

## REDUCED BASIS METHOD FOR FINITE VOLUME APPROXIMATIONS OF PARAMETRIZED LINEAR EVOLUTION EQUATIONS

BERNARD HAASDONK<sup>1</sup> AND MARIO OHLBERGER<sup>2</sup>

**Abstract.** The model order reduction methodology of *reduced basis (RB)* techniques offers efficient treatment of parametrized partial differential equations (P<sup>2</sup>DEs) by providing both approximate solution procedures and efficient error estimates. RB-methods have so far mainly been applied to finite element schemes for elliptic and parabolic problems. In the current study we extend the methodology to general linear evolution schemes such as finite volume schemes for parabolic and hyperbolic evolution equations. The new theoretic contributions are the formulation of a reduced basis approximation scheme for these general evolution problems and the derivation of rigorous *a-posteriori* error estimates in various norms. Algorithmically, an offline/online decomposition of the scheme and the error estimators is realized in case of affine parameter-dependence of the problem. This is the basis for a rapid online computation in case of multiple simulation requests. We introduce a new offline basis-generation algorithm based on our *a-posteriori* error estimator which combines ideas from existing approaches. Numerical experiments for an instationary convection-diffusion problem demonstrate the efficient applicability of the approach.

**Mathematics Subject Classification.** 76M12, 65M15, 35L90, 35K90, 76R99.

Received November 15, 2006. Revised October 23, 2007.

### 1. INTRODUCTION

Many real world problems can be modelled by parametrized partial differential equations (P<sup>2</sup>DEs), where a parameter vector  $\boldsymbol{\mu}$  characterizes the system in terms of material-, geometry- or control-parameters. In simulation of general P<sup>2</sup>DEs, repeated computation runs for varying parameters  $\boldsymbol{\mu}$  and fast simulation response can be required. Such scenarios occur in design optimization, optimal control with PDE-constraints, online-simulation, parameter identification or state estimation. Model order reduction techniques must be applied to satisfy these time demands. The methodology of *reduced basis (RB)* techniques aims at efficient treatment in such cases by providing both an approximate solution procedure and efficient error estimates.

In the current study we extend the methodology to general parametrized linear evolution equations of the form

$$\partial_t u(t; \boldsymbol{\mu}) + L[u(t; \boldsymbol{\mu}); \boldsymbol{\mu}] = 0$$

---

*Keywords and phrases.* Model reduction, reduced basis methods, finite volume methods, *a-posteriori* error estimates.

<sup>1</sup> Institute of Mathematics, University of Freiburg, Hermann-Herder-Str. 10, 79104 Freiburg, Germany.  
[haasdonk@mathematik.uni-freiburg.de](mailto:haasdonk@mathematik.uni-freiburg.de)

<sup>2</sup> Institute of Numerical and Applied Mathematics, University of Münster, Einsteinstr. 62, 48149 Münster, Germany.  
[mario.ohlberger@math.uni-muenster.de](mailto:mario.ohlberger@math.uni-muenster.de)

with suitable initial data  $u(0; \boldsymbol{\mu}) = u_0(\boldsymbol{\mu})$  and boundary conditions. Here  $L$  denotes an operator which comprises the spatial derivatives,  $u(\boldsymbol{\mu}) : [0, T_{\max}] \rightarrow \mathcal{W}$  is the desired parameter-dependent solution and  $\mathcal{W}$  is a subset of  $L^2(\Omega)$  or some other suitable Hilbert space of functions on a domain  $\Omega$ . Such evolution equations are frequently solved by numerical evolution schemes. These do not only comprise finite element methods (FEM), which represent the main focus in the RB-literature. In particular, our framework covers finite volume (FV) schemes [5,13,20,21] for parabolic and hyperbolic transport equations. Similarly, local discontinuous Galerkin (LDG) methods [2,7,8] can be cast in this framework. Discrete evolution schemes are based on an approximating high-dimensional discrete space  $\mathcal{W}_H \subset L^2(\Omega)$  (or subset of some other Hilbert space) and by approximating the exact solution at discrete time-instants  $t_k$ , *i.e.* providing a sequence of functions  $u_H^k(\boldsymbol{\mu}) \in \mathcal{W}_H$  for  $k = 0, \dots, K$  such that  $u_H^k(\boldsymbol{\mu}) \approx u(t_k; \boldsymbol{\mu})$ . The discrete solution  $u_H(\boldsymbol{\mu})$  then is defined to be piecewise constant in time with  $u_H(t_k; \boldsymbol{\mu}) = u_H^k(\boldsymbol{\mu})$ . The evolution scheme itself proceeds by setting  $u_H^0(\boldsymbol{\mu}) := P[u_0(\boldsymbol{\mu})]$ , where  $P$  is a suitable projection operator, and iteratively solving  $L_I^k[u_H^{k+1}(\boldsymbol{\mu})] = L_E^k[u_H^k(\boldsymbol{\mu})] + b^k(\boldsymbol{\mu})$  for  $u_H^{k+1}(\boldsymbol{\mu})$ . Here  $L_I^k$  and  $L_E^k$  denote ( $\boldsymbol{\mu}$ -dependent) implicit and explicit contributions gathering the discretization of the time derivative and spatial differential operator  $L$  at time  $t_k$ .

The main goal of RB-methods is to provide a reduced simulation scheme, in which for a newly given parameter vector  $\boldsymbol{\mu}$  a function  $u_N(\boldsymbol{\mu})$  is determined as an approximation to the unknown detailed solution  $u_H(\boldsymbol{\mu})$ . In addition or alternatively to the field-variable itself, approximate computation of derived outputs can be desired, *i.e.* determination of  $s_N(\boldsymbol{\mu}) \approx s(u_H(\boldsymbol{\mu}))$  for some output functional  $s$ . The underlying reduced simulation scheme is based on a Galerkin projection of the detailed simulation onto a low-dimensional space  $\mathcal{W}_N \subset \mathcal{W}_H$ , the *reduced basis space* of dimension  $N$ . This space is deliberately determined such that it captures the solution-variety under parameter variations. This basis-construction is frequently based on so called *snapshots*, *i.e.* detailed simulation results  $u_H(\boldsymbol{\mu}_i)$  for stationary or  $u_H(t^{k_i}, \boldsymbol{\mu}_i)$  for time-dependent problems. In addition to the reduced simulation scheme, a second main focus of RB-methods is to provide rigorous *a-posteriori* error bounds for the error between the reduced to the detailed solution or output. These error bounds are in general based on effective computation of residual norms during the reduced simulation and are quite tight. A further important aspect in RB-methods is to make the reduced simulation useful in a multi-query or real-time context. This means that a further goal is to provide an offline/online decomposition of the algorithm. A possibly expensive offline-phase computes auxiliary parameter-independent quantities once in advance, which can subsequently be used in the online-phase, which rapidly sets up the final parameter-dependent simulation matrices for each new  $\boldsymbol{\mu}$ , performs the reduced simulation and provides the error bounds. By this offline/online decomposition the online-phase ideally has a complexity independent of the dimensionality  $H$  of the original non-reduced model, but is typically polynomial in  $N$ , the dimensionality of the RB-space.

RB-methods have been applied widely in the last decades ranging back to early works with applications in structural statics, such as [1] where *global shape functions* are used for spanning the approximating space, or [26], where the notion *reduced basis technique* can already be found. Application of reduced basis methods on time-dependent ordinary differential equations and corresponding *a-priori* error analysis can be found in [31]. These methods have mainly been adopted to finite element schemes for elliptic and parabolic problems. In recent years the RB-methods have gained popularity and developed to a rather extensive methodology. Studies on eigenvalue problems are treated with RB-methods in [22]. The general methodology for linear elliptic equations is the subject of [33], details on algorithmic aspects can be found in [32]. The stationary advection-diffusion problem has been treated in the context of optimal control in [34] and with view on geometry optimization in [39]. The methodology has been successively extended in various directions. Time-dependency of problems is addressed for linear parabolic equations with finite difference time discretization in [15] and more general discontinuous Galerkin time integration in [35]. The treatment of polynomial nonlinearities can be realized by interpreting the nonlinearity as a multilinear form, *e.g.* in case of the stationary viscous Burgers equation [41]. Non-affine parameter dependencies can be handled by approximation methods such as the *empirical interpolation* scheme [4], which also is the basis for more general nonlinearities [14]. The RB-methodology is also applicable to systems of equations such as the incompressible Navier-Stokes equations [40]. In combination with geometry optimization the framework is developed in [37]. An overview of the general methodology

for linear and multilinear cases can be found in [25] or with a larger scope in the recent book [30]. More complex systems of algebro-differential equations can also be treated by Galerkin projection approaches. For instance model reduction of a fuel-cell is successfully performed in [23].

Similar to related studies, the RB-approach in the current presentation will be developed by requiring affine parameter dependence in the data functions and linearity in the evolution equation and evolution scheme. This enables a complete offline/online decomposition which is the basis for the efficient reduced approximation scheme and rigorous *a-posteriori* error control. Although we restrict to the linear setting in the current presentation, this is an important step for the general nonlinear case. The treatment of nonlinearities, which can be caused by nonlinear data functions or nonlinear evolution operators, will similarly rely on linearizations and approximate affine decompositions.

The structure of the present study is as follows. In the next section we introduce the notation and terminology for finite volume schemes. An abstraction into a general implicit/explicit operator formulation is performed. In Section 3 the reduced basis technique is transferred to a class of general time evolution schemes. Different *a-posteriori* error estimates are presented in the subsequent section. Under certain assumptions a complete independence of the dimensionality of the discrete space can be obtained during the reduced basis simulation. This is realized by the offline/online decomposition presented in Section 5. The subsequent section presents a new algorithm for reduced basis construction. Numerical experiments in Section 7 demonstrate the applicability of the approach on a transport problem which (largely simplified) models a fuel cell component. We close with concluding remarks in the last section.

## 2. FINITE VOLUME APPROXIMATION

The general model of the P<sup>2</sup>DE under consideration is the following parametrized evolution equation with general initial data and boundary conditions. Let  $\Omega \subset \mathbb{R}^d$  denote some open convex bounded polygonal space domain with boundary  $\partial\Omega = \Gamma_{\text{dir}} \cup \Gamma_{\text{neu}}$  decomposed into Dirichlet and Neuman components and  $[0, T_{\text{max}}]$  be a time interval. For any parameter vector  $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$  the function  $u(\mathbf{x}, t; \boldsymbol{\mu})$  denotes the solution of the parametrized advection-diffusion equation:

$$\partial_t u(\boldsymbol{\mu}) + \nabla \cdot (\mathbf{v}(\boldsymbol{\mu})u(\boldsymbol{\mu}) - d(\boldsymbol{\mu})\nabla u(\boldsymbol{\mu})) = 0 \quad \text{in } \Omega \times [0, T_{\text{max}}], \tag{2.1}$$

$$u(\cdot, 0; \boldsymbol{\mu}) = u_0(\boldsymbol{\mu}) \quad \text{in } \Omega, \tag{2.2}$$

$$u(\boldsymbol{\mu}) = b_{\text{dir}}(\boldsymbol{\mu}) \quad \text{in } \Gamma_{\text{dir}} \times [0, T_{\text{max}}], \tag{2.3}$$

$$(\mathbf{v}(\boldsymbol{\mu})u(\boldsymbol{\mu}) - d(\boldsymbol{\mu})\nabla u(\boldsymbol{\mu})) \cdot \mathbf{n} = b_{\text{neu}}(u; \boldsymbol{\mu}) \quad \text{in } \Gamma_{\text{neu}} \times [0, T_{\text{max}}]. \tag{2.4}$$

In particular, the initial data  $u_0$  is space-dependent, the velocity field  $\mathbf{v}$ , the diffusion coefficient  $d$ , the boundary value function  $b_{\text{dir}}$  and  $b_{\text{neu}}$  may be space- and time-dependent. The Neuman boundary conditions are assumed to be affine in  $u$  by  $b_{\text{neu}}(u, \mathbf{x}, t; \boldsymbol{\mu}) = b_{\text{neu},1}(\mathbf{x}, t; \boldsymbol{\mu})u(\mathbf{x}, t; \boldsymbol{\mu}) + b_{\text{neu},0}(\mathbf{x}, t; \boldsymbol{\mu})$ . This covers usual flow-conditions such as noflow or outflow. Typical requirements on the data functions for this type of problems in the FV-context are  $\mathbf{v}(\mathbf{x}, t) \in C^1(\bar{\Omega} \times [0, T_{\text{max}}], \mathbb{R}^d)$  with  $\nabla \cdot \mathbf{v}(\cdot, t) = 0$ . We assume an upper bound on the velocity magnitude  $V := \max_{\mathbf{x}, t} \|\mathbf{v}(\mathbf{x}, t)\|$ . Further we assume  $d(\mathbf{x}, t) \in C^2(\bar{\Omega} \times [0, T_{\text{max}}])$ ,  $d(\cdot, \cdot) \geq 0$ . We assume  $u_0 \in L^\infty(\Omega) \cap BV(\Omega)$ ,  $b_{\text{dir}} \in L^\infty(\Gamma_{\text{dir}} \times [0, T_{\text{max}}])$  and  $b_{\text{neu},r} \in L^\infty(\Gamma_{\text{neu}} \times [0, T_{\text{max}}])$  for  $r = 0, 1$ . Here  $C^m(\bar{K})$  for an open set  $K$  denotes the set of functions with continuous derivatives up to order  $m$  in  $K$  which are continuously extendible to the closure  $\bar{K}$ .  $L^\infty(K)$  denotes the set of measurable functions, which have finite  $L^\infty$ -norm:  $\|f\|_\infty := \text{ess sup}_{x \in K} |f(x)|$ .  $BV(K)$  denotes the set of functions of bounded variation, *i.e.* the subset of  $L^1(K)$  with finite BV-seminorm  $\|f\|_{BV} := \sup_{\varphi \in C_0^1(K, \mathbb{R}^d)} \int_K f \nabla \cdot \varphi$ . We mainly list these assumptions for completeness, as we will not make strong use of them, except from referring to the zero-divergence velocity field, the upper bound  $V$  on pointwise velocities and the nonnegativity of the diffusion function. The above quite general formulation covers both parabolic and hyperbolic cases, for which the FV-scheme is well-defined and it covers our model example in the experiments-section. However, existence/uniqueness-results on the analytical problem and convergence of FV-schemes usually apply to special cases. For instance, well-posedness

of the hyperbolic analytical problem with Dirichlet boundary conditions is investigated in [3,18] or revisited in [20], Chapter 6, error estimates are derived in [29]. Convergence results on hyperbolic Cauchy problems are given in [10]. For an analysis of parabolic problems with mixed Dirichlet-Neumann boundary conditions we refer to [16]. For well-posedness of degenerate parabolic equations with Dirichlet boundary conditions we refer to [6], in the case of mixed Dirichlet-Neumann conditions to [19]. Convergence results and error estimates were derived in [11,27].

Finite volume schemes are frequently used instead of finite elements for such partial differential equations. Firstly, they enable to model discontinuous solutions, which can occur in the hyperbolic case and secondly they are motivated by and reflect the conservativity of the problem. Therefore, we focus on the following discrete function space for such equations.

