evolution scheme as in (3), but with the operators $\mathscr{L}_I(\mu), \mathscr{L}_E(\mu)$ substituted by reduced counterparts

$$\mathscr{L}_{\mathrm{red},I}^{k+1}(\mu) := \mathscr{P}_{\mathrm{red}} \circ \mathscr{I}_{M^{k+1}}^{k+1} \circ \mathscr{L}_I(\mu) \quad \text{and} \quad \mathscr{L}_{\mathrm{red},E}^{k}(\mu) := \mathscr{P}_{\mathrm{red}} \circ \mathscr{I}_{M^{k}}^{k} \circ \mathscr{L}_E(\mu) \tag{4}$$

at each time instance $k = 0, \dots, K-1$. Here, $\mathscr{P}_{\mathrm{red}} : \mathscr{W}_h \to \mathscr{W}_{\mathrm{red}}$ is a further projection operator and the actual operator evaluations are substituted by approximations in a further low dimensional function space $\mathscr{W}_M \subset \mathscr{W}_h$. This approximation, the so-called *empirical operator interpolation*, is denoted by $\mathscr{I}_M \circ \mathscr{L}$ and shortly summarized in the next subsection. Note that in this scheme the empirical operator interpolation and therefore also the reduced function spaces can vary over time.

**Empirical operator interpolation and offline/online splitting** The idea of empirical interpolation was first introduced in [1]. The empirical operator interpolation presented here is extracted from [2].

The principal idea is to interpolate functions $v_h \in \mathscr{W}_h$ in a *collateral reduced basis space* $\mathscr{W}_M$ spanned by basis functions $q_m, m = 1, \dots, M$ with exact evaluations at interpolation points $x_m \in T_M$, i.e.

$$\mathscr{I}_M[v_h](x_m) = \sum_{m=1}^{M} \sigma_m q_m(x_m) = v_h(x_m), \tag{5}$$

where the coefficients can be determined easily because the construction process for the basis functions ensures for each $m = 0, \dots, M$ that the condition $q_m(x_{m'}) = 0$ is fulfilled for all $m' < m$. By optimizing the collateral reduced basis space such

that it well approximates operator evaluations $\mathscr{L}_h(\mu)[v_h] \in \mathscr{W}_h$ of a parameterized discrete operator $\mathscr{L}_h(\mu)$ on solution snapshots $v_h$, we obtain an approximation $\mathscr{I}_M[\mathscr{L}_h(\mu)[v_h]] \approx \mathscr{L}_h(\mu)[v_h]$ which can be computed by evaluating the operator locally at $M$ given interpolation points. If such an evaluation depends only on a few degrees of freedom of the argument function ($H$ independent Dof-dependence) and $M \ll H$, the interpolation can be computed very efficiently. The interpolant is therefore suitable for the reduced basis method. Furthermore, it can be verified that the same argumentation also applies to Fréchet derivatives of discrete operators fulfilling the $H$ independent Dof-dependence. This result is needed for the efficient implementation of the Newton–Raphson method. In Section 3.2, we summarize how the discrete function space $\mathscr{W}_M$ can be constructed by a greedy search algorithms in a finite set of operator evaluations.

In order to evaluate the reduced numerical scheme efficiently, the high dimensional data needs to be precomputed in an expensive offline phase and to be reduced to low-dimensional matrices and vectors. Afterwards, every Newton step of a reduced simulation can be computed with complexity $\mathscr{O}(NM^2 + N^3)$ including the costs of the linear equation solver. In [2], the computations leading to these results are presented in detail. The same article also introduces an efficiently computable a posteriori error estimator $\eta(\mu)$ estimating the error

$$\max_{k=0,\dots,K} \left\| u_h^k(\mu) - u_{\text{red}}^k(\mu) \right\| \leq \eta(\mu) \tag{6}$$

for a suitable problem-specific norm $\|\cdot\|$.

