Markus A. Dihlmann · Bernard Haasdonk

Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems

Stuttgart, February 2013

Institute for Applied Analysis and Numerical Simulation,
Universität Stuttgart,
Pfaffenwaldring 57, D-70569 Stuttgart, Germany
Tel.: +49-711-685-67646
Fax: +49-711-685-65507
E-mail: Markus.Dihlmann@mathematik.uni-stuttgart.de
E-mail: haasdonk@mathematik.uni-stuttgart.de

Abstract We consider parameter optimization problems which are subject to constraints given by parametrized partial differential equations (P^2DE). Discretizing this problem may lead to a large-scale optimization problem which can hardly be solved rapidly. In order to accelerate the process of parameter optimization we will use a reduced basis surrogate model for numerical optimization. For many optimization methods sensitivity information about the functional is needed. In the following we will show that this derivative information can be calculated efficiently in the reduced basis framework in the case of a general linear output functional and parametrized evolution problems with linear parameter separable operators. By calculating the sensitivity information directly instead of applying the more widely used adjoint approach we can rapidly optimize arbitrary cost functionals using the same reduced basis model. Furthermore, we will derive rigorous a-posteriori error estimators for the solution, the gradient and the optimal parameters, which can all be computed online. The method will be applied to two parameter optimization problems with an underlying advection-diffusion equation.

Keywords PDE-constrained optimization · reduced order modelling · reduced basis method · surrogate model · parameter optimization · a-posteriori error estimation

Stuttgart Research Centre for Simulation Technology (SRC SimTech)
SimTech – Cluster of Excellence
Pfaffenwaldring 5a
70569 Stuttgart
publications@simtech.uni-stuttgart.de
www.simtech.uni-stuttgart.de

1 Introduction

We consider the parameter optimization problem

$$(\mathbb{P}) = \begin{cases} \min_{\mu \in \mathcal{P}} J(u(\boldsymbol{\mu})) \\ \text{s.t.} \\ \mathbb{E}(\boldsymbol{\mu}) \text{ is solved.} \end{cases} \tag{1}$$

where the linear functional J evaluates the solution $u(\boldsymbol{\mu})$ to a parametrized partial differential equation (P^{2}DE) denoted by $\mathbb{E}(\boldsymbol{\mu})$. The parameter $\boldsymbol{\mu}$ stems from a parameter domain $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$. We want to focus on the case of $\mathbb{E}(\boldsymbol{\mu})$ being a general parameterized linear evolution equation of the form

$$(\mathbb{E}(\boldsymbol{\mu})) = \begin{cases} \partial_t u(\cdot, t; \boldsymbol{\mu}) = \mathcal{L}(t; \boldsymbol{\mu}) u(\cdot, t; \boldsymbol{\mu}) + b(\cdot, t; \boldsymbol{\mu}) & \text{in } \Omega \\ u(\cdot, 0; \boldsymbol{\mu}) = u_0(\cdot; \boldsymbol{\mu}). \end{cases} \tag{2}$$

Here $\mathcal{L}(t; \boldsymbol{\mu})$ is a linear spatial differential operator and $b(\cdot, t; \boldsymbol{\mu})$ denotes source or boundary values. Details on operator and functional properties will follow in Section 2.

This kind of parameter optimization problem with an evolution equation as PDE-constraint can be found in numerous engineering applications, for example optimizing the efficiency of a heat exchanger or the simulation based optimization of energy efficient buildings [5]. If the underlying model is complex, a discretization of the governing equation can lead to a large-scale discrete optimization problem [3] causing extensively long computation times for solving the optimization problem. In other cases the problem may not be solvable under real-time requirements.

In order to accelerate the optimization, one possible approach is to replace the discretized PDE model by a reduced order surrogate model obtained by a model reduction technique like POD [1] or reduced basis (RB) methods [14, 19, 30]. In the optimization context POD as well as RB methods have already been introduced, namely to solve optimal control problems [9, 12, 31, 32] and parameter optimization problems [25, 28, 29]. In our work we will use a reduced basis surrogate model to approximate the behaviour of the P^{2}DE, but in contrast to [25, 28, 29] where stationary problems were treated, we will handle evolution equations. We will also provide a more detailed error estimation, including as a main result the certification of the optimal parameters found by the reduced optimization.

The work we present here is not limited to one specific numerical optimization scheme and in principle any technique can be applied (gradient descent, SQP, trust region, etc). For many techniques derivative information of the functional, e.g. the gradient or the Hessian, is needed. Essentially, there are two ways to obtain this information efficiently: the adjoint approach and the sensitivity approach. The first one consists in calculating the functional gradient by a Lagrange formalism which includes the solution of an adjoint problem [6, 8, 15, 22, 23]. When using the adjoint approach in combination with the reduced basis method, the reduced space has to be designed so that the adjoint solutions can be approximated well in this space. Hence both, the state solution and the adjoint solution have to be taken into account during the basis construction [9, 12, 21]. As these adjoint solutions depend on the choice of the functional, one has to build up a new reduced basis for every new functional. Consequently this approach is suited for cases where one specific optimization problem for a given functional has to be solved. In our work we use the second possibility to calculate the functional gradient where the parameter derivative information is obtained by solving a sensitivity PDE [4, 26]. Using the sensitivity approach gives us more flexibility in the use of the reduced model. In contrast to the adjoint approach the reduced basis surrogate model once constructed can be used for all choices of functionals. Furthermore we easily obtain higher order derivative information of the functional like Hessians for example.

Although solving the optimization problem with a reduced basis surrogate model is in general much faster, the found solution can be suboptimal due to the approximative nature of the surrogate model. In order to control this suboptimality we will not only provide rigorous error bounds for the functional and the functional gradient as in [28] but we will also derive rigorous error bounds for the derivative solutions and the optimal parameters. Thereby we can provide a certification of the optimal parameters obtained by the optimization process with the reduced basis surrogate model.

The work is structured as follows: In Section 2 we explain in detail the problem setting and introduce some notation. The subsequent Section 3 discusses briefly the calculation of sensitivity information using the sensitivity PDE. In Section 4 we show how the reduced basis surrogate model can be set

up and how it is used to solve the parameter optimization problem. We will conduct an error analysis in Section 5. In Section 6 we apply the method to an advection-diffusion problem and conclude in Section 7.

2 Problem setting

Discretized models are the starting point for the model reduction with RB-methods, so we directly assume a suitable finite but high dimensional Hilbert space of functions $X_h(\Omega)$ on the domain $\Omega \subseteq \mathbb{R}^d$ with a scalar product $\langle \cdot, \cdot \rangle$ and a norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$. Here we consider the discretized optimization problem

$$(\mathbb{P}_h) = \begin{cases} \min_{\mu \in \mathcal{P}} J(u_h(\boldsymbol{\mu})) \\ \text{s.t.} \\ \mathbb{E}_h(\boldsymbol{\mu}) \text{ is solved} \end{cases} \tag{3}$$

where the linear functional J evaluates solutions $u_h(\boldsymbol{\mu})$ to the discrete evolution scheme

$$\mathbb{E}_h(\boldsymbol{\mu}) = \begin{cases} \mathcal{L}_I(t^{k+1}; \boldsymbol{\mu}) u_h^{k+1}(\boldsymbol{\mu}) = \mathcal{L}_E(t^k; \boldsymbol{\mu}) u_h^k(\boldsymbol{\mu}) + b_h^k(\boldsymbol{\mu}) \\ u_h^0(\boldsymbol{\mu}) = P\left(u_0(\boldsymbol{\mu})\right) \end{cases} \tag{4}$$