**Definition 2.1** (finite volume space).

- Let  $0 = t_0 < t_1 < \dots < t_K = T_{\max}$  be a sequence of time instants with corresponding time-steps  $\Delta t_k := t_{k+1} - t_k$  for  $k = 1, \dots, K - 1$ .
- $\mathcal{T} = \{T_i\}_{i=1}^H$  denotes a convex polygonal tessellation of  $\Omega$ , *i.e.* a collection of convex polygons with  $\bigcup T_i = \Omega$ ,  $T_i \cap T_j = \emptyset$  for  $i \neq j$ .
- The faces of an element  $T_i$  are denoted as  $e_{ij}$  with  $j$  stemming from a corresponding index set  $j \in \mathcal{N}(i) := \mathcal{N}_{\text{in}}(i) \cup \mathcal{N}_{\text{dir}}(i) \cup \mathcal{N}_{\text{neu}}(i)$  consisting of the disjoint index sets of inner, Dirichlet and Neuman boundary segments. In particular we hereby assume that for all edges either holds  $e_{ij} \subset \bar{\Gamma}_{\text{dir}}$  or  $e_{ij} \subset \bar{\Gamma}_{\text{neu}}$ .
- The center of gravity of  $T_i$  is denoted with  $\mathbf{c}_i$  and  $\mathbf{c}_{ij}$  denotes the center of gravity of the face  $e_{ij}$ . The volume of such entities is denoted with  $|T_i|, |e_{ij}|$ .
- Points  $\mathbf{s}_i \in T_i$  are assumed to exist such that  $\mathbf{s}_i \mathbf{s}_j$  is perpendicular to  $e_{ij}$ , the intersection point is denoted as  $\mathbf{s}_{ij}$ .
- The grid-cell diameters are denoted as  $h_i := \text{diam}(T_i)$  with minimal diameter  $h_{\min} := \min_i h_i$ . We assume to have a geometry constant  $a > 0$  such that  $ah_i^d \leq |T_i|$ ,  $a \sum_{j \in \mathcal{N}(i)} |e_{ij}| \leq h_i^{d-1}$  and  $ah_i \leq |\mathbf{s}_i - \mathbf{s}_j|$  for all  $i$  and  $j \in \mathcal{N}(i)$ .
- $\mathcal{W}_H := \text{span}(\chi_{T_i}) \subset L^\infty(\Omega) \subset L^2(\Omega)$  is the discrete space of cell-wise constant functions, where  $\chi_{T_i}$  are the characteristic functions on the elements  $T_i$ . The space is equipped with the scalar product  $\langle \cdot, \cdot \rangle$  and norm  $\|\cdot\|$  inherited from  $L^2(\Omega)$ .

In the space  $\mathcal{W}_H$ , which is usually of high dimension  $H = \dim(\mathcal{W}_H)$ , the corresponding finite volume schemes can be formulated. We allow general inhomogeneous boundary conditions and mixed implicit/explicit contributions.

**Definition 2.2** (explicit/implicit form of FV-approximation).

- $u_H(\mathbf{x}, t; \boldsymbol{\mu})$  denotes the overall *finite volume (FV) approximation*, which is defined to coincide with  $u_H^k(\mathbf{x}; \boldsymbol{\mu})$  in the time-slab  $[t_k, t_{k+1})$ .
- $u_H^0(\boldsymbol{\mu}) := P[u_0(\boldsymbol{\mu})] \in \mathcal{W}_H$  is the  $L^2$ -projection of the initial data, *i.e.*

$$u_H^0(\mathbf{x}; \boldsymbol{\mu}) := u_i^0 := \frac{1}{|T_i|} \int_{T_i} u_0(\cdot; \boldsymbol{\mu}) \quad \text{for } \mathbf{x} \in T_i. \tag{2.5}$$

- $u_H^{k+1}(\boldsymbol{\mu}) \in \mathcal{W}_H$  is the function obtained by a time evolution step defined by

$$u_H^{k+1}(\mathbf{x}; \boldsymbol{\mu}) := \sum_i \chi_{T_i} u_i^{k+1}, \tag{2.6}$$

where the  $u_i^{k+1} \in \mathbb{R}$  satisfy

$$u_i^{k+1} = u_i^k - \frac{\Delta t_k}{|T_i|} \sum_{j \in \mathcal{N}(i)} h_{ij}^k (u_H^k, u_H^{k+1}; \boldsymbol{\mu}). \tag{2.7}$$

- $h_{ij}^k(u, v; \boldsymbol{\mu})$  denotes a numerical flux, discretizing the convective and diffusive flux from element  $i$  over the edge  $e_{ij}$  if  $u, v \in \mathcal{W}_H$  are the functions at time  $t_k$  and  $t_{k+1}$  respectively.

Under the given non-degeneracy requirements on the tessellation, specific consistency requirements between numerical fluxes and analytical data coefficients and a stability bound on the time-step size, such schemes can be proven to converge to the unique entropy solution (see *e.g.* [10] and the references therein).

The above formulation covers a wide range of numerical fluxes discretizing the convective and diffusive part explicitly, implicitly, or in a mixed way, and includes higher order discretizations by using limiters [20]. For the convective part usual choices are monotone fluxes such as the Lax-Friedrichs, Enquist-Osher or the Godunov flux [5,20]. For the diffusive flux simple difference approximations of the gradient can be performed [20,27], occasionally involving harmonic weighting of cell-averaged diffusivities [10,17], or more general gradient reconstruction techniques [9,12].

For instance, in the experiments we use the explicit Lax-Friedrichs flux for the advective contribution, an implicit discretization of the diffusivity and suitable quadrature rules:

$$h_{ij}^k(u_H^k, u_H^{k+1}; \boldsymbol{\mu}) = g_{ij}^k(u_H^k; \boldsymbol{\mu}) + d_{ij}^k(u_H^{k+1}; \boldsymbol{\mu}), \quad (2.8)$$

$$g_{ij}^k(u_H^k; \boldsymbol{\mu}) = \frac{1}{2}|e_{ij}| \left( \mathbf{v}_{ij} \cdot \mathbf{n}_{ij} (u_j^k + u_i^k) - \frac{1}{\lambda} (u_j^k - u_i^k) \right), \quad (2.9)$$

$$d_{ij}^k(u_H^{k+1}; \boldsymbol{\mu}) = -d(\mathbf{s}_{ij}) \frac{|e_{ij}|}{|\mathbf{s}_i - \mathbf{s}_j|} (u_j^{k+1} - u_i^{k+1}). \quad (2.10)$$

Here  $\lambda$  denotes a coefficient which introduces a stabilizing artificial viscosity and  $\mathbf{v}_{ij}$  is (an approximation of) the average velocity on the edge  $e_{ij}$ . For stability of the method, we ensure that the zero divergence of the velocity field transfers to a discrete analogon, *i.e.*  $\sum_{j \in \mathcal{N}(i)} |e_{ij}| \mathbf{v}_{ij} \cdot \mathbf{n}_{ij} = 0$  for all  $i$ . In our example this will be satisfied by simple midpoint evaluation  $\mathbf{v}_{ij} := \mathbf{v}(\mathbf{c}_{ij})$ , in general a more general quadrature must be used. This choice of fluxes can be applied for all inner edges yielding  $h_{ij}^k$  for  $j \in \mathcal{N}_{\text{in}}(i)$ . The Neuman boundary contribution can be respected by (an approximation of) the face-integral of the Neuman boundary values  $h_{ij}^k := \int_{e_{ij}} b_{\text{neu}}$ , where the value  $u$  required for the evaluation of the Neuman boundary value function is taken as  $u_i^k$  on the boundary edge  $e_{ij}$ . The Dirichlet values at a face  $e_{ij} \subset \bar{\Gamma}_{\text{dir}}$  of a boundary element  $T_i$  can be incorporated *via* constructing a ghost cell  $T_j$  outside of  $\Omega$  by reflecting  $T_i$  with respect to the boundary face  $e_{ij}$ , assigning (an approximation of) the Dirichlet-values  $u_j^k := |e_{ij}|^{-1} \int_{e_{ij}} b_{\text{dir}}$  to these elements and computing the numerical flux  $h_{ij}^k$  as defined for inner cells (*cf.* [29]). In our model example, we perform these edge-integral approximations by point-evaluations in  $\mathbf{c}_{ij}$  or  $\mathbf{s}_{ij}$ . Some properties of the data must be respected here to obtain a reasonable discretization. First, the parameter  $\lambda$  in the Lax-Friedrichs flux must be chosen to satisfy  $\lambda \sup_{i,j} |\mathbf{v}_{ij} \cdot \mathbf{n}_{ij}| \leq 1$  as explained in [20]. This is clearly satisfied by setting  $\lambda := 1/V$ , where  $V$  was the upper bound on the norm of the velocity. Additionally, a CFL-condition for the time-step size must be satisfied such that the explicit evolution-operator is monotone [28]. If diffusive components were discretized explicitly this would lead to the following very restrictive condition  $\Delta t_k < a^3 h_{\text{min}}^2 / (L_g a h_{\text{min}} + \sup_{\mathbf{x}, t} d(\mathbf{x}, t))$ . Here  $a$  and  $h_{\text{min}}$  are the grid-constants defined in Definition 2.2 and  $L_g$  is the Lipschitz-constant of the explicit numerical flux satisfying  $|g_{ij}(u_H) - g_{ij}(u'_H)| \leq L_g |e_{ij}| (|u_i - u'_i| + |u_j - u'_j|)$ . In our case of implicit diffusion discretization, the diffusion term can be ignored and the bound simplifies to the following weaker condition:  $\Delta t_k < a^2 h_{\text{min}} / L_g$ .

Overall, the solution-dependent quantities can be collected into implicit and explicit operators  $L_I^k(\boldsymbol{\mu})$  and  $L_E^k(\boldsymbol{\mu})$  and solution-independent functions  $b^k(\boldsymbol{\mu})$  generated by non-homogeneous boundary values. For this, we define the general class of evolution schemes, for which the RB-approximation and error-estimates will be derived.

**Definition 2.3** (parametrized evolution scheme). We denote a numerical scheme, which produces  $\boldsymbol{\mu}$ -dependent solutions  $u_H^k(\boldsymbol{\mu}) \in \mathcal{W}_H$  for  $k = 0, \dots, K$  a *parametrized evolution scheme*, if there exist continuous linear operators  $P : L^2(\Omega) \rightarrow \mathcal{W}_H$ ,  $L_I^k(\boldsymbol{\mu}) : \mathcal{W}_H \rightarrow \mathcal{W}_H$ ,  $L_E^k(\boldsymbol{\mu}) : \mathcal{W}_H \rightarrow \mathcal{W}_H$  and functions  $b^k(\boldsymbol{\mu}) \in \mathcal{W}_H$  where  $L_I^k$  is

positive definite such that the scheme can be written as

$$u_H^0 = P[u_0(\boldsymbol{\mu})], \tag{2.11}$$

$$L_I^k(\boldsymbol{\mu})[u_H^{k+1}] = L_E^k(\boldsymbol{\mu})[u_H^k] + b^k(\boldsymbol{\mu}). \tag{2.12}$$

We will assume a decomposition of these *time-step operators*  $L_I^k(\boldsymbol{\mu})$  and  $L_E^k(\boldsymbol{\mu})$  as  $L_I^k(\boldsymbol{\mu}) = Id + \Delta t_k \bar{L}_I^k(\boldsymbol{\mu})$  and  $L_E^k(\boldsymbol{\mu}) = Id - \Delta t_k \bar{L}_E^k(\boldsymbol{\mu})$ , where  $\bar{L}_I^k(\boldsymbol{\mu})$  and  $\bar{L}_E^k(\boldsymbol{\mu})$  are denoted *space-discretization operators*.

As mentioned in the introduction, this definition captures a variety of discretizations. In particular the decomposition of the time-step operators is a natural consequence of a simple Euler-forward time-discretization with a mixed implicit/explicit discretization of a spatial differential operator. The positive definiteness of  $L_I^k$  clearly follows in case of a space-discretization operator  $\bar{L}_I^k$  which is psd. But also for non-coercive  $\bar{L}_I^k$  the pd-ness of  $L_I^k$  can be obtained by suitable small time-step  $\Delta t_k$ .

We briefly collect the general properties of the FV-scheme that are satisfied by the above choices of fluxes and discretizations and are used in the sequel for well-definedness of the RB-scheme and computation of the error-estimators. By ensuring these assumptions with other discretizations, the RB-approximation and error-estimators can be applied identically. A further assumption on the data and the finite volume scheme, which is a so called *affine parameter dependence*, will be added in Section 5.

**Lemma 2.1** (properties of FV-approximation). *The FV-discretization according to Definition 2.2 of the parametrized advection-diffusion equation (2.1)–(2.4) with fluxes (2.8)–(2.10) and specified conditions on the data, geometry, time and boundary-integral discretization satisfies the following.*

- (i) *The numerical flux is a sum of a linear implicit, a linear explicit and a constant component, i.e. it has a representation as*

$$h_{ij}^k(u_H^k, u_H^{k+1}; \boldsymbol{\mu}) = h_{ij,E}^k(u_H^k; \boldsymbol{\mu}) + h_{ij,I}^k(u_H^{k+1}; \boldsymbol{\mu}) + h_{ij,b}^k(\boldsymbol{\mu}).$$

- (ii) *The resulting implicit time-step operator  $L_I^k$  is positive definite.*
- (iii) *The resulting implicit space-discretization operator  $\bar{L}_I^k$  is psd.*
- (iv) *The resulting explicit space-discretization operator  $\bar{L}_E^k$  is psd.*

*Proof.* Assumption (i) is obviously satisfied due to the linearity of the PDE and our choice of fluxes for the FV-discretization. Concerning points (ii) and (iii), we give here for completeness the implicit and explicit time-step operators and the offset functions, which result from collecting the components of (i): the corresponding operator matrices for  $L_I^k$  and  $L_E^k$  and the functions  $b^k$  are given by their coefficients (with respect to the basis  $\{\chi_{T_i}\}$ ):

$$(L_I^k)_{il} := \begin{cases} 1 + \frac{\Delta t_k}{|T_i|} \sum_{j \in \mathcal{N}_{in}(i) \cup \mathcal{N}_{dir}(i)} d(\mathbf{s}_{ij}) \frac{|e_{ij}|}{|\mathbf{s}_i - \mathbf{s}_j|} & \text{for } l = i, \\ -\frac{\Delta t_k}{|T_i|} d(\mathbf{s}_{il}) \frac{|e_{il}|}{|\mathbf{s}_i - \mathbf{s}_l|} & \text{for } l \in \mathcal{N}_{in}(i), \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

$$(L_E^k)_{il} := \begin{cases} 1 - \frac{\Delta t_k}{|T_i|} \left( \sum_{j \in \mathcal{N}_{in}(i) \cup \mathcal{N}_{dir}(i)} \frac{|e_{ij}|}{2} [\mathbf{v}(\mathbf{c}_{ij}) \cdot \mathbf{n}_{ij} + \lambda^{-1}] \right. \\ \quad \left. + \sum_{j \in \mathcal{N}_{neu}(i)} |e_{ij}| b_{neu,1}(\mathbf{s}_{ij}) \right) & \text{for } l = i, \\ -\frac{\Delta t_k}{|T_i|} \frac{|e_{il}|}{2} [\mathbf{v}(\mathbf{c}_{il}) \cdot \mathbf{n}_{il} - \lambda^{-1}] & \text{for } l \in \mathcal{N}_{in}(i), \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

$$(b^k)_i := \begin{cases} -\frac{\Delta t_k}{|T_i|} \left( \sum_{j \in \mathcal{N}_{dir}(i)} \left( \frac{|e_{ij}|}{2} [\mathbf{v}(\mathbf{c}_{ij}) \cdot \mathbf{n}_{ij} - \lambda^{-1}] b_{dir}(\mathbf{c}_{ij}) \right. \right. \\ \quad \left. \left. - d(\mathbf{s}_{ij}) \frac{|e_{ij}|}{|\mathbf{s}_i - \mathbf{s}_j|} b_{dir}(\mathbf{s}_{ij}) \right) \right. \\ \quad \left. + \sum_{j \in \mathcal{N}_{neu}(i)} (|e_{ij}| b_{neu,0}(\mathbf{c}_{ij})) \right). \end{cases} \tag{2.15}$$

Here, the data functions of the convective and Neuman boundary contributions are evaluated at times  $t_k$ , the diffusive parts at time  $t_{k+1}$ . Due to the assumption of the nonnegativity of the diffusion function, the implicit space-discretization operator  $\bar{L}_I^k$  is positive semidefinite, which proves (iii), so (ii) the final implicit time-step operator  $L_I^k$  is positive definite. For (iv) it is sufficient to note, that the explicit components have discretely zero divergence and we are using a monotone convective flux. So with the given CFL condition on the time-step size, application of [28], Lemma 3, yields the coercivity of  $\bar{L}_E^k$ , which in particular implies its psd-ness.  $\square$

Note that only (i) and (ii) will be required for well-definedness of the RB-scheme. The other points (iii)-(iv) will enable particular simplifications in the error-estimation procedure. If the latter two points are not satisfied by a given scheme, the RB-approach still is applicable, although with rougher error-predictions. The psd-ness of the space-discretization operators is also valid for upwind fluxes as long as the velocity field is (discretely) divergence free [28]. In the nonzero-divergence case, convective upwind fluxes can lead to space-discretization operators, which then are indeed non-coercive. In these cases (iii) or (iv) will not be valid if the convective flux is discretized implicitly or explicitly.