## 3 Adaptive basis generation strategies

In this section, we give an introduction on how reduced basis functions and empirical interpolation data are constructed by algorithms that greedily search in a finite subset of the parameter space for new basis functions. For complex parameter sets or complex dependencies of the solution on the parameter, these algorithms, however, can result in very large reduced basis spaces and therefore make the speed advantages of the reduced simulations obsolete. For this reason, we also discuss variations of the algorithms adapting the parameter search set during the basis construction which lead to smaller and better basis spaces.

### *3.1 POD-greedy algorithm*

The "POD-greedy" algorithm introduced in [4] is used to generate the reduced basis space $\mathscr{W}_{\text{red}}$. Its purpose is to minimize the error $\|u_h(\mu) - u_{\text{red}}(\mu)\|$ for all $\mu \in \mathscr{P}$ in a suitable problem-specific norm. We assume the existence of an estimator $\eta(\mu)$ as introduced in (6), a finite training set $M_{\text{train}} \subset \mathscr{P}$ and an initial choice for the

reduced basis $\Phi_{N_0} := \{\varphi_n\}_{n=1}^{N_0}$. For evolution problems, the span of this initial reduced basis usually comprises all initial data functions. Then, the reduced basis can be iteratively extended by searching for the parameter $\mu_{\max} := \arg\max_{\mu \in M_{\text{train}}} \eta(\mu)$ of the worst approximated trajectory, and adding the first and most significant mode gained from a proper orthogonal decomposition of this trajectory's projection errors $\left\{u_h^k(\mu_{\max}) - \mathscr{P}_{\text{red}}\left[u_h^k(\mu_{\max})\right]\right\}_{k=0}^{K}$ as a new basis function. This algorithm is repeated, until $\eta(\mu_{\max})$ falls beneath a given tolerance.

**Adaptation techniques:** The basic algorithm described above depends on a fixed initial choice for the training subset $M_{\text{train}}$. In case of complex dependencies of the solution trajectories on the parameter, the reduced basis approximation can therefore turn out to be very bad for parameters not in the training set. In [6] this problem is addressed by adaptively refining the parameter space if indicated by bad approximations from a further validation training set.

Other variations of the POD-Greedy algorithm adaptively partition the parameter space and construct different reduced bases for each of these partitions [3, 5] leading to faster reduced simulations at the cost of a more expensive offline phase.

### *3.2 Time-adaptive empirical operator interpolation*

The construction of the collateral reduced basis space and corresponding interpolation points follows a similar idea like the "POD-greedy" algorithms. For the empirical interpolation of an operator $\mathscr{I}_M \circ \mathscr{L}_h$, the interpolation error $\left\| v_h - \sum_{j=1}^{M} \mathscr{I}_M[v_h] \right\|$ is minimized over all $v_h \in \mathbf{L} := \left\{ \mathscr{L}_h(\mu)\left[u_h^k(\mu)\right] \mid \mu \in \mathscr{P}, k = 0, \cdots, K-1 \right\}$. Analogously to the reduced basis generation, we define a finite subset $L_{\text{train}} \subset \mathbf{L}$ and pick one of this set's snapshots as an initial collateral reduced basis function. The extension step for the empirical interpolation then looks as follows:

1. Find the approximation with the worst error $v_M \leftarrow \arg\sup_{v_h \in L_{\text{train}}} \|u_h - \mathscr{I}_M[v_h]\|$.
2. Compute the residual between $v_M$ and its interpolant $r_M \leftarrow v_M - \mathscr{I}_M[v_h]$.
3. Find the interpolation point maximizing the residual $x_M \leftarrow \arg\sup_{x \in X_h} |r_M(x)|$.
4. Normalize to construct new reduced basis space function $q_M \leftarrow \frac{r_M}{r_M(x_M)}$.

These steps are repeated until the maximum interpolation error falls beneath a given tolerance. We call this algorithm EIDETAILED in the sequel.