approximating $\mathbb{E}(\boldsymbol{\mu})$ in X_h. It is obtained by a discretisation of (2) in space (by FEM, FV or LDG for example), a discretisation of the time interval $t \in [0, T]$ into $K+1$ time instants $t^k = k\Delta t$, $k = 0, \ldots, K$ with $\Delta t = \frac{T}{K}$ and integration in time using a general explicit/implicit scheme (c.f.[19]). The discrete initial conditions are obtained by a suitable projection $P : X \to X_h$. Hence, $(\mathbb{E}_h(\boldsymbol{\mu}))$ represents a linear system of equations with solutions $u_h^k(\boldsymbol{\mu}) = u_h(t^k; \boldsymbol{\mu}) \in X_h(\Omega)$. We denote by $u_h(\boldsymbol{\mu}) := \{u_h^0(\boldsymbol{\mu}), u_h^1(\boldsymbol{\mu}), \ldots, u_h^K(\boldsymbol{\mu})\} \in (X_h(\Omega))^{K+1}$ the sequence of solutions at the time steps $t^k \in \mathbb{T}_K := \{t^0, t^1, \ldots, t^K\}$. In order to allow operator splitting in our formulation, the operator $\mathcal{L}$ is discretized into an explicit part $\mathcal{L}_E$ and an implicit part $\mathcal{L}_I$. This allows for example, to choose between an implicit or explicit time integration. Both are assumed to be bounded linear operators $\mathcal{L}_E(t^k; \boldsymbol{\mu}), \mathcal{L}_I(t^k; \boldsymbol{\mu}) : X_h \to X_h$. Additionally, $\mathcal{L}_I(t^{k+1}; \boldsymbol{\mu})$ is assumed to be invertible for any $\boldsymbol{\mu} \in \mathcal{P}$ and the inverse being uniformly bounded $\left\| \mathcal{L}_I(t^{k+1}; \boldsymbol{\mu})^{-1} \right\| \leq 1$. These assumptions imply uniform well-posedness of (4) over the parameter domain. Note, that we do neither require nor assume coercivity of the implicit discretization part, as we also want to cover pure explicit discretizations. However, under an additional coercivity assumption on the implicit part sharper error bounds could be derived [14].

The discrete operators, the inhomogeneous part as well as the initial conditions are assumed to be parameter separable so that they can be represented as linear combinations, e.g.

$$\mathcal{L}_E(t^k; \boldsymbol{\mu}) u_h^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{\mathcal{L}_E}} \Theta_{\mathcal{L}_E}^q(t^k; \boldsymbol{\mu}) \mathcal{L}_E^q u_h^k(\boldsymbol{\mu}) \tag{5}$$

of parameter dependent scalar coefficients $\Theta_{\mathcal{L}_E}^q(t^k; \boldsymbol{\mu}) : \mathbb{T}_K \times \mathcal{P} \to \mathbb{R}$ and parameter independent components $\mathcal{L}_E^q : X_h \to X_h$ (and similar for $\mathcal{L}_I, b_h, u_0$). We assume that all coefficients $\Theta_{[\cdot]}^q$ are differentiable with respect to $\boldsymbol{\mu}$. Parameter separability is an important property for obtaining the offline/online decomposition leading to an efficient calculation of solutions for varying parameters. In case that the operators or the inhomogeneous part are not parameter separable one can apply the empirical interpolation method [2, 20] to generate separable approximations. The above scheme covers finite element or finite volume discretizations of parabolic or hyperbolic problems with first or second order time discretization, e.g. [13, 19].

In order to simplify the notation and to keep the statements flexible for extension to other than linear functionals, e.g. quadratic functionals or adding a parameter dependence to the functional itself, we introduce the definition of a condensed functional (c.f. also [22, Section 1.6] for example)

$$J_h(\boldsymbol{\mu}) := J(u_h(\boldsymbol{\mu})) \text{ with } u_h(\boldsymbol{\mu}) \text{ solution to } (\mathbb{E}_h(\boldsymbol{\mu})). \tag{6}$$

This definition is valid, as we know that for any $\boldsymbol{\mu} \in \mathcal{P}$ a unique solution to $(\mathbb{E}_h(\boldsymbol{\mu}))$ exists and can be evaluated by the functional J. In other words the condensed functional $J_h(\boldsymbol{\mu})$ represents the solving

of $(\mathbb{E}_h(\boldsymbol{\mu}))$ for the given parameter $\boldsymbol{\mu}$ and a subsequent evaluation of the obtained solution $u_h(\boldsymbol{\mu})$ by the functional $J(u_h(\boldsymbol{\mu}))$.

In order to find a solution to the optimization problem, any numerical optimization scheme can be used [11, 24].

3 Evolution schemes for sensitivity derivatives

Some numerical optimization methods use derivative information about the output functional $J_h(\boldsymbol{\mu})$ to solve the optimization problem (e.g. gradient-descent, SQP, etc.). Hence, we seek to calculate the functional gradient

$$\nabla_\mu J_h(\boldsymbol{\mu}) = (\partial_{\mu_1} J_h(\boldsymbol{\mu}), \dots, \partial_{\mu_p} J_h(\boldsymbol{\mu}))^T,$$

where for a $\boldsymbol{w} \in \mathbb{R}^p$ the transposed of $\boldsymbol{w}$ is marked as $\boldsymbol{w}^T$ and where ∂_{μ_i} denotes the derivative in direction of the i-th parameter. A widely used approach to obtain the gradient of the functional is to formulate the KKT-System and to solve the adjoint problem [6, 8, 15, 22, 23, 29]. But in order to provide more flexibility in the use of the reduced basis surrogate model (especially in the choice of the functional) we chose the alternative way of solving a sensitvity PDE [4, 26], which can be solved rapidly in the RB-framework. This approach is also easier to handle and to implement compared to the adjoint approach. Additionally, higher order derivative information (e.g. Hessian matrices) can be derived more easily, if required.

In our case we assume linearity of the functional J in u_h, hence we can make use of $\partial_{\mu_i} J(u_h(\boldsymbol{\mu})) = J(\partial_{\mu_i} u_h(\boldsymbol{\mu}))$ for $i = 1, \dots, p$. This allows to obtain the derivative of the linear output functional J_h by calculating the functional value of the derivative of the solution $\partial_{\mu_i} u_h$. Note that in this case J is not directly dependent on $\boldsymbol{\mu}$ but only via the solution $u_h(\boldsymbol{\mu})$. Direct parameter dependence of $J(u_h(\boldsymbol{\mu}))$ and nonlinear functionals are subject to future work. Note, that despite linearity with respect to u_h, the dependency on the parameter $\boldsymbol{\mu}$ may be highly nonlinear.

Similar to stationary cases [28] the sensitivity derivatives can be obtained using a sensitivity PDE, which in our case is an evolution problem.