The proof of the following statement is omitted, as it can be obtained easily from the definitions and the previous lemma.

**Corollary 2.2** (FV-approximation as parametrized evolution scheme). *A finite volume discretization according to Definition 2.2 of the parametrized advection-diffusion equation (2.1)–(2.4) which satisfies the assumptions (i)-(ii) of Lemma 2.1 is a parametrized evolution scheme in the sense of Definition 2.3 by choosing  $P : \mathcal{W} \rightarrow \mathcal{W}_H$  as the orthogonal  $L^2(\Omega)$ -projection, the operators  $L_I^k, L_E^k$  and the offset functions  $b^k$  are defined by gathering the components of  $h_{ij}^k$  of the FV-discretization.*

### 3. REDUCED BASIS APPROXIMATION

In the following, the goal is to provide an efficient scheme for approximating the solutions  $u_H^k \in \mathcal{W}_H$  from a parametrized evolution scheme with approximations  $u_N^k \in \mathcal{W}_N$  in a lower dimensional space and provide *a-posteriori* error control. The FV-discretization will be used in the experiments as one example of such schemes, but the approach is more general. It is identically valid for, *e.g.* finite difference, finite element space discretization, staggered schemes, *etc.*

We denote the solution  $u_H(\boldsymbol{\mu})$  from the evolution scheme as a *high-resolution* or *detailed* solution in contrast to the *reduced basis* solution  $u_N(\boldsymbol{\mu})$  to be defined in this section. We do not target at the error  $u - u_H$  to the true analytical solution, so we assume that the discretization is chosen adequately fine, *i.e.*  $H$  sufficiently high such that the discretization error  $u - u_H$  is negligible compared to the approximation error  $u_N - u_H$ .

The computational complexity of a single time-step in a parametrized evolution scheme is of order  $\mathcal{O}(H^\gamma)$  for some  $\gamma \geq 1$ . In case of a full explicit discretization, *i.e.*  $L_I^k = Id$ , the exponent is typically  $\gamma = 1$  for the single matrix multiplication. For discretizations with implicit contributions, we typically obtain  $1 \leq \gamma \leq 3$  for the solution of the linear system. For a complete solution  $u_H$  these computational demands scale with the number of time-steps  $K$ , which in turn can be estimated in more detail depending on  $H$  and the dimensionality of the space  $\mathbb{R}^d$ . Irrespective of these details, the complexities are in any case highly dependent on  $H$ , so not suitable for an online simulation setting for very large values of  $H$ . This is the general motivation for reduced basis (RB) methods: the goal is an approximate solution strategy, which renders independent of the dimension  $H$  of the high-dimensional space  $\mathcal{W}_H$ . Instead, a low dimensional, problem specific *reduced basis space*  $\mathcal{W}_N$  with dimension  $N$  is constructed based on particular precomputed high-resolution numerical solutions at certain times, so called snapshots. Based on  $\mathcal{W}_N$  the original numerical scheme is approximated.

**Definition 3.1** (reduced basis space).

- Let  $\mathcal{S} := \{(\boldsymbol{\mu}_n, t_{k_n})\}_{n=1}^S$  be a sample of parameters from  $\mathcal{P} \times \{t_k\}_{k=0}^K$ .
- $\mathcal{W}_N \subset \text{span}\{u_H(\cdot, t_{k_n}; \boldsymbol{\mu}_n) | (\boldsymbol{\mu}_n, t_{k_n}) \in \mathcal{S}\}$  denotes an  $N$ -dimensional subspace of  $\mathcal{W}_H$ , the *reduced basis space*.
- $\Phi_N := \{\varphi_n\}_{n=1}^N$  is an orthonormalized basis of  $\mathcal{W}_N$ , a so called *reduced basis*.

Note that we do not use snapshots directly as basis-functions but use orthonormal bases instead, as these are beneficial for the stability of the numerics [37]. Further, we do not take  $\mathcal{W}_N$  as a space spanned by snapshots, but as a subspace of such. By this choice, a compact basis-construction is possible. Details on the algorithms to construct  $\mathcal{W}_N$  are given in Section 6.

The next step in the RB-methodology is dimension reduction by a Galerkin-projection of the problem onto the reduced space. This means rewriting the parametrized evolution scheme in a weak form and restricting both the solution and the test-functions to the space  $\mathcal{W}_N$ . More precisely, functions  $\{u_N^k(\boldsymbol{\mu})\}_{k=0}^K$  are wanted such that for all  $\varphi \in \mathcal{W}_N$  and all  $k = 0, \dots, K-1$  holds

$$\int_{\Omega} (u_N^0 - P[u_0(\boldsymbol{\mu})])\varphi = 0 \quad \text{and} \quad (3.1)$$

$$\int_{\Omega} (L_I^k(\boldsymbol{\mu})[u_N^{k+1}] - L_E^k(\boldsymbol{\mu})[u_N^k] - b^k(\boldsymbol{\mu}))\varphi = 0. \quad (3.2)$$

By expanding the RB-solution into the RB-basis and by using orthonormality of the basis  $\{\varphi_n\}_{n=1}^N$  and linearity of the operators  $P$ ,  $L_E^k$  and  $L_I^k$ , we obtain the following definition of the RB-approximation scheme for parametrized evolution equations.

**Definition 3.2** (explicit/implicit form of RB-approximation). We assume to have a parametrized evolution scheme as in Definition 2.3. Let  $\{u_N^k(\boldsymbol{\mu})\}_{k=0}^K \subset \mathcal{W}_N$  be a sequence of approximate solutions such that  $u_N^k(\mathbf{x}; \boldsymbol{\mu}) := \sum_{n=1}^N a_n^k(\boldsymbol{\mu})\varphi_n(\mathbf{x})$ , where the coefficient vectors  $\mathbf{a}^k := (a_n^k)_{n=1}^N$  satisfy

$$\mathbf{a}^0 = \left( \int_{\Omega} P[u_0(\boldsymbol{\mu})]\varphi_1, \dots, \int_{\Omega} P[u_0(\boldsymbol{\mu})]\varphi_N \right)^T, \quad (3.3)$$

$$\mathbf{L}_I^k(\boldsymbol{\mu})\mathbf{a}^{k+1} = \mathbf{L}_E^k(\boldsymbol{\mu})\mathbf{a}^k + \mathbf{b}^k(\boldsymbol{\mu}), \quad (3.4)$$

and the required matrices and vectors are defined by the entries

$$(\mathbf{L}_I^k(\boldsymbol{\mu}))_{nm} := \int_{\Omega} \varphi_n L_I^k(\boldsymbol{\mu})[\varphi_m], \quad (\mathbf{L}_E^k(\boldsymbol{\mu}))_{nm} := \int_{\Omega} \varphi_n L_E^k(\boldsymbol{\mu})[\varphi_m], \quad (3.5)$$

$$(\mathbf{b}^k(\boldsymbol{\mu}))_n := \int_{\Omega} \varphi_n b^k(\boldsymbol{\mu}). \quad (3.6)$$

Then  $u_N(\mathbf{x}, t; \boldsymbol{\mu})$  denotes a *reduced basis (RB) approximation*, which is defined to coincide with  $u_N^k(\boldsymbol{\mu})$  in the time-slab  $t \in [t_k, t_{k+1})$ .

The well-definedness of the RB-approximation is valid under the assumptions on the operators of the evolution scheme.

**Lemma 3.1** (well-definedness of RB-approximation). *Given a parametrized evolution scheme and a RB-space  $\mathcal{W}_N$ , the RB-approximation  $u_N$  is unique. In particular, the explicit/implicit form in Definition 3.2 is a well-defined construction algorithm.*

*Proof.* The initial data coefficient vector  $\mathbf{a}^0$  is obviously well defined by (3.3). By induction it is sufficient to argue that the RB evolution equation (3.4) is well defined, i.e. a unique  $\mathbf{a}^{k+1}$  can be found for given  $\mathbf{a}^k$ . If  $\mathbf{L}_I^k(\boldsymbol{\mu})$  was not regular, we could find linearly dependent rows, i.e.  $\boldsymbol{\lambda} = (\lambda_n)_{n=1}^N \neq \mathbf{0}$  with  $0 = \sum_n \lambda_n \langle \varphi_n, L_I^k[\varphi_m] \rangle$  for all  $m$ . Multiplying with  $\lambda_m$  and summation over  $m$  yields  $0 = \langle \sum_n \lambda_n \varphi_n, L_I^k[\sum_m \lambda_m \varphi_m] \rangle$ . The positive definiteness of  $L_I^k(\boldsymbol{\mu})$  implies that  $\sum_n \lambda_n \varphi_n = \mathbf{0}$ , which contradicts to the linear independence of the basis-vectors in  $\Phi_N$ .  $\square$

4. A-POSTERIORI ERROR ESTIMATES

The main goal of RB-methods in addition to the efficient approximation method itself is that they aim at tight *a-posteriori* error estimates, which enable to estimate the deviation  $u_N - u_H$  from the unavailable high-resolution solution  $u_H$  during the online-phase. As usual in RB-methods, *a-posteriori* error estimators are obtained by using *a-priori* analysis techniques based on residuals. By exact computation of the residual norms during the reduced simulation, this turns the *a-priori* estimates into *a-posteriori* estimators. We can explicitly derive such error bounds, based on the following representation of the residuals.

**Lemma 4.1** (residual decomposition). *Let  $u_N(\boldsymbol{\mu})$  denote the RB-approximation of a parametrized evolution scheme. For  $k = 0, \dots, K - 1$  we introduce the residuals*

$$R^{k+1}(\boldsymbol{\mu}) := \frac{1}{\Delta t_k} (L_I^k(\boldsymbol{\mu})[u_N^{k+1}] - L_E^k(\boldsymbol{\mu})[u_N^k] - b^k(\boldsymbol{\mu})). \tag{4.1}$$

Then the  $L^2$ -norms of the residuals can be efficiently computed by

$$\|R^{k+1}(\boldsymbol{\mu})\|^2 = \frac{1}{(\Delta t_k)^2} ((\mathbf{a}^{k+1})^T \mathbf{K}_{II}^k \mathbf{a}^{k+1} - 2(\mathbf{a}^{k+1})^T \mathbf{K}_{IE}^k \mathbf{a}^k \tag{4.2}$$

$$+ (\mathbf{a}^k)^T \mathbf{K}_{EE}^k \mathbf{a}^k + m^k - 2(\mathbf{a}^{k+1})^T \mathbf{m}_I^k + 2(\mathbf{a}^k)^T \mathbf{m}_E^k) \tag{4.3}$$

with corresponding matrices, vectors and scalars

$$(\mathbf{K}_{II}^k(\boldsymbol{\mu}))_{nm} := \int_{\Omega} L_I^k[\varphi_n] L_I^k[\varphi_m], \quad (\mathbf{K}_{IE}^k(\boldsymbol{\mu}))_{nm} := \int_{\Omega} L_I^k[\varphi_n] L_E^k[\varphi_m], \tag{4.4}$$

$$(\mathbf{K}_{EE}^k(\boldsymbol{\mu}))_{nm} := \int_{\Omega} L_E^k[\varphi_n] L_E^k[\varphi_m], \quad m^k(\boldsymbol{\mu}) := \int_{\Omega} b^k(\boldsymbol{\mu})^2, \tag{4.5}$$

$$(\mathbf{m}_I^k(\boldsymbol{\mu}))_n := \int_{\Omega} L_I^k[\varphi_n] b^k(\boldsymbol{\mu}), \quad (\mathbf{m}_E^k(\boldsymbol{\mu}))_n := \int_{\Omega} L_E^k[\varphi_n] b^k(\boldsymbol{\mu}). \tag{4.6}$$

*Proof.* The statement follows from squaring (4.1) and applying linearity of the detailed evolution-operators and the definitions from Definition 3.2. □

Based on this residual definition we present two error estimates with different assumptions, error-norms and resulting accuracies. We assume in the following that the operator norms of  $(L_I^k)^{-1}$  and  $L_E^k$  (with respect to the  $L^2$ -norm) can be upper bounded by  $\|(L_I^k)^{-1}\| \leq C_I$  and  $\|L_E^k\| \leq C_E$ , with known (possibly parameter dependent) constants.

First we obtain a  $L^2(\Omega)$ -error estimate for all time-steps, which in principle gives an estimate with respect to the  $L^\infty([0, T_{\max}], L^2(\Omega))$ -norm. It is applicable both in the parabolic and hyperbolic case, it can be applied for full explicit, semi-implicit or full implicit discretizations.