**Adaptation techniques:** The adaptation techniques mentioned in Section 3.1 can also be applied to the empirical interpolation algorithm EIDETAILED, but so far no actual implementation for this is known to us. Supplementary to the adaptive search in the parameter space, we propose to build different collateral reduced basis spaces for different time instant sets $\mathscr{K} \subset \{0, \ldots, K-1\}$. As this time-adaptation strategy is the main focus of this article, we want to give a detailed description of the algorithm:

**procedure** TIMESLICEDEI($\mathscr{W}_{\text{init}}, \mathscr{K}, L_{\text{train}}^{\mathscr{K}}$)
$\mathscr{W}_M \leftarrow$ EIDETAILED($\mathscr{W}_{\text{init}}, L_{\text{train}}^{\mathscr{K}}, M_{\max}, \varepsilon_{tol}$)

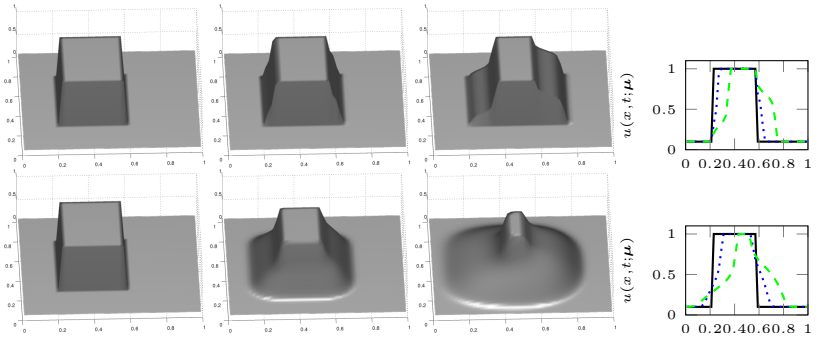


**Fig. 1** Detailed simulation solution snapshots at time instants $t = 0.0$ (first column), $t = 0.1$ (second column), $t = 0.3$ (third column) and for different parameters $\mu = (0, 0.1, 0.4)$ (first row) and $\mu = (2, 0.1, 0.4)$ (second row). The last column shows the reduced solution on cross-sections at $y = 0.5$ for the time instants $t = 0$ (solid line), $t = 0.1$ (dotted line), $t = 0.3$ (dashed line).

**if** $\varepsilon_{tol}$ reached **then**
  $M^k \leftarrow M$ and $\mathscr{W}_{M^k}^k \leftarrow \mathscr{W}_M$ for all $k \in \mathscr{K}$.
**else if** $\mathrm{card}(\mathscr{K}) \leq 2c_{\min}$ **then**
  $\mathscr{W}_{M^k}^k \leftarrow \mathrm{EIDETAILED}(\mathscr{W}_M, L_{\mathrm{train}}^{\mathscr{K}}, \infty, \varepsilon_{tol})$ for all $k \in \mathscr{K}$.
**else** *% maximum number of extensions* $M_{\max}$ *reached*
  $\mathscr{K}_1, \mathscr{K}_2 \leftarrow \mathrm{SPLITTIMEINTERVAL}(\mathscr{K}, \mathscr{W}_M)$
  $\mathrm{TIMESLICEDEI}(\mathscr{W}_M^{\mathscr{K}_1}, L_{\mathrm{train}}^{\mathscr{K}_1})$
  $\mathrm{TIMESLICEDEI}(\mathscr{W}_M^{\mathscr{K}_2}, L_{\mathrm{train}}^{\mathscr{K}_2})$
**end if**
**end procedure**

The training sets $L_{\mathrm{train}}^{\mathscr{K}}$ are restrictions of the full training set $L_{\mathrm{train}}$ to operator evaluations on solutions snapshots at time steps $t^k$ for $k \in \mathscr{K}$. Likewise $\mathscr{W}_M^{\mathscr{K}}$ is a restriction of the discrete space $\mathscr{W}_M$ build only out of solution snapshots with time indices stemming from $\mathscr{K}$. This strategy reduces the computation time, as no computed reduced basis function needs to be thrown away. The method SPLITTIMEINTERVAL splits the interval $\mathscr{K}$ such that afterwards the spaces $\mathscr{W}_M^{\mathscr{K}_1}$ and $\mathscr{W}_M^{\mathscr{K}_2}$ are of equal dimension. The threshold $c_{\min}$ asserts a lower bound on the size of the time intervals.