Remark 1 If the solution $u(\boldsymbol{\mu})$ of $(\mathbb{E}(\boldsymbol{\mu}))$ is differentiable at $\boldsymbol{\mu}$ with respect to the parameter μ_i, and there exists a solution $v_i(\cdot, t; \boldsymbol{\mu})$ to the following auxiliary evolution problem

$$\partial_t v_i(\cdot, t; \boldsymbol{\mu}) = \mathcal{L}(t; \boldsymbol{\mu})\, v_i(\cdot, t; \boldsymbol{\mu}) + b^*(\cdot, u, t; \boldsymbol{\mu}) \tag{7}$$

$$v_i(\cdot, 0, \boldsymbol{\mu}) = \partial_{\mu_i} u_0(\cdot, \boldsymbol{\mu}) \tag{8}$$

with

$$b^*(\cdot, u, t; \boldsymbol{\mu}) = (\partial_{\mu_i} \mathcal{L}(t; \boldsymbol{\mu}))\, u(\cdot, t; \boldsymbol{\mu}) + \partial_{\mu_i} b(\cdot, t; \boldsymbol{\mu})) \tag{9}$$

then

$$\partial_{\mu_i} u = v_i. \tag{10}$$

The proof of the statement can be obtained by differentiating the evolution equation (2). We remark, that the differentiability assumption can be obtained in case of parameter separable operator and source term, if the coefficient functions are differentiable, and the well-posedness of the evolution problem transfers to modified initial/boundary functions as in (7), (8). Note also, that the parameter derivatives of the operator $(\partial_{\mu_i} \mathcal{L})$ and the right hand side $(\partial_{\mu_i} b)$ inherit the parameter separability from $\mathcal{L}$ and b as

$$(\partial_{\mu_i} \mathcal{L})(t; \boldsymbol{\mu}) u(\boldsymbol{\mu}) = \sum_{q=1}^{Q_\mathcal{L}} (\partial_{\mu_i} \Theta^q_\mathcal{L}(t, \boldsymbol{\mu})) \mathcal{L}^q u(\boldsymbol{\mu}) \tag{11}$$

$$(\partial_{\mu_i} b)(\cdot, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} (\partial_{\mu_i} \Theta^q_b(t, \boldsymbol{\mu})) b^q. \tag{12}$$

The same holds for the initial conditions.

We nicely see in Remark 1, that in order to obtain the derivative of u we have to solve the same problem as in (2) using the same operators but with a changed inhomogeneous part b^* depending on the solution u.

After discretization of (7) we obtain the evolution scheme to calculate the derivative $\partial_{\mu_i} u_h = v_{h,i}$ as

$$(\mathbb{E}_{\partial_{\mu_i},h}(\boldsymbol{\mu})) = \begin{cases} \mathcal{L}_I\left(t^{k+1};\boldsymbol{\mu}\right) v_{h,i}^{k+1}(\boldsymbol{\mu}) = \mathcal{L}_E\left(t^k;\boldsymbol{\mu}\right) v_{h,i}^{k}(\boldsymbol{\mu}) + b_h^{*k}(\boldsymbol{\mu}) \\ \text{with} \\ \quad b_h^{*k}(\boldsymbol{\mu}) = \sum\limits_{q=1}^{Q_{\mathcal{L}_E}} (\partial_{\mu_i}\Theta_{\mathcal{L}_E}^q(t^k;\boldsymbol{\mu}))\mathcal{L}_{h,E}^q u_h^k(\boldsymbol{\mu}) \\ \qquad\qquad + \sum\limits_{q=1}^{Q_{\mathcal{L}_I}} (\partial_{\mu_i}\Theta_{\mathcal{L}_I}^q(t^{k+1};\boldsymbol{\mu}))\mathcal{L}_{h,I}^q u_h^{k+1}(\boldsymbol{\mu}) \\ \qquad\qquad + \sum\limits_{q=1}^{Q_b} (\partial_{\mu_i}\Theta_b^q(t^k;\boldsymbol{\mu}))b_h^q \\ v_{h,i}^0(\cdot;\boldsymbol{\mu}) = P\left(\partial_{\mu_i} u_0(\cdot;\boldsymbol{\mu})\right). \end{cases} \tag{13}$$

Remark 2 Under the assumption that we use the same discretization methods for the discretization of (7) - (9) as for the discretization of (2) one can prove that the solution $v_h(\boldsymbol{\mu})$ obtained after discretization in time and space of (7) - (9) is the sensitivity derivative of the discrete solution $u_h(\boldsymbol{\mu})$ to the evolution scheme (4).

Remark 3 Note, that higher order derivative information like for example Hessian matrices can be derived by applying the same procedure once more to the sensitivity PDE (7) - (9).

4 Reduced basis method for parameter optimization

Applying a numerical optimization method on the problem $(\mathbb{P}_h)$ comes in general with multiple solves of the P^{2}DE (and maybe the sensitivity PDE) and subsequent evaluations of the functional for several parameters $\boldsymbol{\mu}$. When using a high dimensional space for numerical calculations this can take an excessive amount of time which rises with higher complexity of the underlying PDE and high number of parameters in the optimization problem. In order to avoid the time consuming calculations we approximate the solution $u_h(\boldsymbol{\mu})$ by a reduced solution $u_N(\boldsymbol{\mu}) \in (X_N)^{K+1}$ in a lower dimensional space $X_N \subseteq X_h$ with $\dim(X_N) = N \ll \dim(X_h)$ and replace the detailed optimization problem $(\mathbb{P}_h)$ by the reduced optimization problem

$$(\mathbb{P}_N) = \begin{cases} \min_{\mu\in\mathcal{P}} J(u_N(\boldsymbol{\mu})) \\ \text{s.t.} \\ \mathbb{E}_N(\boldsymbol{\mu}) \text{ is solved.} \end{cases} \tag{14}$$

Here $\mathbb{E}_N(\boldsymbol{\mu})$ represents the surrogate model obtained by a reduced basis model reduction of $(\mathbb{E}_h)$. Clearly, this is only a low dimensional approximation of the high dimensional problem and consequently the solution will not be exact. The quality of the result depends on the chosen solution space X_N. In the following we will derive the reduced basis model (Section 4.1) and present a scheme for calculating the sensitivity derivatives (Section 4.2). A feature of the reduced basis method is the separatation of the calculations into an offline and an online phase, guaranteeing rapid simulations for a given parameter. This will be explained in Section 4.3. In Section 4.4 we will outline how to actually generate a suitable reduced space X_N.

4.1 Reduced evolution scheme $(\mathbb{E}_N(\boldsymbol{\mu}))$

In a first step we assume that the reduced basis vectors $\varphi_n \in X_h$ for $n = 1, \ldots, N$ are given and span the reduced basis space

$$X_N = \text{span}\Phi_N = \text{span}\{\varphi_1, ..., \varphi_N\}.$$

Furthermore, we assume that the basis vectors φ_n are orthonormal, so that $\langle \varphi_n, \varphi_m \rangle = \delta_{nm} \ \ \forall n, m = 1, \dots, N$ with δ_{nm} being the Kronecker delta. For the reduced solution we make the ansatz

$$u_N^k(\boldsymbol{\mu}) = \sum_{n=1}^{N} a_n^k(\boldsymbol{\mu}) \varphi_n(x) \tag{15}$$

with $a_n^k : \mathcal{P} \to \mathbb{R}$. By performing a Galerkin projection with (4) and (15) we obtain for the reduced evolution step from k to $k+1$

$$\left\langle \mathcal{L}_I(t^{k+1}; \boldsymbol{\mu}) u_N^{k+1}(\boldsymbol{\mu}) - \mathcal{L}_E(t^k; \boldsymbol{\mu}) u_N^k(\boldsymbol{\mu}) - b_h(t^k; \boldsymbol{\mu}), \varphi_m \right\rangle = 0 \tag{16}$$