**Proposition 4.2** (*a-posteriori*  $L^2$ -error estimate). *Let  $u_H$  denote the detailed solution of a parametrized evolution scheme and  $u_N$  be the corresponding RB-approximation for a given RB-space  $\mathcal{W}_N$ . We assume that the RB-space contains the initial data,  $u_H^0(\boldsymbol{\mu}) \in \mathcal{W}_N$ . Then, the  $L^2(\Omega)$ -approximation error is bounded for all time-steps  $k$  by*

$$\|u_N^k(\boldsymbol{\mu}) - u_H^k(\boldsymbol{\mu})\| \leq \Delta_N^k(\boldsymbol{\mu}) \tag{4.7}$$

where the bound can be effectively computed by

$$\Delta_N^k(\boldsymbol{\mu}) := \sum_{n=0}^{k-1} \Delta t_n \|R^{n+1}\| (C_E)^{k-1-n} (C_I)^{k-n} \tag{4.8}$$

and simplifies in case of positive semidefinite  $\bar{L}_I^k$  and  $\bar{L}_E^k$  to

$$\Delta_N^k(\boldsymbol{\mu}) := \sum_{n=0}^{k-1} \Delta t_n \|R^{n+1}\|. \tag{4.9}$$

*Proof.* We want to bound the error  $e^k := u_N^k - u_H^k$ . For  $k = 0$  the assumption  $u_H^0 \in \mathcal{W}_N$  implies the claimed bound as  $\|e^0\| = 0$ . The definitions of the residual (4.1) and the exact evolution scheme (2.11), (2.12) allow rewriting

$$\begin{aligned} L_I^k[e^{k+1}] &= L_I^k[u_N^{k+1}] - L_I^k[u_H^{k+1}] \\ &= \Delta t_k R^{k+1} + L_E^k[u_N^k] + b^k - L_I^k[u_H^{k+1}] \\ &= \Delta t_k R^{k+1} + L_E^k[u_N^k] - L_E^k[u_H^k]. \end{aligned}$$

Thus, the error satisfies an evolution equation

$$L_I^k[e^{k+1}] = \Delta t_k R^{k+1} + L_E^k[e^k]. \tag{4.10}$$

With continuity of the operators we obtain

$$\|e^{k+1}\| \leq \|(L_I^k)^{-1}\| (\Delta t_k \|R^{k+1}\| + \|L_E^k\| \|e^k\|) \leq \Delta t_k C_I \|R^{k+1}\| + C_I C_E \|e^k\|.$$

Resolving this recursion yields the proposed bound (4.7)-(4.8). In case of  $\bar{L}_I^k(\boldsymbol{\mu})$  being psd (but in general nonsymmetric), the upper bound on the continuity constant of  $(L_I^k)^{-1}$  can be chosen as  $C_I := 1$ . This can be concluded if we show that all eigenvalues of  $L := ((L_I^k)^{-1})^{ad} \circ (L_I^k)^{-1}$  have magnitude less than 1. Then its spectral radius and hence the continuity constant of  $(L_I^k)^{-1}$  is upper bounded by 1. As  $L$  is self-adjoint, it is sufficient to show that all eigenvalues of  $L^{-1} = (L_I^k) \circ (L_I^k)^{ad}$  have magnitude larger than 1. Assuming that we have an eigenvalue  $\lambda$  and corresponding eigenfunction  $v$  with  $\|v\|_{L^2(\Omega)} = 1$ , we use the decomposition  $L_I^k = Id + \Delta t_k \bar{L}_I^k$  to obtain

$$\lambda^2 = \|L^{-1}[v]\|^2 = \|v + \Delta t_k (\bar{L}_I^k + (\bar{L}_I^k)^{ad})[v] + (\Delta t_k)^2 \bar{L}_I^k \circ (\bar{L}_I^k)^{ad}[v]\|^2.$$

With the abbreviation  $M := (\bar{L}_I^k + (\bar{L}_I^k)^{ad}) + \Delta t_k \bar{L}_I^k \circ (\bar{L}_I^k)^{ad}$  we simplify this to

$$\lambda^2 = \|v + \Delta t_k M[v]\|^2 = \|v\|^2 + \Delta t_k \langle v, (M + M^{ad})[v] \rangle + (\Delta t_k)^2 \|M[v]\|^2 \geq 1$$

as  $M$  and  $M^{ad}$  are psd. In a similar way, one can see that in case of  $\bar{L}_E^k$  being psd, we can choose the upper bound on the continuity constant of  $L_E^k$  as  $C_E := 1$ . This implies the simplified bound (4.9).  $\square$

The latter condition of psd-ness of the space-discretization operators  $\bar{L}_I^k$  and  $\bar{L}_E^k$  is in particular satisfied by our choice of discretization in Section 2 and Lemma 2.1, assumptions (iii) and (iv).

Estimates for the error measured in energy-norms can be provided if coercivity of  $\bar{L}_I^k$  is assumed. These estimates are not applicable to the hyperbolic case (vanishing diffusivity) or in case of pure explicit discretizations. But they measure the error in a natural norm for cases where the implicit space-discretization operator is coercive. The following is a generalization of the estimate in [15], where we additionally treat the explicit operator contributions and emphasize the applicability to different norms by introducing a parameter  $\gamma$ . This parameter can be chosen in applications to optimize the effectivity of the bounds. In particular by  $\gamma = 0$  we obtain an estimate for the  $L^2(\Omega)$ -error, for  $\gamma = 1$  we obtain the norm as used in the given reference. We additionally require in the following that  $C_E \leq 1$ . As seen in the proof of the previous estimate, this choice is satisfied, if the explicit space-discretization operator  $\bar{L}_E^k$  is positive semidefinite.

**Proposition 4.3** (*a-posteriori* weighted energy-error estimate). *Let  $u_H$  denote the detailed solution of a parametrized evolution scheme and  $u_N$  be the corresponding RB-approximation for a given RB-space  $\mathcal{W}_N$ . We assume that the RB-space contains the initial data,  $u_H^0(\boldsymbol{\mu}) \in \mathcal{W}_N$ . If the explicit time-step operator bound satisfies  $C_E \leq 1$  and the implicit operator contribution  $\bar{L}_I^k$  is coercive with coercivity constant  $\alpha > 0$ , then we define an operator dependent norm for functions  $v = \{v^k\}_{k=1}^K \subset (\mathcal{W}_H)^K$  by*

$$|v|_I := \left( \sum_{k=0}^{K-1} \Delta t_k \langle v^{k+1}, \bar{L}_I^k[v^{k+1}] \rangle \right)^{1/2}.$$

With the abbreviation  $C := ((1 - C_E^2)^{1/2} + 1)/2$  we define for any weight  $\gamma \in [0, C^{-1})$  a spatio-temporal energy-norm<sup>1</sup> by

$$\|v\|_\gamma := \left( \|v^K\|^2 + \gamma |v|_I^2 \right)^{1/2}.$$

Then, the error can be bounded by

$$\|u_N(\boldsymbol{\mu}) - u_H(\boldsymbol{\mu})\|_\gamma \leq \bar{\Delta}_{N,\gamma}^K(\boldsymbol{\mu}) \quad \text{with} \quad (4.11)$$

$$\bar{\Delta}_{N,\gamma}^K(\boldsymbol{\mu}) := \left( \frac{1}{4\alpha C(1-\gamma C)} \sum_{k=0}^{K-1} \Delta t_k \|R^{k+1}\|^2 \right)^{1/2}. \quad (4.12)$$

*Proof.* The proof is a modification of [15]. As before, we start with the evolution equation for the error (4.10). Testing with  $e^{k+1}$ , applying the Cauchy-Schwarz and operator estimates yields

$$\langle e^{k+1}, L_I^k[e^{k+1}] \rangle \leq C_E \|e^{k+1}\| \|e^k\| + \Delta t_k \|R^{k+1}\| \|e^{k+1}\|.$$

We apply Young's inequality  $|c||d| \leq \frac{1}{2\rho^2}c^2 + \frac{1}{2}\rho^2d^2$  with  $\rho^2 = C_E^{-1} + \sqrt{C_E^{-2} - 1}$ ,  $c = \|e^{k+1}\|$  and  $d = \|e^k\|$  on the first term of the right hand side and obtain with the definition of  $C$

$$C_E \|e^{k+1}\| \|e^k\| \leq \frac{C_E}{2\rho^2} \|e^{k+1}\|^2 + \frac{C_E\rho^2}{2} \|e^k\|^2 = (-C + 1) \|e^{k+1}\|^2 + C \|e^k\|^2.$$

This and the claimed decomposition  $L_I^k = Id + \Delta t_k \bar{L}_I^k$  can correspondingly be used to rewrite the previous inequality

$$C \|e^{k+1}\|^2 - C \|e^k\|^2 + \Delta t_k \langle e^{k+1}, \bar{L}_I^k[e^{k+1}] \rangle \leq \Delta t_k \|R^{k+1}\| \|e^{k+1}\|. \quad (4.13)$$

Second application of Young's inequality with  $\rho^2 = 2\alpha(1-\gamma C)$ ,  $c = \|R^{k+1}\|$  and  $d = \|e^{k+1}\|$  and coercivity of  $\bar{L}_I^k$  allows to estimate the right hand term

$$\Delta t_k \|R^{k+1}\| \|e^{k+1}\| \leq \frac{\Delta t_k}{4\alpha(1-\gamma C)} \|R^{k+1}\|^2 + \Delta t_k(1-\gamma C) \langle e^{k+1}, \bar{L}_I^k[e^{k+1}] \rangle.$$

Combining this with (4.13) and division by  $C$  yields the inequality

$$\|e^{k+1}\|^2 - \|e^k\|^2 + \gamma \Delta t_k \langle e^{k+1}, \bar{L}_I^k[e^{k+1}] \rangle \leq \frac{\Delta t_k}{4\alpha(1-\gamma C)C} \|R^{k+1}\|^2.$$

<sup>1</sup>Note that for  $\gamma = 0$  it actually only is a seminorm.

Summation over  $k = 0, \dots, K - 1$  results in

$$\|e^K\|^2 - \|e^0\|^2 + \gamma|e|_I^2 \leq \frac{1}{4\alpha(1-\gamma C)C} \sum_{k=0}^{K-1} \Delta t_k \|R^{k+1}\|^2.$$

As  $e^0 \equiv 0$ , this is the claimed bound (4.11)-(4.12).  $\square$

Some remarks on these results are in order. First, both are a rigorous means for error control which are completely independent of the dimensionality  $H$  as long as the  $\mu$ -dependent quantities (4.4)–(4.6),  $\alpha(\mu)$  and  $C_I, C_E$  are available. This means that either prior-knowledge or explicit formulas for these problem-specific quantities must be available, or additional online-algorithms must be applied which provide upper bounds for the continuity constants or lower bounds for the coercivity constant *e.g.* Section 4.2 in [30]. In these cases the error-estimator can be monitored during RB-simulation and used for predicting reliable accuracies or selecting basis-functions as applied in the next section.

Note that the growing of the first bound  $\Delta_N^k$  with  $k$  is not exponential, but at most linear if  $C_E \leq 1$  and the residual norms remain bounded. The growing of the second bound  $\bar{\Delta}_{N,\gamma}^k$  is of order  $k^{1/2}$ , so asymptotically, this bound is more tight. However, the estimates are useful in different parameter domains. The former is informative for non-asymptotic time-ranges and the hyperbolic or weakly parabolic case. As soon as the coercivity of the problem is large, the latter estimate is accurate due to the inverse proportional dependence of (4.12) on  $\alpha$ . As both estimators are cheap to compute during simulation, the minimum of both can be taken as a combined  $L^2$ -error estimator, *i.e.*  $\tilde{\Delta}_N^k := \min(\Delta_N^k, \bar{\Delta}_{N,\gamma}^k)$  for  $\gamma = 0$ . These estimators certainly do not cover the whole landscape of possible estimators. For instance there might exist  $L^2$ -error estimators for the non-coercive case, which have similar asymptotical behaviour than the energy-error estimator. Similarly, due to our weak assumptions on the space-discretization operators, it might be that for certain classes of problems better estimators can be derived.

The error estimates emphasize the need for good approximation of the initial data. If the initial data is not contained in  $\mathcal{W}_N$ , additional error terms are appearing in the error bounds, which makes them unnecessarily larger and dependent on  $H$ . This inhibits an offline/online decomposition as described in the next sections. Therefore, inclusion of the initial data  $P[u_0(\mu)]$  in the RB-space is beneficial. The inclusion of these possibly highly varying parameter dependent initial-data projections in  $\mathcal{W}_N$  can effectively be realized by assuming an affine parameter dependence of the initial data. We refrain from giving details here but clarify this aspect in Section 6.

## 5. OFFLINE/ONLINE DECOMPOSITION

The main goal of RB-methods is to provide a fast simulation-phase, where multiple new, formerly unseen parameter choices  $\mu$  are given, and fast simulation output including error bounds are required. Ideally, the online RB-simulation is completely independent of the dimensionality  $H$  of the detailed discretization space  $\mathcal{W}_H$ .

For time-independent scenarios (*i.e.* the operators in the evolution scheme are  $k$ -independent), the method up to now is already largely satisfactory, as only few operations depend on  $H$ : for a given reduced basis and a new parameter  $\mu$ , the  $H$ -dependent but time-independent quantities (3.3), (3.5), (3.6) and (4.4)–(4.6) are computed once before the RB-simulation and  $\alpha(\mu)$ ,  $C_E$  and  $C_I$  are determined. The approximate time evolution itself can be performed  $H$ -independently, *i.e.* all quantities (4.2), (4.3), (4.8) and (4.12) can be computed very efficiently.

The RB-methodology provides means for even resolving this marginal  $H$ -dependence and treating time-dependent problems. This is obtained by an offline/online decomposition, where the reduced basis and all  $H$ -dependent quantities are computed offline, while in the online phase, *i.e.* given a new  $\mu$ , all computations are completely independent of  $H$ . This has been realized in RB-FEM-approaches as mentioned in the introduction, now an analogous decomposition can be performed in the case of parametrized evolution schemes. For this we assume an affine parameter dependence of all data functions, as done for elliptic or parabolic problems, *e.g.* [15,33].

**Definition 5.1** (affine parameter decomposition of data). The data functions of an advection-diffusion-problem allow an *affine parameter decomposition*, if they can be decomposed into finite sums of products of parameter-independent and space-independent functions:

$$\mathbf{v}(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_v} \mathbf{v}^q(\mathbf{x}, t) \sigma_v^q(\boldsymbol{\mu}), \quad \mathbf{d}(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_d} \mathbf{d}^q(\mathbf{x}, t) \sigma_d^q(\boldsymbol{\mu}), \quad (5.1)$$

$$u_0(\mathbf{x}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_{u_0}} u_0^q(\mathbf{x}) \sigma_{u_0}^q(\boldsymbol{\mu}), \quad b_{\text{dir}}(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_{b_{\text{dir}}}} b_{\text{dir}}^q(\mathbf{x}, t) \sigma_{b_{\text{dir}}}^q(\boldsymbol{\mu}) \quad (5.2)$$

$$\text{and } b_{\text{neu},r}(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_{b_{\text{neu},r}}} b_{\text{neu},r}^q(\mathbf{x}, t) \sigma_{b_{\text{neu},r}}^q(\boldsymbol{\mu}) \quad \text{for } r = 0, 1. \quad (5.3)$$

Note that this is not a severe restriction, as any smooth function can be approximated in this way, *e.g.* by Taylor expansion or more sophisticated *empirical interpolation* [4]. It is not crucial that the coefficient functions  $\sigma_{\cdot}^q$  are constant in time, this is only assumed for simplifying the presentation. Similar to the decomposition of the data, we define an affine decomposition for the evolution operators. This is the main assumption on a given parametrized evolution scheme, which must be ensured. We will demonstrate this for FV-discretizations in Lemma 5.2 below.

**Definition 5.2** (affine parameter decomposition of parametrized evolution scheme). The parametrized evolution scheme allows an *affine parameter decomposition* if it can be written as

$$P[u_0(\boldsymbol{\mu})] = \sum_{q=1}^{Q_P} P[u_0]^q \sigma_P^q(\boldsymbol{\mu}), \quad L_I^k(\boldsymbol{\mu})[u] = \sum_{q=1}^{Q_{L_I}} L_I^{k,q}[u] \sigma_{L_I}^q(\boldsymbol{\mu}), \quad (5.4)$$

$$b^k(\mathbf{x}, \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} b^{k,q}(\mathbf{x}) \sigma_b^q(\boldsymbol{\mu}), \quad L_E^k(\boldsymbol{\mu})[u] = \sum_{q=1}^{Q_{L_E}} L_E^{k,q}[u] \sigma_{L_E}^q(\boldsymbol{\mu}). \quad (5.5)$$

The decomposition of the evolution scheme directly implies a corresponding decomposition for the RB-operators and auxiliary quantities for the error estimators. The following lemma is a direct consequence of the previous definitions.