## 4 Example: Buckley–Leverett equation

We consider a Buckley–Leverett type problem in two space dimensions fulfilling the equations (1)-(2) on a rectangular domain $\Omega := [0,1]^2$ with initial data function $u_0(\mu) = c_{low} + (1 - c_{low})\chi_{[0.2,0.6]\times[0.25,0.75]}$, velocity vector $\mathbf{v}(u;\mu) = (0,1)^t f(u;\mu)$ and diffusion $d(u;\mu) = KD(s;\mu)$. Here $f(u;\mu) = \frac{u^3}{\mu_1} \cdot \left(\frac{u^3}{\mu_1} + \frac{(1-u)^3}{\mu_2}\right)^{-1}$ denotes the fractional flow rate, $D(u;\mu) = \frac{(1-u)^3}{\mu_2} f(u;\mu) p_c'(u;\mu)$ the capillary diffusion for a capillary pressure $p_c(u;\mu) = u^{-\lambda}$. The variable parameters are chosen as $\mu :=$

.

| adaptation | no. of bases | ø-dim(CRB) | offline time[h] | ø-runtime[s] | max. error |
|---|---|---|---|---|---|
| no | 1 | 350 | 1.47 | 6.79 | $5.88 \cdot 10^{-4}$ |
| yes, $c_{\min} = 5$ | 11 | 223.09 | 2.08 | 4.06 | $5.80 \cdot 10^{-4}$ |
| yes, $c_{\min} = 1$ | 26 | 198.42 | 8.40 | 3.38 | $5.75 \cdot 10^{-4}$ |

**Table 1** Comparison of the number of bases, the reduced basis sizes averaged over sub-intervals, offline time, averaged online reduced simulation times and maximum errors for non-adaptive and adaptive runs with threshold $c_{\min} = 5$, and $= 1$. The average online run-times and maximum errors are obtained from 20 simulations with randomly selected parameters $\mu$.

$(K, c_{\text{low}}, \lambda)$ and the parameter space is given by $\mathscr{P} := [1,2] \times [0,0.1] \times [0.1,0.4]$. The scalar viscosities are fixed at $\mu_1 = \mu_2 = 5$. At the boundary of the domain a Dirichlet condition applies with $u_{\mathscr{N}_{\text{dir}}}(\mu) = c_{\text{low}}$.

**Discretization** The problem is discretized with a standard finite volume scheme comprising an explicitly computed Engquist–Osher flux for the convective terms and an implicit discretization of the diffusive terms. The underlying grid has a dimension of $H = 25 \times 25$ grid cells and the time interval $[0, T_{\max}]$ is discretized by 60 uniformly distributed time steps. Fig. 1 illustrates solution snapshots for two different parameters with different diffusion levels $K = 0$ respectively $K = 2$. The cross-section plots in the last column show the expected behaviour of combinations of rarefaction waves and smoothed shocks.

**Offline phase** In order to assess the effects of the adaptation algorithms, the reduced basis algorithms are run three times, once without the time adaptive empirical operator interpolation and two times with adaptation, but different thresholds $c_{\min}$ to bound the time interval size from below. The results concerning reduced basis sizes, offline and reduced simulation time, are summarized in Table 1.