$\forall m = 1, \dots, N$. The projection results in the reduced basis surrogate model

$$(\mathbb{E}_N(\boldsymbol{\mu})) = \begin{cases} \boldsymbol{L}_I(t^{k+1}; \boldsymbol{\mu}) \boldsymbol{a}^{k+1} = \boldsymbol{L}_E(t^k; \boldsymbol{\mu}) \boldsymbol{a}^k + \boldsymbol{b}(t^k; \boldsymbol{\mu}) \\ \boldsymbol{a}^0 = \left(\langle u_h^0, \varphi_n \rangle \right)_{n=1}^N \end{cases} \tag{17}$$

with

$$\boldsymbol{a}^k = \left(a_1^k, a_2^k, \dots, a_N^k \right)^T \tag{18}$$

$$\left(\boldsymbol{L}_I(t^{k+1}; \boldsymbol{\mu}) \right)_{m,n} = \left\langle \mathcal{L}_I(t^{k+1}; \boldsymbol{\mu}) \varphi_n, \varphi_m \right\rangle \tag{19}$$

$$\left(\boldsymbol{L}_E(t^k; \boldsymbol{\mu}) \right)_{m,n} = \left\langle \mathcal{L}_E(t^k; \boldsymbol{\mu}) \varphi_n, \varphi_m \right\rangle \tag{20}$$

$$\left(\boldsymbol{b}(t^k; \boldsymbol{\mu}) \right)_n = \left\langle b_h(t^k; \boldsymbol{\mu}), \varphi_n \right\rangle \tag{21}$$

for $n, m = 1, \dots, N$. This is a slight reformulation of [19]. The reduced initial conditions were obtained by the orthogonal projection $P_N : X_h \to X_N$ of u_h^0. We see that $(\mathbb{E}_N(\boldsymbol{\mu}))$ is a low dimensional evolution scheme as $\boldsymbol{L}_I(t^{k+1}; \boldsymbol{\mu})$, $\boldsymbol{L}_E(t^k; \boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$ and $\boldsymbol{a}^k, \boldsymbol{b}(t^k; \boldsymbol{\mu}) \in \mathbb{R}^N$ are of low order and thus solutions to this evolution scheme can be clculated rapidly.

4.2 Reduced evolution scheme for the sensitivity $(\mathbb{E}_{\partial_{\mu_i}, N})(\boldsymbol{\mu})$

Similar to the detailed condensed functional we define a reduced condensed functional

$$J_N(\boldsymbol{\mu}) = J(u_N(\boldsymbol{\mu})) \text{ with } u_N(\boldsymbol{\mu}) \text{ solution to } (\mathbb{E}_N(\boldsymbol{\mu})). \tag{22}$$

Again, as the functional is linear we need reduced derivative solutions $\partial_{\mu_i} u_N(\boldsymbol{\mu})$ in order to calculate the reduced gradient $\nabla_\mu J_N(\boldsymbol{\mu})$.

We use different approximation spaces for the original solutions and for each directional derivative solution. Using the same approximation space for the solution and its derivatives turned out to be inefficient in terms of required basis size for a given approximation error (see experiments in Section 6.1). All derivative solutions $\partial_{\mu_i} u_h$ for $i = 1, \dots, p$ will be approximated in their own approximation spaces $X_{N_{\partial_{\mu_i}}} = \operatorname{span} \Psi_i = \operatorname{span}\{\psi_{1,i}, \dots, \psi_{N_{\partial_{\mu_i}}, i}\} \subseteq X_h$ which are spanned by the basis vectors $\psi_{n,i}$ with $n = 1, \dots, N_{\partial_{\mu_i}}$. Here again, the spaces $X_{N_{\partial_{\mu_i}}}$ are of low dimension $\dim(X_{N_{\partial_{\mu_i}}}) = N_{\partial_{\mu_i}} \ll \dim(X_h)$.

For the derivative solution we make the ansatz

$$\partial_{\mu_i} u_N^k(\boldsymbol{\mu}) = v_{N,i}^k(\boldsymbol{\mu}) = \sum_{n=1}^{N_{\partial_{\mu_i}}} c_{n,i}^k(\boldsymbol{\mu}) \psi_{n,i}(x) \tag{23}$$

with $c_{n,i}^k : \mathcal{P} \to \mathbb{R}$. In order to find the coefficients $c_{n,i}^k$ of (23) we again perform a Galerkin projection of (13) onto the space $X_{N_{\partial_{\mu_i}}}$ with the substitution of u_h by u_N in the derivative and obtain thereby the reduced basis surrogate model for the derivative solution

$$(\mathbb{E}_{\partial_{\mu_i}, N}(\boldsymbol{\mu})) = \begin{cases} \boldsymbol{L}_{I\Psi_i\Psi_i}(t^{k+1}; \boldsymbol{\mu}) \boldsymbol{c}_i^{k+1} = \boldsymbol{L}_{E\Psi_i\Psi_i}(t^k; \boldsymbol{\mu}) \boldsymbol{c}_i^k + \partial_{\mu_i} \boldsymbol{L}_{E\Phi\Psi_i}(t^k; \boldsymbol{\mu}) \boldsymbol{a}^k \\ \qquad\qquad + \partial_{\mu_i} \boldsymbol{L}_{I\Phi\Psi_i}(t^{k+1}; \boldsymbol{\mu}) \boldsymbol{a}^{k+1} + (\partial_{\mu_i} \boldsymbol{b})(t^k; \boldsymbol{\mu}) \\ \boldsymbol{c}_i^0 = \left(\langle \partial_{\mu_i} u_h^0, \psi_{n,i} \rangle \right)_{n=1}^N \end{cases} \tag{24}$$

with

$$\begin{aligned}
\boldsymbol{c}_i^k &= (c_{1,i}^k, c_{2,i}^k, \dots, c_{N_{\partial_{\mu_i}},i}^k)^T \\
\left(\partial_{\mu_i} \boldsymbol{L}_{E\Phi\Psi_i}(t^k;\boldsymbol{\mu})\right)_{n,m} &= \left\langle (\partial_{\mu_i}\mathcal{L}_E)(t^k;\boldsymbol{\mu})\varphi_m, \psi_{n,i}\right\rangle \\
\left(\partial_{\mu_i} \boldsymbol{L}_{I\Phi\Psi_i}(t^{k+1};\boldsymbol{\mu})\right)_{n,m} &= \left\langle (\partial_{\mu_i}\mathcal{L}_I)(t^{k+1};\boldsymbol{\mu})\varphi_m, \psi_{n,i}\right\rangle \\
\left(\partial_{\mu_i} \boldsymbol{b}(t^k;\boldsymbol{\mu})\right)_n &= \left\langle \partial_{\mu_i} b_h(t^k;\boldsymbol{\mu}), \psi_{n,i}\right\rangle \\
\left(\boldsymbol{L}_{I\Psi_i\Psi_i}(t^{k+1};\boldsymbol{\mu})\right)_{n,m} &= \left\langle \mathcal{L}_I(t^{k+1};\boldsymbol{\mu})\psi_{m,i}, \psi_{n,i}\right\rangle \\
\left(\boldsymbol{L}_{E\Psi_i\Psi_i}(t^k;\boldsymbol{\mu})\right)_{n,m} &= \left\langle \mathcal{L}_E(t^k;\boldsymbol{\mu})\psi_{m,i}, \psi_{n,i}\right\rangle .
\end{aligned}$$