**Lemma 5.1** (affine parameter decomposition of the RB-scheme). *If the parametrized evolution scheme allows an affine parameter decomposition, the RB-scheme can similarly be decomposed as*

$$\mathbf{a}^0(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{u_0}} \mathbf{a}^{0,q} \sigma_P^q(\boldsymbol{\mu}), \quad \mathbf{L}_I^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_I}} \mathbf{L}_I^{k,q} \sigma_{L_I}^q(\boldsymbol{\mu}), \quad (5.6)$$

$$\mathbf{b}^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \mathbf{b}^{k,q} \sigma_b^q(\boldsymbol{\mu}), \quad \mathbf{L}_E^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_E}} \mathbf{L}_E^{k,q} \sigma_{L_E}^q(\boldsymbol{\mu}), \quad (5.7)$$

with definition of  $\mathbf{a}^{0,q}, \mathbf{L}_I^{k,q}, \mathbf{L}_E^{k,q}, \mathbf{b}^{k,q}$  analogous to (3.3) and (3.5)-(3.6). Similarly, the quantities required for the error estimate can be decomposed as

$$\mathbf{K}_{II}^k(\boldsymbol{\mu}) = \sum_{q,q'=1}^{Q_{L_I}} \mathbf{K}_{II}^{k,q,q'} \sigma_{L_I}^q(\boldsymbol{\mu}) \sigma_{L_I}^{q'}(\boldsymbol{\mu}), \quad \mathbf{m}_I^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_I}} \sum_{q'=1}^{Q_b} \mathbf{m}_I^{k,q,q'} \sigma_{L_I}^q(\boldsymbol{\mu}) \sigma_b^{q'}(\boldsymbol{\mu}), \quad (5.8)$$

$$\mathbf{K}_{IE}^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_I}} \sum_{q'=1}^{Q_{L_E}} \mathbf{K}_{IE}^{k,q,q'} \sigma_{L_I}^q(\boldsymbol{\mu}) \sigma_{L_E}^{q'}(\boldsymbol{\mu}), \quad \mathbf{m}_E^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_E}} \sum_{q'=1}^{Q_b} \mathbf{m}_E^{k,q,q'} \sigma_{L_E}^q(\boldsymbol{\mu}) \sigma_b^{q'}(\boldsymbol{\mu}), \quad (5.9)$$

$$\mathbf{K}_{EE}^k(\boldsymbol{\mu}) = \sum_{q,q'=1}^{Q_{L_E}} \mathbf{K}_{EE}^{k,q,q'} \sigma_{L_E}^q(\boldsymbol{\mu}) \sigma_{L_E}^{q'}(\boldsymbol{\mu}), \quad m^k(\boldsymbol{\mu}) = \sum_{q,q'=1}^{Q_b} m^{k,q,q'} \sigma_b^q(\boldsymbol{\mu}) \sigma_b^{q'}(\boldsymbol{\mu}), \quad (5.10)$$

with definitions of the  $\mathbf{K}_{II}^{k,q,q'}, \mathbf{K}_{IE}^{k,q,q'}, \mathbf{K}_{EE}^{k,q,q'}, \mathbf{m}_I^{k,q,q'}, \mathbf{m}_E^{k,q,q'}, m^{k,q,q'}$  analogous to (4.4)–(4.6).

With these decompositions the offline/online procedure and herewith the goal of fast online phase for varying  $\boldsymbol{\mu}$  can be realized. The price for this fast online stage is the offline-stage, in which auxiliary quantities are computed. This can be both time and memory intensive. We will quantify this in the experiments section.

We now specify the properties of our FV-discretization in addition to those of Lemma 2.1 which guarantee the applicability of the previous result. This means, we give specific conditions which guarantee the transfer of the affine decomposition of the data to the affine decomposition of the finite volume operators. Then due to the previous lemma, the affine parameter dependence of the RB-scheme and error estimators follows.

**Lemma 5.2** (affine parameter decomposition of FV-evolution scheme). *We assume to have a FV-discretization according to Definition 2.2 of the parametrized advection-diffusion equation (2.1)–(2.4) with affine parameter dependent data functions in the sense of Definition 5.1.*

- (1) *The choice of fluxes (2.8)–(2.10) and specified discretizations of the edge-integrals imply the following.*
  - (i) *The affine decomposition of the velocity field  $\mathbf{v}$  transfers to the velocity discretization  $\mathbf{v}_{ij}$ .*
  - (ii) *The affine decomposition of the boundary value functions transfers to the discretization of their boundary integrals.*
  - (iii) *The affine decomposition of the velocity discretization, the diffusivity and boundary discretizations transfer to the numerical flux  $h_{ij}^k(u_H^k, u_H^{k+1})$ , such that all components are constant or linear in either  $u_H^k$  or  $u_H^{k+1}$ :*

$$h_{ij}^k(u_H^k, u_H^{k+1}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_{h_I}} \sigma_{h_I}^q(\boldsymbol{\mu}) h_{ij,I}^q(u_H^{k+1}) + \sum_{q=1}^{Q_{h_E}} \sigma_{h_E}^q(\boldsymbol{\mu}) h_{ij,E}^q(u_H^k) + \sum_{q=1}^{Q_{h_b}} \sigma_{h_b}^q(\boldsymbol{\mu}) h_{ij,b}^q. \quad (5.11)$$

- (2) *Any choice of fluxes and integral discretization, that satisfies above properties (i)–(iii) and the assumptions (i)–(ii) of Lemma 2.1 leads to a parametrized evolution scheme, that is affinely parameter dependent in sense of Definition 5.2.*

*Proof.* (1) Points (i) and (ii) are clearly obtained by any quadrature for the edge integrals based on point-evaluations, e.g.  $\mathbf{v}_{ij}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_v} \mathbf{v}_q^k(\mathbf{c}_{ij}) \sigma_v(\boldsymbol{\mu})$  and similarly for the boundary values. The decomposition in (iii) is resulting from separately regarding the implicit, explicit and constant components. We only discretize the diffusivity in an implicit way (2.10), which gives  $Q_{h_I} := Q_d$  components, as the decomposition of the diffusivity and corresponding coefficients simply carry through into the first sum. We performed an explicit discretization of the convection (2.9) and the Neuman boundary-conditions. This gives  $Q_{h_E} := Q_{b_{\text{neu},1}} + Q_v + 1$  components  $h_{ij,E}^q$  due to the additional artificial viscosity term in the Lax-Friedrichs flux. Of course, on each edge  $e_{ij}$  either the convective contributions or the Neuman boundary contributions are nonzero, so in principle this number of components could be reduced. This redundant representation, however, allows a uniform treatment of all edges.

The constant components  $h_{ij,b}^q$  result on one hand from Dirichlet-boundary contributions. The value  $u_j^k$  in the numerical flux functions must be replaced by the affine decomposition of the  $b_{\text{dir}}$  boundary-integral, which results in a multiplicatively enlarged number of components. This means  $(Q_{\mathbf{v}} + 1)Q_{b_{\text{dir}}}$  for convective flux components,  $Q_d Q_{b_{\text{dir}}}$  for diffusive flux components. On the other hand, constant components stem from the constant Neuman-function part, *i.e.*  $Q_{b_{\text{neu},0}}$ , which gives in total  $Q_{h_b} := Q_{b_{\text{dir}}}Q_d + Q_{b_{\text{dir}}}(Q_{\mathbf{v}} + 1) + Q_{b_{\text{neu},0}}$  components.

(2) Due to linearity of the initial-data projection  $P$ , this allows a parameter decomposition using  $P[u_0]^q = P[u_0^q]$  and  $\sigma_P^q = \sigma_{u_0}^q$ . Inserting equation (5.11) from assumption (iii) into (2.7) and noting, that the coefficients  $\sigma_{h_E}^q(\boldsymbol{\mu})$ , *etc.*, do not depend on  $j$  allows to change the order of the double-sums:

$$u_i^{k+1} = u_i^k - \sum_{q=1}^{Q_{h_E}} \sigma_{h_E}^q(\boldsymbol{\mu}) \sum_{j \in \mathcal{N}(i)} \frac{\Delta t_k}{|T_i|} h_{ij,E}^k(u_H^k) - \sum_{q=1}^{Q_{h_I}} \sigma_{h_I}^q(\boldsymbol{\mu}) \sum_{j \in \mathcal{N}(i)} \frac{\Delta t_k}{|T_i|} h_{ij,I}^k(u_H^k) - \sum_{q=1}^{Q_{h_b}} \sigma_{h_b}^q(\boldsymbol{\mu}) \sum_{j \in \mathcal{N}(i)} \frac{\Delta t_k}{|T_i|} h_{ij,b}^k(u_H^k).$$

By sorting the implicit, explicit and constant contributions into the operators  $L_E^k$  and  $L_I^k$  and functions  $b^k$  we obtain an affine decomposition of the evolution scheme with  $Q_{L_E} = Q_{h_E} + 1$ ,  $Q_{L_I} = Q_{h_I} + 1$  and  $Q_b := Q_{h_b}$  components.  $\square$

More general quadratures and fluxes also satisfy these assumptions (i)–(iii). One notable point is that even upwind-fluxes can be used, although they are in general highly nonlinear due to the inherent *switching* depending on the direction of  $\mathbf{v}$ . If the sign of  $\mathbf{v}_{ij}(\boldsymbol{\mu}) \cdot \mathbf{n}_{ij}$  does not change under variation of  $\boldsymbol{\mu}$ , these numerical fluxes are also linear in  $u_H$ , so the affine decomposition also directly transfers to these fluxes. For instance if  $\theta := \min_{ij} |\arcsin((\mathbf{v}_{ij} \cdot \mathbf{n}_{ij}) / \|\mathbf{v}_{ij}\|)| > 0$  is the minimum angle between the discretized velocities and corresponding edges, arbitrary direction changes of  $\mathbf{v}_{ij}$  in the range  $(-\theta, \theta)$  and magnitude variations by positive scalings are possible while using upwind fluxes.

### 5.1. Computational steps of the offline-phase

Given a reduced basis  $\Phi_N$ , the offline computation consists of the following  $\boldsymbol{\mu}$ -independent steps, which involve expensive operations of complexity  $H$ :

#### 1. Data components $\implies$ evolution scheme components:

Based on the affine decomposition components of the data  $(\mathbf{v}^q, d^q, b_{\text{dir}}^q, u_0^q, b_{\text{neu},1}^q, b_{\text{neu},0}^q)$ , the affine decomposition components of the evolution operators  $(L_I^{k,q}, L_E^{k,q})$  and functions  $b^{k,q}$ ,  $P[u_0]^q$  are computed.

#### 2. Evolution scheme components $\implies$ RB-scheme components:

Based on the evolution components  $(L_I^{k,q}, L_E^{k,q}, b^{k,q}, P[u_0]^q)$ , the RB-scheme components  $(\mathbf{L}_I^{k,q}, \mathbf{L}_E^{k,q}, \mathbf{b}^{k,q}, \mathbf{a}^{0,q})$  are computed.

#### 3. Evolution scheme components $\implies$ error estimator components:

Based on the evolution components  $(L_I^{k,q}, L_E^{k,q}, b^{k,q}, P[u_0]^q)$ , the components for the error estimation  $(\mathbf{K}_{II}^{k,q,q'}, \mathbf{K}_{IE}^{k,q,q'}, \mathbf{K}_{EE}^{k,q,q'}, \mathbf{m}_I^{k,q,q'}, \mathbf{m}_E^{k,q,q'}, m^{k,q,q'})$  are computed.

### 5.2. Computational steps of the online-phase

Given the precomputed quantities from the offline phase and a new parameter  $\boldsymbol{\mu}$ , the following steps are performed without any need of operations of complexity  $H$ :

#### 1. Data coefficients $\implies$ evolution scheme coefficients:

Based on the affine decomposition coefficients of the data  $(\sigma_{\mathbf{v}}^q(\boldsymbol{\mu}), \sigma_d^q(\boldsymbol{\mu}), \sigma_{b_{\text{dir}}}^q(\boldsymbol{\mu}), \sigma_{u_0}^q(\boldsymbol{\mu}), \sigma_{b_{\text{neu},1}}^q(\boldsymbol{\mu}), \sigma_{b_{\text{neu},0}}^q(\boldsymbol{\mu}))$ , the affine decomposition coefficients of the evolution operators  $(\sigma_{L_I}^q(\boldsymbol{\mu}), \sigma_{L_E}^q(\boldsymbol{\mu}))$  and functions  $(\sigma_b^q(\boldsymbol{\mu}), \sigma_P^q(\boldsymbol{\mu}))$  are computed.

**2. Evolution scheme coefficients  $\implies$  RB-scheme assembling:**

Based on the evolution coefficients, the RB-operators and functions  $(\mathbf{a}^0(\boldsymbol{\mu}), \mathbf{L}_I^k(\boldsymbol{\mu}), \mathbf{L}_E^k(\boldsymbol{\mu}), \mathbf{b}^k(\boldsymbol{\mu}))$  are assembled following (5.6)–(5.7).

**3. Evolution scheme coefficients  $\implies$  error estimator assembling:**

Based on the evolution coefficients the auxiliary quantities for the error estimation  $(\mathbf{K}_{II}^k(\boldsymbol{\mu}), \mathbf{K}_{IE}^k(\boldsymbol{\mu}), \mathbf{K}_{EE}^k(\boldsymbol{\mu}), \mathbf{m}_I^k(\boldsymbol{\mu}), \mathbf{m}_E^k(\boldsymbol{\mu}), m^k(\boldsymbol{\mu}))$  are assembled following (5.8)–(5.10).

Based on these components, the RB-online simulation (3.3)–(3.4), residual computation (4.2), (4.3) and error estimation according to (4.8), (4.9) and/or (4.12) can be performed.

## 6. REDUCED BASIS CONSTRUCTION

The prerequisite of the preceding sections was the availability of a reduced basis  $\Phi_N$ . This choice determines the quality of the RB-simulation and error estimates, which means the algorithmic efficiency and numerical accuracy. Too few basis-vectors cannot guarantee a good solution and small predictive error, too many basis-vectors will deteriorate the computational complexity for the online computations.

We want to propose a method for incremental basis-construction (or incremental basis-refinement) by improving on an approach proposed in [15]. For simplification of presentation, we restrict ourselves to the  $L^2$ -error and error estimator  $\Delta_N^k$ . The method transfers identically to energy-error and estimators. The desirable goal is to construct for any given tolerance  $\epsilon > 0$  a basis  $\Phi_N$  with preferably small  $N$  such that for all  $\boldsymbol{\mu} \in \mathcal{P}$  and all  $k = 0, \dots, K$  holds  $\|u_H^k - u_N^k\| \leq \epsilon$ . Due to the lack of such uniform  $\boldsymbol{\mu}$ -independent *a-posteriori* estimates one confines to a weaker statement of guaranteeing this estimate uniformly for a previously selected finite subset  $M \subset \mathcal{P}$ . The general procedure of incremental basis-construction in the offline-phase [14,15] can be divided into the following steps:

- (1) compute a small initial reduced basis  $\Phi_{N_0}$  of  $N_0$  basis-vectors;
- (2) set  $\Phi_N := \Phi_{N_0}$  and  $N := N_0$ ;
- (3) determine all offline-simulation quantities for the current RB-basis  $\Phi_N$ ;
- (4) for all  $\boldsymbol{\mu} \in M$  determine the posterior error estimates  $\Delta_N^k(\boldsymbol{\mu})$ ,  $0 \leq k \leq K$ ;
- (5) select a  $\boldsymbol{\mu}^*, k^*$  with  $\Delta_N^{k^*}(\boldsymbol{\mu}^*) = \max_{\boldsymbol{\mu} \in M, 0 \leq k \leq K} \Delta_N^k(\boldsymbol{\mu})$ ;
- (6) perform the detailed simulation for parameter  $\boldsymbol{\mu}^*$  resulting in  $u_H(\boldsymbol{\mu}^*)$ ;
- (7) determine a suitable new basis-function  $\varphi_{N+1}$  from this detailed simulation;
- (8) generate the extended reduced basis  $\Phi_{N+1} := \Phi_N \cup \{\varphi_{N+1}\}$ ;
- (9) set  $N := N + 1$  and repeat steps 3–9 until  $\max_{\boldsymbol{\mu} \in M} \Delta_N^k(\boldsymbol{\mu}) \leq \epsilon$ .