In order to assure that the generated reduced basis leads to equally small reduction errors for all parameters of the parameter space, the parameter training set for the "POD-greedy" algorithm has been adapted with a validation set of randomly chosen parameters $\mu$ in both runs. In the test runs, after three refinement steps the training parameter set comprises 255 elements, and the chosen validation ratio of 1.4 is assured after the maximum error for the training parameters has fallen beneath the targeted level of $5 \cdot 10^{-4}$. The target interpolation error for the empirical interpolation was set to $10^{-6}$ in all runs. This error is reached with an average number of 198 respectively 223 basis functions in the adaptive cases, and 350 basis functions without adaptation. In the adaptive runs, the time interval has been decomposed into 11 respectively 26 sub-intervals (cf. Fig 2(a)&(b)). Fig. 2(c) illustrates the error decrease during the generation of the reduced spaces for selected time intervals (dashed lines) for the run with $c\min = 1$. It can be observed that the slopes for the error graphs are much steeper than in the non-adaptive case illustrated with a dashed line. Because of the larger variation of the solutions for larger time steps, however, the basis on the last interval $[0.29, 0.30]$ still shows the slowest error decrease. Fig.

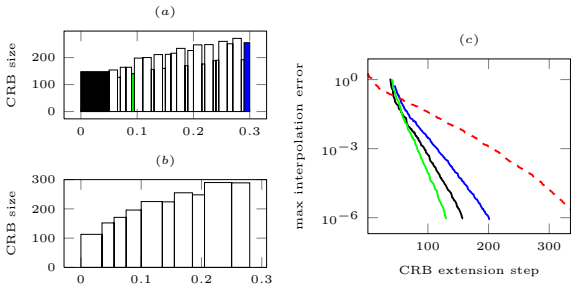


**Fig. 2** Illustration of basis sizes on time intervals after adaptation with (a) $c_{\min} = 1$ and (b) $c_{\min} = 5$. Plot (c) illustrates the error decrease during generation of bases on three intervals marked with the same color in plot (a). The dashed line graph shows the slower decrease for a single basis without adaptation.

2(a+b) show that for both adaptive runs the bases dimensions for all intervals stay significantly below the non-adaptive basis size of 350.

**Conclusion** We observed that the adaptive search in the time domain can lead to faster reduced simulations. However, the costs of 26 generated basis spaces for an average dimension reduction by a factor of approximately 0.56 turned out to be very expensive. We therefore advice to combine the time domain search with a parameter domain search to obtain a further improvement of the method.

**Acknowledgements** This work was supported by the German Science Foundation (DFG) under the contract number OH 98/2-1. The second author was supported by the Baden Württemberg Stiftung gGmbH.

## References

1. Barrault, M., Maday, Y., Nguyen, N., Patera, A.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C. R. Math. Acad. Sci. Paris Series I **339**, 667–672 (2004)
2. Drohmann, M., Haasdonk, B., Ohlberger, M.: Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation. Tech. rep., FB10, University of Münster (2010)
3. Eftang, J.L., Patera, A.T., Rønquist, E.M.: An hp Certified Reduced Basis Method for Parametrized Parabolic Partial Differential Equations. Technical report, MIT, Cambridge, 2009. Submitted to SISC
4. Haasdonk, B. and Ohlberger, M.: Reduced basis method for finite volume approximations of parametrized evolution equations. M2AN Math. Model. Numer. Anal., **42**(2):277-302 (2008)
5. Haasdonk, B., Dihlmann, M., Ohlberger, M.: A training set and multiple bases generation approach for parametrized model reduction based on adaptive grids in parameter space. Tech. rep., University of Stuttgart (submitted) (2010)
6. Haasdonk, B., Ohlberger, M.: Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. In: Proc. 5th International Symposium on Finite Volumes for Complex Applications, pp. 471–478 (2008)
7. Patera, A., Rozza, G.: Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations. MIT (2007). http://augustine.mit.edu/methodology/methodology_bookPartI.htm. Version 1.0, Copyright MIT 2006-2007, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering

The paper is in final form and no similar paper has been or is being submitted elsewhere.