4.3 Offline / online separation

In the reduced basis method the calculation process is divided into an offline phase, where the high dimensional (maybe time consuming) operations are conducted, and an online phase, where we have only low dimensional and fast calculations. In order to realize the offline/online separation we use the parameter separability (5) of the operators, the source term and the initial conditions. During the offline phase the reduced bases are constructed and the parameter independent components

$$\begin{aligned}
\left(\boldsymbol{L}_I^q\right)_{n,m} &= \langle \mathcal{L}_I^q\varphi_m, \varphi_n\rangle , & \left(\boldsymbol{L}_E^q\right)_{n,m} &= \langle \mathcal{L}_E^q\varphi_m, \varphi_n\rangle , \\
\left(\boldsymbol{L}_{E\Psi_i\Psi_i}^q\right)_{k,l} &= \langle \mathcal{L}_E^q\psi_{l,i}, \psi_{k,i}\rangle , & \left(\boldsymbol{L}_{I\Psi_i\Psi_i}^q\right)_{k,l} &= \langle \mathcal{L}_I^q\psi_{l,i}, \psi_{k,i}\rangle , \\
\left(\boldsymbol{L}_{I\Phi\Psi_i}^q\right)_{k,m} &= \langle \mathcal{L}_I^q\varphi_m, \psi_{k,i}\rangle , & \left(\boldsymbol{L}_{E\Phi\Psi_i}^q\right)_{k,m} &= \langle \mathcal{L}_E^q\varphi_m, \psi_{k,i}\rangle , \\
\left(\boldsymbol{b}^q\right)_n &= \langle b_h^q, \varphi_n\rangle & \left(\boldsymbol{u}_0^q\right)_n &= \left\langle u_h^{0,q}, \varphi_n\right\rangle
\end{aligned} \tag{25}$$

$\forall n, m = 1, \dots, N$ and $\forall k, l = 1, \dots, N_{\partial_{\mu_i}}$ are calculated. All these matrices and vectors are of small dimension, i.e. N and $N_{\partial_{\mu_i}}$. After having calculated these components we can switch to the online phase. Now, for a given parameter value $\boldsymbol{\mu}$ the required operators can be assembled rapidly. First, we evaluate the required coefficients $\Theta_{\mathcal{L}_E}^q(t^k;\boldsymbol{\mu})$, $\Theta_{\mathcal{L}_I}^q(t^{k+1};\boldsymbol{\mu})$, $\Theta_b^q(t^k;\boldsymbol{\mu})$, $\partial_{\mu_i}\Theta_{\mathcal{L}_E}^q(t^k;\boldsymbol{\mu})$, $\partial_{\mu_i}\Theta_{\mathcal{L}_I}^q(t^{k+1};\boldsymbol{\mu})$, $\partial_{\mu_i}\Theta_b^q(t^k;\boldsymbol{\mu})$, $\partial_{\mu_i}\Theta_{u_0}^q(t^k;\boldsymbol{\mu})$ and then we assemble the reduced matrices by a linear combination of the coefficients and the appropriate components, e.g.:

$$\begin{aligned}
\boldsymbol{L}_I(t^k;\boldsymbol{\mu}) &= \sum_{q=1}^{Q_{\mathcal{L}_I}} \Theta_{\mathcal{L}_I}^q(t^k;\boldsymbol{\mu})\boldsymbol{L}_I^q, \\
(\partial_{\mu_i}\boldsymbol{L}_{I\Phi\Psi_i})(t^k;\boldsymbol{\mu}) &= \sum_{q=1}^{Q_{\mathcal{L}_I}} (\partial_{\mu_i}\Theta_{\mathcal{L}_I}^q(t^k;\boldsymbol{\mu}))\boldsymbol{L}_{I\Phi\Psi_i}^q,
\end{aligned} \tag{26}$$

and similar for $(\partial_{\mu_i}\boldsymbol{L}_{E\Phi\Psi_i})(t^k;\boldsymbol{\mu})$, $\boldsymbol{L}_{I\Psi_i\Psi_i}(t^k;\boldsymbol{\mu})$, $\partial_{\mu_i}\boldsymbol{u}_0(\boldsymbol{\mu})$, $\boldsymbol{L}_E(t^k;\boldsymbol{\mu})$, $\boldsymbol{b}(t^k;\boldsymbol{\mu})$, $(\partial_{\mu_i}\boldsymbol{b})(t^k;\boldsymbol{\mu})$, $\boldsymbol{L}_{E\Psi_i\Psi_i}(t^k;\boldsymbol{\mu})$.

4.4 Basis generation

There exist numerous approaches to generate a reduced approximation space. The POD-Greedy algorithm [18] turned out to be an efficient method in the case of evolution problems. In every iteration j of the POD-Greedy algorithm we search over a training set of parameters $\mathcal{M}_{train}$ for the parameter producing the maximum error estimator $\boldsymbol{\mu}^j = \max_{\boldsymbol{\mu}\in\mathcal{M}_{train}} \Delta_u(\boldsymbol{\mu})$ being introduced in the next section. Then a detailed solution $u_h(\boldsymbol{\mu}^j)$ is calculated and orthogonalized to the existing reduced basis Φ^j resulting in $u_{h,\perp}(\boldsymbol{\mu}^j)$. The information contained in this projection error trajectory is compressed by performing a POD and taking the first POD-mode $\varphi_j = \mathrm{POD}(u_{h,\perp}(\boldsymbol{\mu}^j), 1)$ as new basis vector. This new basis vector is used to extend the reduced basis $\Phi^{j+1} = \Phi^j \cup \{\varphi_j\}$. For details concerning the basis generation we refer to [16, 18, 33].

As we work with separate reduced basis spaces for the solution and the derivative solutions we conduct a POD-Greedy algorithm for constructing the reduced spaces for the solution and the derivative solutions individually. In the algorithm we use then the error estimator for the solution from Section 5.1 and the error estimator for the derivative solutions from Section 5.2 respectively. Furthermore, in order to improve the efficiency of the reduced basis we realized a combination of the P-partitioning and the T-partitioning approach [10, 17].

5 Error analysis

The reduced basis surrogate model provides, of course, only approximate solutions. In order to provide a "certification" of the approximations we will give rigorous error bounds quantifiying the quality of the approximate solutions $u_N(\boldsymbol{\mu})$, the derivative solutions $\partial_{\mu_i} u_N(\boldsymbol{\mu})$, the reduced gradient $\nabla_\mu J_N(\boldsymbol{\mu})$ and the optimal parameter solution $\boldsymbol{\mu}_N^*$ for the reduced optimization problem $(\mathbb{P}_N)$. For calculating these error bounds only low dimensional operations are needed, so that they can be calculated together with the actual simulations in the online phase. In the following we assume a uniform boundedness of the explicit operator by $\left\|\mathcal{L}_E(t^k;\boldsymbol{\mu})\right\| \leq C_E(\boldsymbol{\mu}) \leq \bar{C}_E$ with known bound constants $C_E(\boldsymbol{\mu})$ and $\bar{C}_E$.

5.1 Error estimators for the reduced solution u_N

We can approximate the high dimensional detailed solution $u_h^k(\boldsymbol{\mu}) \in X_h$ to (4) by the reduced solution $u_N^k(\boldsymbol{\mu}) \in X_N$ with an error $e^k(\boldsymbol{\mu}) \in X_h$ so that

$$u_h^k(\boldsymbol{\mu}) = u_N^k(\boldsymbol{\mu}) + e^k(\boldsymbol{\mu}). \tag{27}$$

As calculating this true error would be a high dimensional (and thereby costly) operation, we want to avoid calculating it in the online phase. Consequently we seek to derive an a-posteriori error estimator $\Delta_u^k(\boldsymbol{\mu})$ bounding the error $\|e^k(\boldsymbol{\mu})\| \leq \Delta_u^k(\boldsymbol{\mu})$ at every timestep $k = 0, \ldots, K$, which is calculable rapidly online.