As the  $\Delta_N^k$  are upper bounds for the true error, the resulting reduced basis in particular guarantees  $\max_{\boldsymbol{\mu} \in M} \|u_H^k - u_N^k\|_{L^2(\Omega)} \leq \epsilon$ . Note that in case of  $C_E \geq 1$  the error estimates  $\Delta_N^k(\boldsymbol{\mu})$  are monotonically increasing with  $k$ , and the maxima will be obtained at the final time-step  $K$ . This insight can be used for basis-construction as the maximum search can be simplified by choosing  $k^* = K$ .

Two components in this algorithm must be specified: step 1, the initial choice of basis, and step 7, the rule for choice of the basis-extension vector.

### Choice of initial basis

The insights on the *a-posteriori* error estimate already give a guideline for the choice of a good initial reduced basis: it should comprise all different possible initial data constellations  $P[u_0(\boldsymbol{\mu})]$  for varying parameters in order to have the error estimate as small as possible. Due to the affine parameter dependence of  $u_0$  in (5.2) and linearity of  $P$  this is obtained by including the functions  $P[u_0^q]$  for  $q = 1, \dots, Q_{u_0}$  into the RB-space. Therefore, we choose  $\Phi_{N_0}$  as an orthonormalization of these initial data components in step 1 of the basis-generation process.

### Choice of basis-extension method

For the extension of the reduced basis in step 7 of the basis-construction, two algorithms are used. The first one is chosen as a reference method, which is similar to the one applied in [15].

#### Basis-extension by maximum $\Delta$ -increase snapshot:

Given that  $\{\Delta_N^k(\boldsymbol{\mu}^*)\}_{k=0}^K$  is a monotonically increasing sequence, a large estimator increase from  $\Delta_N^{k-1}$  to  $\Delta_N^k$  is taken as an indicator that the corresponding detailed solution  $u_H^k$  is not sufficiently accurate. Thus, the next vector  $\varphi_{N+1}$  is chosen by selecting the time index  $k^* := \arg \max_k (\Delta_N^k(\boldsymbol{\mu}^*) - \Delta_N^{k-1}(\boldsymbol{\mu}^*))$  and orthonormalizing the snapshot  $u_H^{k^*}(\boldsymbol{\mu}^*)$  with respect to the given basis  $\Phi_N$ .

This algorithm has certain disadvantages as already partially mentioned in [14]: only a single snapshot from the whole time-evolution is used in the new basis, which is a large loss of information and expensive computation time. Further, the assumption of estimator reduction by including the snapshot turns out to be wrong. It can happen that the selected snapshot already is in the current RB-space, the estimator increase is simply due to that  $u_N^{k-1}$  is already an inaccurate approximation of  $u_H^{k-1}$ . To prevent an occasional infinite loop of repeatedly selecting the same snapshot for basis-extension an additional test and a *backup procedure* has to be included in the above extension method [14].

Therefore, we propose a new selection method by using the whole detailed time evolution  $\{u_H(\boldsymbol{\mu}^*)\}_{k=0}^K$ , instead of only selecting one snapshot. Further, we want to avoid getting stuck in an infinite loop of falsely chosen new basis-functions, which are already in the current RB-space. Lastly, we want to perform information compression in order to have an as compact basis as possible. These points can be realized by introducing procedures known from the model reduction technique of *proper orthogonal decomposition*, see for instance [24]. In particular we apply a version of the Karhunen-Loève transformation or Principal Component Analysis (PCA) for data compression, which respects initially defined fixspaces. Instead of solving the PCA *via* an expensive or infeasible  $H \times H$  eigenvalue problem, it can be performed based on the  $K \times K$  *empirical Gramian*. For computational details we refer to control theory literature [36], which denotes it the *method of snapshots* or to machine-learning literature, which denotes it as kernel-PCA, *cf.* [38].

#### Basis-extension by PCA with fixspace:

The trajectory  $\{u_H^k\}_{k=0}^K$  is projected on  $\mathcal{W}_N^\perp$  and the projected sequence is subject to a PCA. The principal component, *i.e.* the vector of maximum variance is chosen as  $\varphi_{N+1}$ , which is readily orthonormal to  $\Phi_N$  due to construction.

The latter choice of basis-extension vector ensures that the maximum trajectory information is used, as  $\varphi_{N+1}$  represents the optimal new variance direction. Thus, a very compact basis is generated as demonstrated by the experiments. The empirical Gramian and the eigenvalue problem to be solved in this procedure have size  $K \times K$ . The procedure therefore is only practical until a number  $K_{max}$  of a few thousand time-instances. If  $K > K_{max}$  the basic idea of the basis-extension method can still be realized. As the changes between subsequent snapshots in time are mostly small, an equidistant selection of  $K_{max}$  snapshots from the complete trajectory or more general subsampling still seems to be capable of capturing the main trajectory variation and still leads to a tractable  $K_{max} \times K_{max}$  eigenvalue problem.

## 7. EXPERIMENTS

The experiments have been realized in a MATLAB-environment ranging from detailed FV-simulations to full offline/online decomposition leading to an interactive online-demo, where parameter variation by the user is possible with instant RB-simulation result and error estimation.

### Model problem

The geometry of the model problem is illustrated in Figure 1 and represents a simplified section of a gas-diffusion layer (gdl) of a fuel cell  $\Omega = [0, 1 \times 10^{-3}] \times [0, 2 \times 10^{-4}]$  with impermeable boundaries  $\Gamma_3, \Gamma_6$  at the middle of the top and the bottom, a gas inlet  $\Gamma_2$  at the left top, a gas-outlet  $\Gamma_4$  at the right top connecting

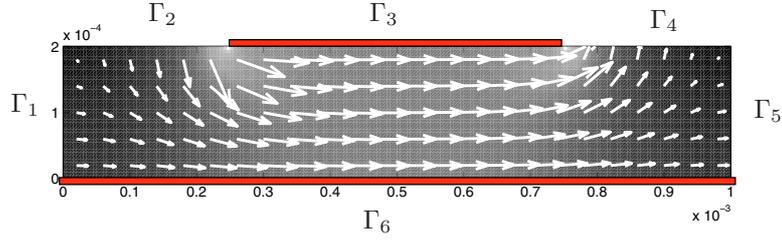


FIGURE 1. Illustration of the geometry and velocity field.

the gdl to gas channels, and the left and right boundaries  $\Gamma_1$  and  $\Gamma_5$  leading to identical gdl-sections. The space discretization is a cartesian grid of  $40 \times 200$  cells, the time range is  $t \in [0, T_{\max} = 0.5]$  discretized with  $K = 200$  equally sized time-intervals.

A time-independent velocity field is obtained from an elliptic finite element simulation for the pressure on an enlarged domain with a pressure decrease between the inlet and outlet. The negative of the resulting pressure gradient is taken as velocity field  $\mathbf{v}(\mathbf{x})$  indicated with arrows in Figure 1. The diffusivity is taken as  $d(\mathbf{x}, t) \equiv \delta$  with parameter  $\delta$ . For realizing some variety in the resulting dynamics for different parameters, we consider initial data  $u_0(\mathbf{x}) = \frac{1}{2}c_{\text{init}}(\sin(10\,000\pi x) + 1)$  with a parameter  $c_{\text{init}}$  interpolating between homogeneous zero initial data and the full sine-wave. The Dirichlet boundary values are set as  $b_{\text{dir}}(\mathbf{x}, t) = \beta\chi_{\Gamma_2} + (1-\beta)\chi_{\Gamma_4}$ , where  $\chi_{\Gamma_i}$  denote the indicator functions of the corresponding boundary segments. Thus,  $b_{\text{dir}}$  is parametrized by  $\beta$ , which models concentration differences between the inlet and outlet. The left boundary is an inflow boundary set to Dirichlet-value zero, the right boundary is an outflow boundary condition, the middle top and lower boundary are assigned noflow conditions, *i.e.* the Neuman boundary values are chosen as  $b_{\text{neu}} = \chi_{\Gamma_5}(\mathbf{v}u) \cdot \mathbf{n}$ .

By this we have specified our P<sup>2</sup>DE (2.1)–(2.4) with parameter vector  $\boldsymbol{\mu} = (c_{\text{init}}, \beta, \delta)^T$  being variable in the range  $\mathcal{P} := [0, 1] \times [0, 1] \times [0, 5 \times 10^{-8}]$ . The solution variation with different parameter choices are visualized in Figure 2. At initial time  $t = 0$ , the variation of the initial data is shown for  $c_{\text{init}} = 0$  and  $c_{\text{init}} = 1$  in the upper row, while the other parameters are fixed as  $\beta = 1, \delta = 5 \times 10^{-8}$ . At some intermediate time  $t = 0.1$ , the different effects of diffusivity choice are visualized in the middle row for  $\delta = 0$  and  $\delta = 5 \times 10^{-8}$  with fixed  $c_{\text{init}} = 1, \beta = 0$ . The lower row illustrates the effect of the different inlet/outlet concentrations by  $\beta = 0$  and  $\beta = 1$  at the end time  $t = 0.5$  with fixed  $c_{\text{init}} = 1, \delta = 0$ .

This choice of data functions allows an affine parameter dependence in terms of Definition 5.1 by treating  $\mathbf{v}(\mathbf{x}, t)$ ,  $d(\mathbf{x}, t)$ ,  $u_0(\mathbf{x})$ ,  $b_{\text{neu},1}(\mathbf{x}, t)$  and  $b_{\text{neu},0}(\mathbf{x}, t)$  each as one component ( $Q_{\mathbf{v}} = Q_{\mathbf{d}} = Q_{u_0} = Q_{b_{\text{neu},1}} = Q_{b_{\text{neu},0}} = 1$ ,  $\mathbf{v}^1(\mathbf{x}, t) = \mathbf{v}(\mathbf{x})$ ,  $\sigma_{\mathbf{v}}^1(\boldsymbol{\mu}) = 1$ ,  $d^1(\mathbf{x}, t) = 1$ ,  $\sigma_d^1(\boldsymbol{\mu}) = \delta$ ,  $u_0^1(\mathbf{x}) = \frac{1}{2}(\sin(10\,000\pi x) + 1)$ ,  $\sigma_{u_0}^1(\boldsymbol{\mu}) = c_{\text{init}}$ ,  $b_{\text{neu},1}^1(\mathbf{x}, t) = \chi_{\Gamma_5} \mathbf{v} \cdot \mathbf{n}$ ,  $\sigma_{b_{\text{neu},1}}^1(\boldsymbol{\mu}) = 1$ ,  $b_{\text{neu},0}^1(\mathbf{x}, t) = 0$ ,  $\sigma_{b_{\text{neu},0}}^1(\boldsymbol{\mu}) = 0$ ) and  $u_{\text{dir}}(\mathbf{x}, t)$  as two components ( $Q_{b_{\text{dir}}} = 2$ ,  $b_{\text{dir}}^1(\mathbf{x}, t) = \chi_{\Gamma_2}$ ,  $\sigma_{b_{\text{dir}}}^1(\boldsymbol{\mu}) = \beta$ ,  $b_{\text{dir}}^2(\mathbf{x}, t) = \chi_{\Gamma_4}$ ,  $\sigma_{b_{\text{dir}}}^2(\boldsymbol{\mu}) = (1-\beta)$ ).

As seen in Section 5 the affine parameter dependence of the data functions transfers to the flux, to the operator matrices and therefore with Lemma 5.1 to the complete RB-scheme and error estimators. Concerning the *a-posteriori* error estimators, we have psd-ness of both our implicit and explicit space-discretization operators  $\bar{L}_I^k$  and  $\bar{L}_E^k$  due to Lemma 2.1 (iii)-(iv), so we can use  $C_I = C_E = 1$  as upper bounds for the continuity constants of  $(L_I^k(\boldsymbol{\mu}))^{-1}$  and  $L_E^k(\boldsymbol{\mu})$ . Further, in view of the linear dependency of  $\bar{L}_I^k(\boldsymbol{\mu})$  on the diffusivity, we can exactly compute the parameter dependent coercivity bound  $\alpha(\boldsymbol{\mu}) = \alpha(\delta) = \delta \cdot \alpha(\delta = 1)$ .

### Detailed *versus* reduced simulation

We perform a complexity comparison of the detailed FV-simulation *versus* the RB-simulation with 20 basis-functions, which is already a very good approximation if the basis-functions are chosen suitably, for instance as described in Section 6. We do not focus on accuracy of the RB-simulation in this section, the emphasis is put on memory and speed demands.

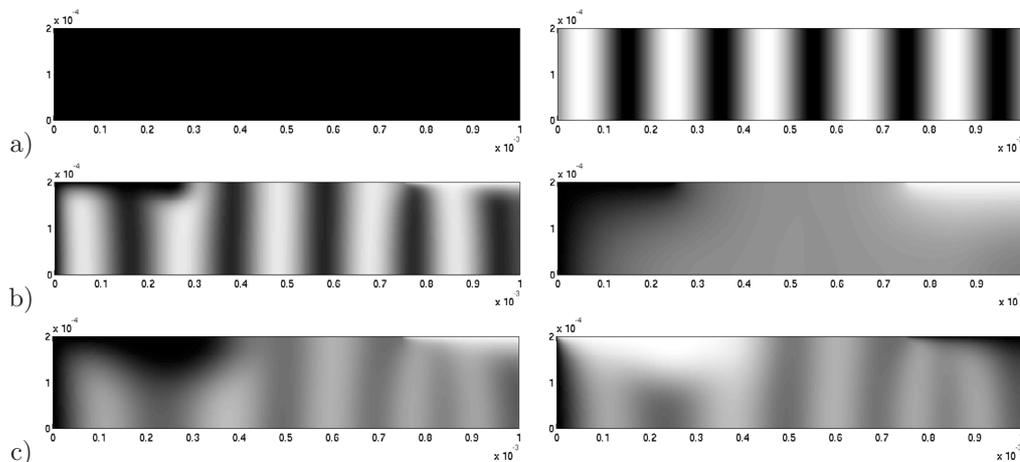


FIGURE 2. Illustration of solution variety by parameter variation. a) Initial data variation at start time; b) diffusion variation at intermediate time; c) inlet variation at end time.

Concerning the runtime complexity, the main determining factors are the number of time-steps  $K$ , space-dimensionality  $H$  and number of reduced basis vectors  $N$ . Table 1 presents the results of the experimental assessments on a standard PC (Pentium 4, 2.8 GHz, 2 GB RAM) for two different space/time discretizations. The left part illustrates the general situation where the data functions (velocities, diffusivities, boundary conditions) are time-dependent, *i.e.* the offline-data construction and online-assembly are performed for all time-steps. The right part demonstrates the case where the data is constant in time, so the operators, offline-data and online-assembly only have to be performed once and can be reused for all time-steps. We report the simulation times for the detailed simulation, the reduced basis online and offline contributions. Here the offline-computations consist of preparing the matrices and auxiliary quantities. The reduced basis is assumed to be generated in advance. Results for the basis construction are given in a subsequent section. We consider the general case of advection-diffusion ( $\delta > 0$ ), where the semi-implicit discretization requires the solution of a linear system in each time-step. In the second case of pure advection ( $\delta = 0$ ), the implicit contributions vanish and the simulations simplify to matrix-vector multiplications. The time-step restrictions in both cases are given by the advection term which results in identical  $K$  in the advection-diffusion and pure advection case.

We first focus on the upper half of the table. In the setting of time-dependent data, the online RB-simulation is about 6–9 times faster than the detailed FV-simulation. About 1/3rd of the detailed computation time is spent in solving the linear system in the advection-diffusion case. This is not a relevant contribution in the RB case, as the matrices are smaller and the dominating time is required for assembling of the online auxiliary quantities, which has almost identical complexity in the explicit or the semi-implicit case.

In the right part of the upper table, the case of data functions which are constant in time, all times are largely reduced. The RB-offline-phase is almost ideally sped up by factor  $K = 200$ . The complexity difference between the detailed and the RB-online-phase is more expressed here than in the time-dependent case, the speedup-factors range from 2–45.

In the lower part of the table we repeat the experiment with a refined discretization in space (factor 2 per dimension) and time (factor 5) in order to give an impression of the scaling behaviour. The high number of time-steps is required due to higher velocities near the better resolved singularities at the interface of boundary segments  $\Gamma_2/\Gamma_3$  and  $\Gamma_3/\Gamma_4$ . The main qualitative observations are valid as before. However, the relative complexity gains are much better expressed, as the detailed simulations largely depend on the space resolution in contrast to the reduced online simulations. The speedup factors in the non-constant data setting range from 21–28 and in the constant data case from 5–150.

TABLE 1. Time complexity comparisons between detailed and RB-simulation in the cases of advection-diffusion and pure advection and the data functions being constant in time or not.

<b>Discretization: <math>40 \times 200</math> cells, <math>K = 200</math> time-steps</b>						
	Data non const. in time			Data const. in time		
	Detailed	RB online	RB offline	Detailed	RB online	RB offline
advection-diffusion	155.94 s	16.67 s	447.16 s	45.67 s	1.02 s	2.41 s
advection	105.97 s	16.53 s	437.20 s	1.51 s	0.79 s	2.31 s

<b>Discretization: <math>80 \times 400</math> cells, <math>K = 1000</math> time-steps</b>						
	Data non const. in time			Data const. in time		
	Detailed	RB online	RB offline	Detailed	RB online	RB offline
advection-diffusion	4043.18 s	143.57 s	8693.90 s	924.91 s	6.18 s	9.22 s
advection	2758.20 s	134.00 s	8506.60 s	17.41 s	3.64 s	8.83 s

TABLE 2. Memory requirement [KB] with increasing number of basis-functions and time-dependent or time-independent data.

<b>Discretization: <math>40 \times 200</math> cells, <math>K = 200</math> time-steps</b>						
	Number of basis-functions					
	20	40	60	80	100	120
Reduced Basis	1250	2500	3750	5000	6250	7500
Offline data const. in time	230	898	2002	3545	5525	7942
Offline data non const. in time	46 014	179 452	400 390	708 827	1 104 765	1 588 203

Overall, high speedup can be obtained, *e.g.* in case of implicit contributions, where the dominating time of the detailed simulation is the solution of the large linear systems. On the other hand, the RB-offline phase is very demanding, in particular scaling linearly both with  $H$  and  $K$ , which is clearly reflected in the factor 20 in the left part of the table. Remind that this is no problem for the scenario of repeated RB-simulations with varying parameters, as only the online phase is relevant here.

Concerning memory requirements, the main factor again is the number of time-steps and the number of basis-functions as the required matrices scale quadratically with  $N$  and the vectors and reduced basis size grow linearly. For the implementation of the model problem with FV-discretization as given in the previous sections, the affine decompositions of the detailed operators result in  $Q_P = 1$ ,  $Q_{L_I} = 2$ ,  $Q_{L_E} = 4$ ,  $Q_b = 7$  components. Correspondingly, the number of components of the RB-operators range from 2 to 7, the number of components for the error estimator range from 4 to 49. As the matrices in the offline and online data are non-sparse in general, we obtain the memory requirements for the offline data for growing basis size as given in Table 2. The linear scaling in memory storage for the reduced basis and the quadratic scaling in the offline data is clearly visible.

In general, for low number of basis-functions, the memory requirements may be dominated by the detailed reduced basis vectors, but with increasing number of basis-functions, at some point the offline data requirements are the demanding and limiting factor. In the case of time-dependent data functions, this is even more expressed. The requirements mainly scale with the number of time-steps. With 200 time-steps in our case, 120 reduced basis vectors already imply 1.6 GB of offline data. This is not only a capacity problem, but also limiting the reduced basis simulation. This offline data must be accessed in total in each online simulation run during operator assembly. A solution for this problem can be based on the idea of keeping the reduced basis small [14]. As a small basis can only capture limited variability in time, only a limited time-interval can be modelled well with a small basis. Therefore, multiple reduced bases can be constructed, each suited to a certain time-interval. Note that still certain types of time-variant data can be treated with the memory complexities as listed in the tables

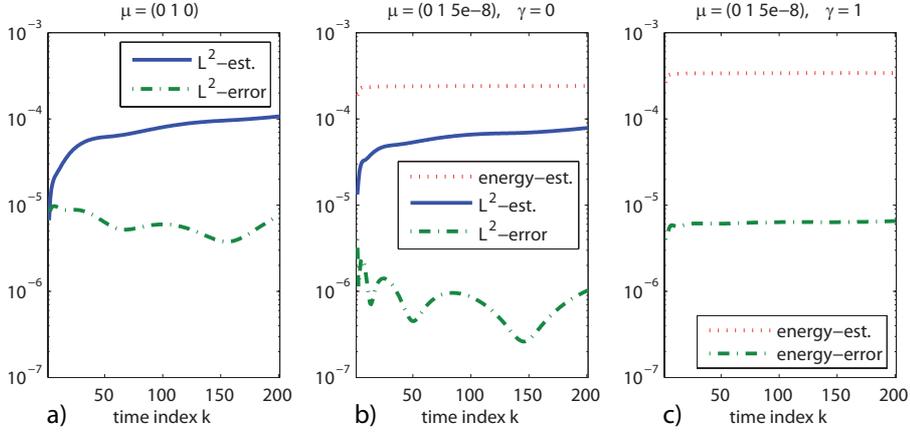


FIGURE 3. Time-behaviour of error estimators  $\Delta_N^k$  and  $\bar{\Delta}_{N,\gamma}^k$  and true error  $\|e^k\|$ ,  $\|e^k\|_\gamma$ . a)  $L^2$ -error and estimator in case of no diffusivity; b) energy- and  $L^2$ -estimator and  $L^2$ -error for large diffusivity; c) energy-estimator and -error for large diffusivity.

as *const. in time*. These cases are those, where the data components of the affine-decompositions  $d^q, \mathbf{v}^q$ , etc. are constant in time, but the coefficients  $\sigma_d^q, \sigma_v^q$ , etc. might be varying.

### Quality of error estimators

Reusing the RB-space spanned by the 20 basis-vectors as in the preceding section, we investigate the predictivity of the error estimators in Figures 3 and 4. ( $L^2$ -estimator  $\Delta_N^k$ : blue, solid; energy-estimator  $\bar{\Delta}_{N,\gamma}^k$ : red, dotted) in comparison with the true error ( $\|e^k\|$ ,  $\|e^k\|_\gamma$ : green, dashed-dotted). We choose a corner of the domain  $\mathcal{P}$  with diffusivity  $\delta = 0$  and logarithmically plot the true  $L^2$ -error  $\|e^k(\boldsymbol{\mu})\| = \|u_H^k(\boldsymbol{\mu}) - u_N^k(\boldsymbol{\mu})\|_{L^2(\Omega)}$  and the error estimator  $\Delta_N^k(\boldsymbol{\mu})$  over time indexed by  $k$  in the first plot of Figure 3. For diffusivity  $\delta > 0$  the energy-estimators are defined, so we additionally plot this quantity for  $\gamma = 0$  in the middle plot. The right plot illustrates the  $\gamma = 1$  energy-error estimator and the corresponding energy-error.

The results uniformly reflect the conservativity of the error estimators by  $\Delta_N^k(\boldsymbol{\mu}) \geq \|e^k(\boldsymbol{\mu})\|$  and  $\bar{\Delta}_{N,\gamma}^k \geq \|e^k(\boldsymbol{\mu})\|_\gamma$ . Due to our operator-bound  $C_E = 1$ , the estimators are monotonically increasing, whereas the true  $L^2$ -error can be non-monotonic, depending, on whether the current state is captured well or inaccurate in the RB-space. In the present range of diffusivity and time index, the  $L^2$ -estimators are preferable to the energy-estimates, as they are more tight. However, the slower asymptotic growing rate of the energy-estimators in time is also visible. Despite omitting the presentation of detailed results, one can indeed observe that the energy-estimator can be more tight than the corresponding  $L^2$ -error estimator. This can for instance be shown by extending the the time range of the middle plot by the factor 5. Similarly, increasing the diffusivity leads to constellations where the energy-error is preferable: by regarding the original time range and increasing the diffusivity  $\delta$  by a factor of 4, the energy-error estimator is better on a dominating part of the time interval.

We want to illustrate the predictivity of the error estimators under varying parameters. Figure 4 presents some results of the quantities evaluated at the end time  $t = T_{\max}$ . The first plot demonstrates the  $L^2$ -error estimator  $\Delta_N^K$  and the true error at  $\boldsymbol{\mu} = (c_{\text{init}} = 1, \beta, \delta = 0)$  under variation of  $\beta$ , the middle plot for  $\boldsymbol{\mu} = (1, 0, \delta)$ , additionally contains the energy-estimator  $\bar{\Delta}_{N,\gamma}^k$  for  $\gamma = 0$ . The last plot illustrates the  $\gamma = 1$  error estimator  $\bar{\Delta}_{N,\gamma}^K$  for  $\boldsymbol{\mu} = (c_{\text{init}}, 1, 5 \times 10^{-8})$  and corresponding energy-error.

In all plots the parameters  $\boldsymbol{\mu}$  with maximum error are correctly predicted by the estimators. This is an underpinning of incremental basis-construction algorithms, as indeed the parameters that are identified online are likely to be parameters with maximum true error. The three plots indicate the difference and applicability

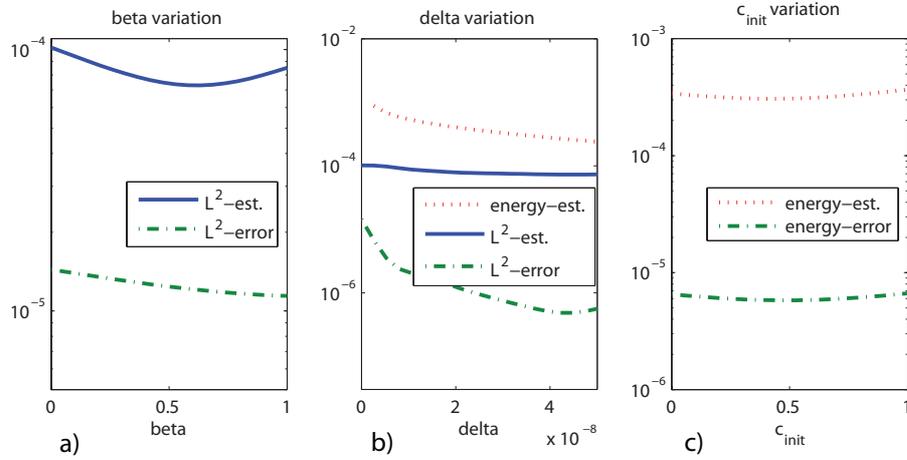


FIGURE 4. Predictivities of error estimators at  $t = T_{\max}$  for parameter variations. a)  $\Delta_N^k$  and  $\|e^k\|$  for  $c_{\text{init}} = 1, \delta = 0$  and variation of  $\beta$ ; b)  $\bar{\Delta}_{N,\gamma}^k, \Delta_N^k$  and  $\|e^k\|_\gamma$  for  $\gamma = 0, c_{\text{init}} = 1, \beta = 0$  and variation of  $\delta$ ; c)  $\bar{\Delta}_{N,\gamma}^k$  and  $\|e^k\|_\gamma$  for  $\gamma = 1, \beta = 1, \delta = 5 \times 10^{-8}$  and variation of  $c_{\text{init}}$ .

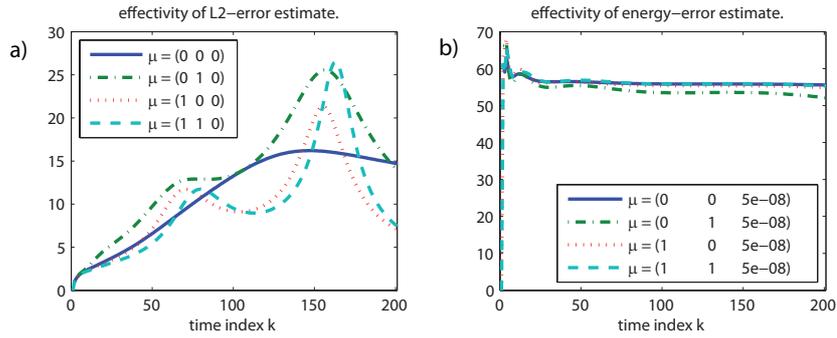


FIGURE 5. Effectivities of error estimators over time for different  $\mu$ . a) Effectivity  $\eta_N^k(\mu)$  of the  $L^2$ -estimator  $\Delta_N^k(\mu)$  for  $\delta = 0$ ; b) effectivity  $\bar{\eta}_{N,\gamma}^k(\mu)$  of the energy-estimator  $\bar{\Delta}_{N,\gamma}^k(\mu)$  with  $\gamma = 1$  and  $\delta = 5 \times 10^{-8}$ .

of the  $L^2$ - and energy-error estimators: the former is well defined in the whole parameter range covering hyperbolic and parabolic problems, the latter is only applicable in the parabolic case. As soon as the diffusivity goes to zero, the energy-estimators diverge and are not informative.

In addition to the conservativity and predictivity of the error estimators, we want to assess the effectivities  $\eta_N^k(\mu) := \Delta_N^k(\mu) / \|e^k(\mu)\|$  and  $\bar{\eta}_{N,\gamma}^k(\mu) := \bar{\Delta}_{N,\gamma}^k(\mu) / \|e^k(\mu)\|_\gamma$ . Figure 5 gives the results of these quantities over time. Plot a) illustrates the effectivities for the  $L^2$ -estimator in the hyperbolic case, *i.e.*  $\delta = 0$ , by plotting one curve for each of the four corresponding corners  $\mu$  of the parameter cube  $\mathcal{P}$ . Plot b) depicts the effectivities of the energy-error estimators in the parabolic case  $\delta = 5 \times 10^{-8}$  by similarly plotting one curve for each of the corresponding corners  $\mu$ .

It is obvious that the conservativity of the error estimator guarantees  $\eta_N^k(\mu), \bar{\eta}_{N,\gamma}^k(\mu) \geq 1$ . The factor of overestimation of the error is ranging from one to two orders of magnitude, which is a quite accurate prediction in addition to the large computational gain.

## Reduced basis construction

In this last experimental section we report on results concerning the reduced basis construction with the algorithms given in Section 6. The estimator-threshold was set to  $\epsilon = 1 \times 10^{-7}$  and a finite subset  $M \subset \mathcal{P}$  consisting of an equidistant cartesian grid was chosen. Figure 6 illustrates the main findings. In plot a) the decrease of the maximum error estimator  $\max_{\boldsymbol{\mu} \in M} \Delta_N^K(\boldsymbol{\mu})$  is plotted with increasing basis size  $N$ . The set  $M$  consists of  $5^3$  parameters. It can be seen in the logarithmic scale for the error estimator that both the PCA-fixspace and the max-error-snapshot extension methods are able to decrease the maximum error estimator exponentially in  $N$ .