The following Proposition is a slight reformulation of [19]:

Proposition 1 *Let $e^k = u_h^k - u_N^k$ be the approximation error at time step k, while assuming that $\|e^0\| = 0$. Then the error can be bounded by*

$$\|e^k\| \leq \Delta_u^k(\boldsymbol{\mu}) \tag{28}$$

with

$$\Delta_u^k(\boldsymbol{\mu}) = \sum_{j=1}^{k} C_E(\boldsymbol{\mu})^{k-j} \|\mathrm{Res}^j\|. \tag{29}$$

Here $C_E(\boldsymbol{\mu}) > 0$ is an upper bound for the continuity constant of the explicit operator $\mathcal{L}_E$. The residual Res^j is defined as

$$\mathrm{Res}^{k+1}(\boldsymbol{\mu}) := \mathcal{L}_I(t^{k+1};\boldsymbol{\mu}) u_N^{k+1} - \mathcal{L}_E(t^k;\boldsymbol{\mu}) u_N^k - b_h(t^k;\boldsymbol{\mu}). \tag{30}$$

Proof We start with the "exact" evolution scheme in (4). Putting the definition of the error (27) and the residual (30) in (4) results in

$$\begin{aligned} &\mathcal{L}_I(t^{k+1};\boldsymbol{\mu})(u_N^{k+1} + e^{k+1}) = \mathcal{L}_E(t^k;\boldsymbol{\mu})(u_N^k + e^k) + b_h^k \\ \Leftrightarrow\ &\mathcal{L}_I(t^{k+1};\boldsymbol{\mu}) e^{k+1} = \mathcal{L}_E(t^k;\boldsymbol{\mu}) e^k - \mathrm{Res}^{k+1}. \end{aligned}$$

As the implicit operator $\mathcal{L}_I$ satisfies by assumption $\|\mathcal{L}_I(t^{k+1};\boldsymbol{\mu})^{-1}\| \leq 1$,

$$\begin{aligned} e^{k+1} &= \mathcal{L}_I(t^k;\boldsymbol{\mu})^{-1} \left(\mathcal{L}_E(t^k;\boldsymbol{\mu})(e^k) - \mathrm{Res}^{k+1} \right) \\ \Rightarrow \|e^{k+1}\| &\leq \|\mathcal{L}_I(t^k;\boldsymbol{\mu})^{-1}\| \left(\|\mathcal{L}_E(t^k;\boldsymbol{\mu}) e^k\| + \|\mathrm{Res}^{k+1}\| \right) \\ \Rightarrow \|e^{k+1}\| &\leq \underbrace{\max_k (\|\mathcal{L}_E(t^k;\boldsymbol{\mu})\|)}_{\leq C_E(\boldsymbol{\mu})} \|e^k\| + \|\mathrm{Res}^{k+1}\|. \end{aligned}$$

By solving this equation recursively and assuming $\|e^0\| = 0$ we obtain

$$\|e^k\| \leq \sum_{j=1}^{k} C_E(\boldsymbol{\mu})^{k-j} \|\mathrm{Res}^j\| = \Delta_u^k(\boldsymbol{\mu}).$$

□

The norm of the residual needed for the error estimator can be calculated rapidly online by

$$\begin{aligned} \|\mathrm{Res}^{k+1}\|^2 =& {\boldsymbol{a}^{k+1}}^T \boldsymbol{M}_{II}(t^k;\boldsymbol{\mu})\boldsymbol{a}^{k+1} + {\boldsymbol{a}^k}^T \boldsymbol{M}_{EE}(t^k;\boldsymbol{\mu})\boldsymbol{a}^k \\ &+ \boldsymbol{M}_{bb}(t^k;\boldsymbol{\mu}) + 2{\boldsymbol{a}^k}^T \boldsymbol{M}_{Eb}(t^k;\boldsymbol{\mu}) \\ &- 2{\boldsymbol{a}^{k+1}}^T \boldsymbol{M}_{IE}(t^k;\boldsymbol{\mu})\boldsymbol{a}^k - 2{\boldsymbol{a}^{k+1}}^T \boldsymbol{M}_{Ib}(t^k;\boldsymbol{\mu}) \end{aligned} \tag{31}$$

using the low dimensional matrices

$$\begin{aligned} \left(\boldsymbol{M}_{EE}(t^k;\boldsymbol{\mu})\right)_{n,m} &= \left\langle \mathcal{L}_E(t^k;\boldsymbol{\mu})\varphi_n, \mathcal{L}_E(t^k;\boldsymbol{\mu})\varphi_m \right\rangle, \\ \left(\boldsymbol{M}_{IE}(t^k;\boldsymbol{\mu})\right)_{n,m} &= \left\langle \mathcal{L}_I(t^{k+1};\boldsymbol{\mu})\varphi_n, \mathcal{L}_E(t^k;\boldsymbol{\mu})\varphi_m \right\rangle, \\ \left(\boldsymbol{M}_{II}(t^k;\boldsymbol{\mu})\right)_{n,m} &= \left\langle \mathcal{L}_I(t^{k+1};\boldsymbol{\mu})\varphi_n, \mathcal{L}_I(t^{k+1};\boldsymbol{\mu})\varphi_m \right\rangle, \\ \left(\boldsymbol{M}_{Eb}(t^k;\boldsymbol{\mu})\right)_{n} &= \left\langle \mathcal{L}_E(t^k;\boldsymbol{\mu})\varphi_n, b^k \right\rangle, \\ \left(\boldsymbol{M}_{Ib}(t^k;\boldsymbol{\mu})\right)_{n} &= \left\langle \mathcal{L}_I(t^{k+1};\boldsymbol{\mu})\varphi_n, b(t^k;\boldsymbol{\mu}) \right\rangle, \\ \boldsymbol{M}_{bb}(t^k;\boldsymbol{\mu}) &= \left\langle b(t^k;\boldsymbol{\mu}), b(t^k;\boldsymbol{\mu}) \right\rangle, \end{aligned} \tag{32}$$

$\forall n, m = 1, \ldots, N$. This can be verified by squaring the formula for the residual from Proposition 1 and applying the ansatz (15). Note, that (30) does not grow arbitrarly with finer time-discretization, but the residual norms roughly scale with Δt. The condition $e^0 = 0$ can be obtained by including the exact initial conditions u_0^q in the reduced space. Otherwise a small initial error contribution will appear in (29).

5.2 Error estimator for the reduced derivative solution $\partial_{\mu_i} u_N$

We will now derive a rigorous error bound for the reduced derivative solution. It will turn out useful later in the derivation of an error bound for the reduced functional gradient $\nabla_\mu J_N(\boldsymbol{\mu})$.