Obviously, the PCA-based algorithm indeed is more efficient than the snapshot selection both in terms of final basis size and the computation time for assembling the basis: the PCA-fixspace algorithm obtains the desired threshold with 123 basis-functions in 6.1 h, the snapshot-selection algorithm ends with 136 basis-functions in 7.4 h. Concerning the offline-computation time, this is a 17.4% improvement in computation time and a 9.6% saving in basis size for given accuracy. Due to the squared memory requirements in the online-phase, this factor results in a 18.2% memory saving for the auxiliary matrices.

The set  $M$  can be understood as the training-set for the basis-generation. So more interesting than the error on the training-set, is a test-set accuracy by choosing a random set of new parameters  $M'$  with  $M \cap M' = \emptyset$  and similarly determining  $\max_{\boldsymbol{\mu} \in M'} \Delta_N^K(\boldsymbol{\mu})$ . This quantity for the PCA-fixspace algorithms is plotted in b) for a uniformly distributed random test-set  $M'$  of size 100 for different sets  $M$  used in basis-construction. Several curves are plotted corresponding to subsequent globally refined training-parameter sets  $M$  of size  $3^3$ ,  $5^3$  and  $9^3$ . It is obvious in the  $|M| = 5^3$  curve that the error estimates for test-parameters are up to two orders of magnitude larger than for training-parameters. The convergence in the test-error is not exponential as in the first plot, which indicates that the set  $M$  is chosen suboptimally. By refinement of the training-set  $M$  from size  $3^3$  over  $5^3$  to  $9^3$ , the test-error can be lowered by one order of magnitude, but still being far from the exponential decay of the training-error. So we conclude that methods of suitably and adaptively choosing the set  $M$  must be developed instead of choosing  $M$  *a-priori* fixed.

The incremental basis-construction with both extension methods can result in one  $\boldsymbol{\mu}$  to be chosen as basis-extension parameter  $\boldsymbol{\mu}^*$  multiple times. For preventing subsequent multiple identical detailed simulation runs, these trajectories  $\{u_H^k(\boldsymbol{\mu}^*)\}_{k=0}^K$  could be cached temporarily until the basis-construction is finished. For our current implementation, however, the runtimes are acceptable without this caching strategy. The frequency of selecting any  $\boldsymbol{\mu} \in M$  as  $\boldsymbol{\mu}^*$  in the given experiment for the  $|M| = 5^3$  cartesian grid is plotted in Figure 7. It can be seen that the automatic parameter selection is much more efficient than for instance selecting one basis-function for all parameters in  $M$ : only a small subset of the 125 parameter vectors in  $M$  is selected for basis-construction. Most of the vectors are required for small diffusion coefficients  $\delta$ , whereas for large diffusivities, only few basis-vectors are required. This is in accordance with the intuition: for large diffusivities  $\delta$ , the end-time solutions of various choices of  $c_{\text{init}}, \beta$  are similar as detailed effects are smeared out. For small diffusivities  $\delta$ , the end-state  $u_H^K(\boldsymbol{\mu})$  still has a high spacial variability with respect to the remaining parameters  $c_{\text{init}}, \beta$ , so more basis-vectors are required here.

For parameters, which effect in a linear superposition of data, *e.g.*  $c_{\text{init}}$  for the initial data or  $\beta$  for the inlet-parameter, the selected parameters  $\boldsymbol{\mu}^*$  tend to be at the borders of the intervals. For other parameters, such as the diffusivity  $\delta$ , the selected values can range over the whole allowed interval. This different quality of parameter dependence is reflected in the plot.

## 8. DISCUSSION AND CONCLUSION

We presented a formulation of the RB-methodology for arbitrary parametrized linear evolution schemes, which allow an explicit/implicit operator decomposition. We have demonstrated that our approach in particular covers FV-methods for parabolic and hyperbolic equations. Our operator formulation also covers finite element formulations. Thus, the RB-scheme formulation and error estimates can be seen an extension or alternative to existing approaches. Our operator formulation also covers more general schemes such as higher order schemes

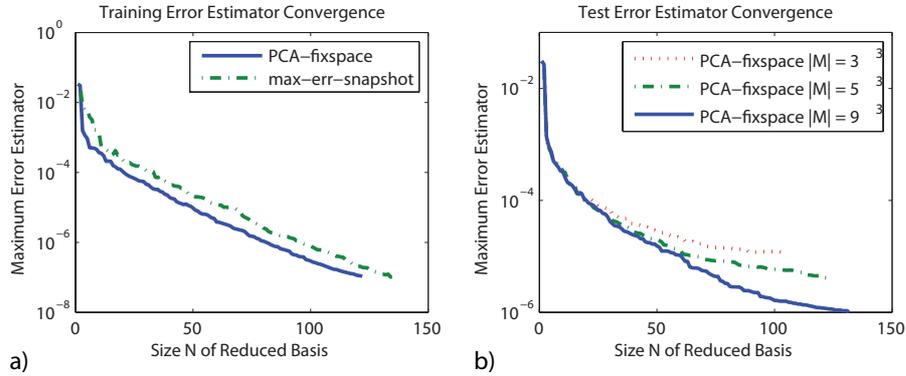


FIGURE 6. Illustration of basis-generation. a) Comparison of different algorithms by maximum training error estimator for  $\mu$  chosen from the set  $M$ ; b) maximum test-error estimator on a random test-set  $M'$  for different training-set sizes.

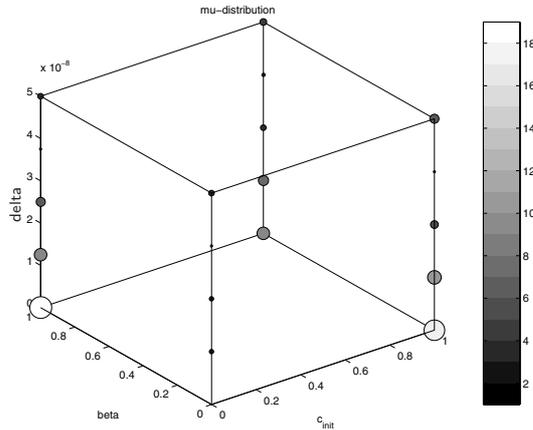


FIGURE 7. Frequency of  $\mu^*$ -selection during reduced basis generation on a training-set  $M \subset \mathcal{P}$  consisting of an equidistant cartesian grid of  $5^3$  points  $\mu$ .

which include Runge-Kutta time discretizations, or local discontinuous Galerkin discretizations in space [7]. As long as the problem and the fluxes are linear and no limiters are involved, the methodology is directly transferable.

As an extension, estimates for output functionals can be derived instead of only considering the field variable. If, for instance, a linear functional output  $s(u_H^k)$  is desired, this can be approximated by  $s(u_N^k)$ . The given  $L^2$ -error-bound  $\Delta_N^k$  directly implies a bound on the output-error by  $|s(u_N^k) - s(u_H^k)| = |s(e^k)| \leq \|s\| \Delta_N^k$ . More sophisticated error bounds might be derived with primal-dual techniques as they exist for the elliptic or parabolic case [15,33].

As a second attractive perspective, the extension of the method to nonlinear FV or local discontinuous Galerkin (LDG) schemes seems possible by affine approximations and collateral reduced basis spaces for the nonlinearity. This is currently under detailed investigation.

A further promising direction for future research is seen in the offline automatic basis-construction. We have illustrated that automatic basis-construction yields information about variability of the underlying physics: parameter distributions during basis-generation indicate parameter ranges with high dynamics in time or with

solution structure being distinct from other parameter ranges. The empirical convergence analysis demonstrated that there is a big gap between the optimal exponential error decrease on the training-set of parameter values  $M$  and the error decrease on arbitrary test-parameters. If uniform convergence statements cannot be guaranteed, at least algorithms should be found, which construct reduced bases with empirical uniform exponential convergence. The greedy search may provide such exponential error decrease up to a certain  $N$  for a large enough training set of parameters. But in presence of *a-posteriori* error estimators which do not have negligible computation time, as in our time dependent case, such large training sets may be infeasible. This motivates basis-construction methods that start with a small training set  $M$  and apply careful extension of  $M$  based on adaptive exploration of the parameter space. This may make use of an available offline/online decomposition and error estimators of the reduced scheme, but may as well be valuable in cases without such decomposition, or cases without error-estimators, where detailed simulations are required for error-quantification.

*Acknowledgements.* The first author was funded by the German Federal Ministry of Education and Research under grant number 03SF0310C. Additionally, we acknowledge funding by the Landesstiftung Baden-Württemberg gGmbH. We thank Gianluigi Rozza for fruitful discussions on the subject and the anonymous referees for the constructive comments.

## REFERENCES

- [1] B.O. Almroth, P. Stern and F.A. Brogan, Automatic choice of global shape functions in structural analysis. *AIAA J.* **16** (1978) 525–528.
- [2] D.N. Arnold, F. Brezzi, B. Cockburn and L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* **39** (2002) 1749–1779.
- [3] C. Bardos, A.Y. Leroux and J.C. Nedelec, First order quasilinear equations with boundary conditions. *Comm. Partial Diff. Eq.* **4** (1979) 1017–1034.
- [4] M. Barrault, Y. Maday, N.C. Nguyen and A.T. Patera, An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris Ser. I Math.* **339** (2004) 667–672.
- [5] T. Barth and M. Ohlberger, Finite volume methods: Foundation and analysis, in *Encyclopedia of Computational Mechanics*, E. Stein, R. de Borst and T.J.R. Hughes Eds., John Wiley & Sons (2004).
- [6] J. Carrillo, Entropy solutions for nonlinear degenerate problems. *Arch. Ration. Mech. Anal.* **147** (1999) 269–361.
- [7] B. Cockburn, Discontinuous Galerkin methods for computational fluid dynamics, in *Encyclopedia of Computational Mechanics*, E. Stein, R. de Borst and T.J.R. Hughes Eds., John Wiley & Sons (2004).
- [8] B. Cockburn and C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.* **16** (2001) 173–261.
- [9] Y. Coudiere, J.P. Vila and P. Villedieu, Convergence rate of a finite volume scheme for a two dimensional convection-diffusion problem. *ESAIM: M2AN* **33** (1999) 493–516.
- [10] R. Eymard, T. Gallouët and R. Herbin, Finite volume methods, in *Handbook of numerical analysis*, volume **VII**, North-Holland, Amsterdam (2000) 713–1020.
- [11] R. Eymard, T. Gallouët, R. Herbin and A. Michel, Convergence of a finite volume scheme for nonlinear degenerate parabolic equations. *Numer. Math.* **92** (2002) 41–82.
- [12] R. Eymard, T. Gallouët and R. Herbin, A cell-centred finite volume approximation for anisotropic diffusion operators on unstructured meshes in any space dimension. *IMA J. Numer. Anal.* **26** (2006) 326–353.
- [13] E. Godlewski and P.-A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer (1996).
- [14] M.A. Grepl, *Reduced-basis Approximations and a Posteriori Error Estimation for Parabolic Partial Differential Equations*. Ph.D. thesis, Massachusetts Institute of Technology, USA (2005).
- [15] M.A. Grepl and A.T. Patera, *A posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: M2AN* **39** (2005) 157–181.
- [16] P. Grisvard, *Singularities in boundary value problems, Recherches en Mathématiques Appliquées* **22** [Research in Applied Mathematics]. Masson, Paris (1992).
- [17] R. Herbin and M. Ohlberger, A posteriori error estimate for finite volume approximations of convection diffusion problems, in *Proc. 3rd Int. Symp. on Finite Volumes for Complex Applications - Problems and Perspectives* (2002) 753–760.
- [18] R.L. Higdon, Initial-boundary value problems for linear hyperbolic systems. *SIAM Rev.* **28** (1986) 177–217.
- [19] M.-J. Jator and L. Lévi, Singular perturbations for a class of degenerate parabolic equations with mixed Dirichlet-Neumann boundary conditions. *Ann. Math. Blaise Pascal* **10** (2003) 269–296.
- [20] D. Kröner, *Numerical Schemes for Conservation Laws*. John Wiley & Sons and Teubner (1997).
- [21] R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press (2002).

- [22] L. Machiels, Y. Maday, I.B. Oliveira, A. Patera and D.V. Rovas, Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *C. R. Acad. Sci. Paris Ser. I Math.* **331** (2000) 153–158.
- [23] M. Mangold and M. Sheng, Nonlinear model reduction of a 2D MCFC model with internal reforming. *Fuel Cells* **4** (2004) 68–77.
- [24] B.C. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control* **AC-26** (1981) 17–32.
- [25] N.C. Nguyen, K. Veroy and A.T. Patera, Certified real-time solution of parametrized partial differential equations, in *Handbook of Materials Modeling*, S. Yip Ed., Springer (2005) 1523–1558.
- [26] A.K. Noor and J.M. Peters, Reduced basis technique for nonlinear analysis of structures. *AIAA J.* **18** (1980) 455–462.
- [27] M. Ohlberger, *A posteriori* error estimates for vertex centered finite volume approximations of convection-diffusion-reaction equations. *ESAIM: M2AN* **35** (2001) 355–387.
- [28] M. Ohlberger, *A posteriori* error estimate for finite volume approximations to singularly perturbed nonlinear convection-diffusion equations. *Numer. Math.* **87** (2001) 737–761.
- [29] M. Ohlberger and J. Vovelle, Error estimate for the approximation of non-linear conservation laws on bounded domains by the finite volume method. *Math. Comp.* **75** (2006) 113–150.
- [30] A.T. Patera and G. Rozza, Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations. Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) *MIT Pappalardo Graduate Monographs in Mechanical Engineering*.
- [31] T.A. Porsching and M.L. Lee, The reduced basis method for initial value problems. *SIAM J. Numer. Anal.* **24** (1987) 1277–1287.
- [32] C. Prud'homme, D. Rovas, K. Veroy and A.T. Patera, A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *ESAIM: M2AN* **36** (2002) 747–771.
- [33] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera and G. Turinici, Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluids Engineering* **124** (2002) 70–80.
- [34] A. Quarteroni, G. Rozza, L. Dede and A. Quaini, Numerical approximation of a control problem for advection-diffusion processes, in *System Modeling and Optimization, Proceedings of 22nd IFIP TC7 Conference* (2006).
- [35] D.V. Rovas, L. Machiels and Y. Maday, Reduced basis output bound methods for parabolic problems. *IMA J. Numer. Anal.* **26** (2006) 423–445.
- [36] C.W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. Bifurcat. Chaos* **15** (2005) 997–1013.
- [37] G. Rozza, *Shape design by optimal flow control and reduced basis techniques: Applications to bypass configurations in haemodynamics*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Switzerland (2005).
- [38] B. Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press (2002).
- [39] T. Tonn and K. Urban, *A reduced-basis method for solving parameter-dependent convection-diffusion problems around rigid bodies*. Technical Report 2006-03, Institute for Numerical Mathematics, Ulm University, ECCOMAS CFD (2006).
- [40] K. Veroy and A.T. Patera, Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds. *Int. J. Numer. Meth. Fluids* **47** (2005) 773–788.
- [41] K. Veroy, C. Prud'homme and A.T. Patera, Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds. *C. R. Acad. Sci. Paris Ser. I Math.* **337** (2003) 619–624.