Proposition 2 *Let $v_{h,i}^k$ with $k = 0, \ldots, K$ be the solution to (13), $v_{N,i}^k$ its reduced basis approximation and $e_{\partial_{\mu_i}}^k = v_{h,i}^k - v_{N,i}^k$ the approximation error. We assume that $\|e_{\partial_{\mu_i}}^0\| = 0$. Then the derivative approximation error can be bounded by*

$$\|e_{\partial_{\mu_i}}^k\| \leq \Delta_{\partial_{\mu_i} u}^k(\boldsymbol{\mu}) \tag{33}$$

with

$$\Delta_{\partial_{\mu_i} u}^k(\boldsymbol{\mu}) = \sum_{j=1}^{k} C_E(\boldsymbol{\mu})^{k-j} \left(C_{\partial_{\mu_i} L_E} \Delta_u^{j-1} + C_{\partial_{\mu_i} L_I} \Delta_u^j + \|\mathrm{Res}_{\partial_{\mu_i} u}^j\| \right). \tag{34}$$

Δ_u^k and $C_E(\boldsymbol{\mu})$ are the error estimator for the solution and the constant from Proposition 1. $C_{\partial_{\mu_i} L_E}$ and $C_{\partial_{\mu_i} L_I}$ are upper bounds for the induced operator norms of $\partial_{\mu_i}\mathcal{L}_E$ and $\partial_{\mu_i}\mathcal{L}_I$ respectively. The derivative residual is defined as

$$\mathrm{Res}_{\partial_{\mu_i} u}^{k+1} = \mathcal{L}_I(t^{k+1};\boldsymbol{\mu}) v_N^{k+1} - \mathcal{L}_E(t^k;\boldsymbol{\mu}) v_{N,i}^k - b_h^*(u_N^k, u_N^{k+1}, (\partial_{\mu_i} b)). \tag{35}$$

Proof We start with the "exact" evolution scheme for the derivative solution from (13). When applying the definition of the error $v_{h,i}^k = v_{N,i}^k + e_{\partial_{\mu_i}}^k$ to (13) we obtain

$$\begin{aligned} &\mathcal{L}_I^{k+1}(v_N^{k+1} + e_{\partial_{\mu_i} u}^{k+1}) \\ &\quad = \mathcal{L}_E^{k+1}(v_{N,i}^k + e_{\partial_{\mu_i} u}^k) + (\partial_{\mu_i}\mathcal{L})_E^k(u_h^k) + (\partial_{\mu_i}\mathcal{L})_I^{k+1}(u_h^{k+1}) + (\partial_{\mu_i} b)_h^k. \end{aligned} \tag{36}$$

As the implicit operator $\mathcal{L}_I$ satisfies $\|\mathcal{L}_I(t^{k+1};\boldsymbol{\mu})^{-1}\| \leq 1$ and with linearity of the operators we can rewrite (36) using the definition of the residual as

$$\begin{aligned} e^{k+1}_{\partial_{\mu_i}} = \left(\mathcal{L}^{k+1}_I\right)^{-1} \Big[&\mathcal{L}^k_E(e^k_{\partial_{\mu_i}}) + (\partial_{\mu_i}\mathcal{L})^k_E(u^k_h - u^k_N) \\ &+(\partial_{\mu_i}\mathcal{L})^{k+1}_I(u^{k+1}_h - u^{k+1}_N) - \mathrm{Res}^{k+1}_{\partial_{\mu_i} u}\Big] \end{aligned}$$

$$\begin{aligned} \Rightarrow \|e^{k+1}_{\partial_{\mu_i}}\| \leq \| \left(\mathcal{L}^{k+1}_I\right)^{-1} \| \Big[&\|\mathcal{L}^k_E\| \cdot \|e^k_{\partial_{\mu_i}}\| + \|(\partial_{\mu_i}\mathcal{L})^k_E\| \cdot \|u^k_h - u^k_N\| \\ &+\|(\partial_{\mu_i}\mathcal{L})^{k+1}_I\| \cdot \|u^{k+1}_h - u^{k+1}_N\| + \|\mathrm{Res}^{k+1}_{\partial_{\mu_i} u}\|\Big] . \end{aligned}$$

With $C_{\partial_{\mu_i} L_E} \geq \|(\partial_{\mu_i}\mathcal{L})^k_E\|$, $C_{\partial_{\mu_i} L_I} \geq \|(\partial_{\mu_i}\mathcal{L})^k_I\|$ and by using the error estimator for the solution $\|u^k_h - u^k_N\| \leq \Delta^k_{u_N}$ we obtain the recursion of the error

$$\|e^{k+1}_{\partial_{\mu_i}}\| \leq C_E(\boldsymbol{\mu})\|e^k_{\partial_{\mu_i}}\| + C_{\partial_{\mu_i} L_E}\Delta^k_{u_N} + C_{\partial_{\mu_i} L_I}\Delta^{k+1}_{u_N} + \|Res^{k+1}_{\partial_{\mu_i} u}\|.$$

Resolving this recursion yields the claimed error bound. □

Again it is possible to calculate the norm of the residual $\mathrm{Res}^k_{\partial_{\mu_i} u}$ rapidly online via

$$\begin{aligned} \|\mathrm{Res}^{k+1}_{\partial_{\mu_i} u}\|^2 =& \boldsymbol{c}^{k+1^T}\boldsymbol{K}^k_{IIi}\boldsymbol{c}^{k+1} + \boldsymbol{c}^{k^T}\boldsymbol{K}^k_{EEi}\boldsymbol{c}^k + \boldsymbol{a}^{k^T}\boldsymbol{K}^k_{\partial E\partial Ei}\boldsymbol{a}^k \\ &+ \boldsymbol{a}^{k+1^T}\boldsymbol{K}^k_{\partial I\partial Ii}\boldsymbol{a}^{k+1} + \boldsymbol{K}_{\partial b\partial bi} + 2\Big(-\boldsymbol{c}^{k+1^T}\boldsymbol{K}^k_{IEi}\boldsymbol{c}^k \\ &- \boldsymbol{c}^{k+1^T}\boldsymbol{K}^k_{I\partial Ei}\boldsymbol{a}^k - \boldsymbol{c}^{k+1^T}\boldsymbol{K}^k_{I\partial Ii}\boldsymbol{a}^{k+1} - \boldsymbol{c}^{k+1^T}\boldsymbol{K}^k_{I\partial bi} \\ &+ \boldsymbol{c}^{k^T}\boldsymbol{K}^k_{E\partial Ei}\boldsymbol{a}^k + \boldsymbol{c}^{k^T}\boldsymbol{K}^k_{E\partial Ii}\boldsymbol{a}^{k+1} + \boldsymbol{c}^{k^T}\boldsymbol{K}^k_{E\partial bi} \\ &+\boldsymbol{a}^{k^T}\boldsymbol{K}^k_{\partial E\partial Ii}\boldsymbol{a}^{k+1} + \boldsymbol{a}^{k^T}\boldsymbol{K}^k_{\partial E\partial bi} + \boldsymbol{a}^{k+1^T}\boldsymbol{K}^k_{\partial I\partial b}\Big) . \end{aligned}$$

All the matrices used are of low dimension $\mathbb{R}^{N\times N}, \mathbb{R}^{N\times N_{\partial_{\mu_i}}}, \mathbb{R}^{N_{\partial_{\mu_i}}\times N}, \mathbb{R}^N$ or $\mathbb{R}$. Their corresponding exact expressions can be found in appendix A.

5.3 Error estimator for the functional and the functional gradient

We define the approximation error of the functional value by our reduced basis approximation as

$$e_J(\boldsymbol{\mu}) := J_h(\boldsymbol{\mu}) - J_N(\boldsymbol{\mu}).$$

It is easy to see that in our case of a linear functional the norm of the error e_J can be bounded by

$$\begin{aligned} |e_J(\boldsymbol{\mu})| &= |J\left(u_h\left(\boldsymbol{\mu}\right) - u_N\left(\boldsymbol{\mu}\right)\right)| \\ &\leq \Delta_J(\boldsymbol{\mu}) := \|J\|_{l^q(\mathbb{T},X_h)}\|\boldsymbol{\Delta}_u\|_{l^q(\mathbb{T})} \end{aligned} \tag{37}$$

with $\|\cdot\|_{l^q(\mathbb{T},X_h)}$ being a discrete l^q-norm on a sequence of functions in X_h and $\boldsymbol{\Delta}_u = \left(\Delta^k_u\right)^K_{k=0}$ being the vector of error estimator values from Proposition 1 for all time steps. A similar error estimator can be found in [27]. Now we can express an upper bound for the approximation error of the functional gradient:

Proposition 3 *The approximation error*

$$e_{\nabla J}(\boldsymbol{\mu}) := \nabla_\mu J_h(\boldsymbol{\mu}) - \nabla_\mu J_N(\boldsymbol{\mu}) \tag{38}$$

can be bounded by

$$\|e_{\nabla J}(\boldsymbol{\mu})\|_{\mathbb{R}^p} \leq \Delta_{\nabla J}(\boldsymbol{\mu}) := \|J\|_{l^q(\mathbb{T},X_h)}\sqrt{\sum_{j=1}^p \|\boldsymbol{\Delta}_{\partial_{\mu_j} u}(\boldsymbol{\mu})\|^2_{l^q(\mathbb{T})}} \tag{39}$$

with $\boldsymbol{\Delta}_{\partial_{\mu_j} u}(\boldsymbol{\mu}) = \left(\Delta^k_{\partial_{\mu_j} u}(\boldsymbol{\mu})\right)_{k=0}^{K}$ *being the time sequence of error estimators from Proposition 2,* $\|\cdot\|_{\mathbb{R}^p}$ *being the euclidean norm and* $\|\cdot\|_{l^q(\mathbb{T})}$ *being a* l^q*-norm over the sequence of time steps.*

Proof Due to the linearity of the functional we can state that $e_{\nabla J} = \big(J(\partial_{\mu_1} u_h - \partial_{\mu_1} u_N), \dots, J(\partial_{\mu_p} u_h - \partial_{\mu_p} u_N)\big)^T$. Consequently

$$\begin{aligned}
\|e_{\nabla J}\|_{\mathbb{R}^p} &= \sqrt{\sum_{i=1}^{p} \left(J(\partial_{\mu_i} u_h(\boldsymbol{\mu}) - \partial_{\mu_i} u_N(\boldsymbol{\mu}))\right)^2} \\
&\leq \|J\|_{l^q(\mathbb{T}, X_h)} \sqrt{\sum_{i=1}^{p} \|\partial_{\mu_i} u_h(\boldsymbol{\mu}) - \partial_{\mu_i} u_N(\boldsymbol{\mu})\|^2_{l^q(\mathbb{T}, X_h)}} \\
&\leq \|J\|_{l^q(\mathbb{T}, X_h)} \sqrt{\sum_{i=1}^{p} \|\boldsymbol{\Delta}_{\partial_{\mu_i} u}\|^2_{l^q(\mathbb{T})}}
\end{aligned}$$

with $\boldsymbol{\Delta}_{\partial_{\mu_j} u} = \left(\Delta^k_{\partial_{\mu_j} u}(\boldsymbol{\mu})\right)_{k=0}^{K}$ being the time sequence of error estimators from Proposition 2. □

5.4 Error estimator for the optimal parameter $\boldsymbol{\mu}^*$

When solving the optimization problem with a reduced basis surrogate model we obtain a suboptimal solution $\boldsymbol{\mu}^*_N$ as minimizer of $J_N(\boldsymbol{\mu})$. The following error bound, which includes all the previous results, will allow a control of this suboptimality by bounding the error in the optimal parameters $\|\boldsymbol{\mu}^*_h - \boldsymbol{\mu}^*_N\|_{\mathbb{R}^p}$. Thereby we can certify the result of the optimal solution obtained by a reduced optimization.

We propose two different error estimators for the optimal parameters, each having its own advantages and drawbacks. The first one is a non-incremental error estimator inspired by the implicit function theorem and the general theory of nonlinear approximation problems of Caloz and Rappaz [7]. It can be applied to the results of any numerical optimization method used to solve $(\mathbb{P}_N)$. For that reason we call it "general" error estimator in the following. However, in order to apply the error estimator a reduced basis model of very good quality is needed and the optimization method applied must approach the optimum very closely, e.g. $\|\nabla J_N(\boldsymbol{\mu})\|_{\mathbb{R}^p}$ must be close to zero.

The second error estimator is an incremental error estimator for a gradient optimization method. There are no a-priori requirements to the quality of the surrogate model and it gives an error estimation in every step of the optimization iteration. However, in case that many gradient steps are needed to find the optimum, the effectivity of the error bounds is of low quality.

In the following we assume that $(\mathbb{P}_h)$ has a single stationary point $\boldsymbol{\mu}^*_h \in \mathring{\mathcal{P}}$ in the interior of the parameter domain being the minimum. Hence, $\boldsymbol{\mu}^*_h$ can be identified by the first order optimality criterion $\nabla_\mu J_h(\boldsymbol{\mu}^*_h) = 0$.

5.4.1 General error estimator

We assume that the reduced optimization problem $(\mathbb{P}_N)$ was solved using any numerical optimization procedure resulting in an approximate stationary point $\boldsymbol{\mu}^*_N$. Having the reduced approximation $\boldsymbol{\mu}^*_N$ we would like to bound the distance to the "true" optimum $\boldsymbol{\mu}^*_h$ which can be identified by $\nabla J_h(\boldsymbol{\mu}^*_h) = 0$.

Proposition 4 (Error Estimator optimal parameters) *Let* $\boldsymbol{\mu}^*_N \in \mathring{\mathcal{P}}$ *be the optimal parameter found solving the reduced optimization problem* $(\mathbb{P}_N)$ *satisfying the stopping criterion* $\|\nabla J_N(\boldsymbol{\mu}^*_N)\|_{\mathbb{R}^p} \leq \varepsilon_J$ *and introduce*

$\bar{B}(\boldsymbol{\mu}^*_N, \alpha)$ *a closed ball around* $\boldsymbol{\mu}^*_N$ *with radius* α*,*
$\nabla^2 J_h(\boldsymbol{\mu}^*_N) \in \mathbb{R}^{p\times p}$ *regular ,*
$\gamma := \|\nabla^2 J_h(\boldsymbol{\mu}^*_N)^{-1}\|_{\mathbb{R}^p}$ *with* $\|\cdot\|_{\mathbb{R}^p}$ *the induced matrix norm,*
$\varepsilon := \Delta_{\nabla J}(\boldsymbol{\mu}^*_N) + \varepsilon_J$*,*

$L(\alpha) := \sup_{\boldsymbol{\mu}\in\bar{\mathcal{B}}(\boldsymbol{\mu}_N^*,\alpha)} \|\nabla^2 J_h(\boldsymbol{\mu}_N^*) - \nabla^2 J_h(\boldsymbol{\mu})\|_{\mathbb{R}^p}$.

If the condition

$$2\gamma L(2\gamma\varepsilon) \leq 1 \tag{40}$$

holds, then there exists a unique solution $\boldsymbol{\mu}_h^*$ *to the optimization problem* $(\mathbb{P}_h)$ *with* $\boldsymbol{\mu}_h^* \in \bar{\mathcal{B}}(\boldsymbol{\mu}_N^*, 2\gamma\varepsilon)$ *and the rigorous error bound*

$$\|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} \leq 2\gamma\varepsilon \tag{41}$$

holds.

Proof The proof follows [7, Theorem 2.1] adapted to our optimization setting. Let $H : \mathbb{R}^p \to \mathbb{R}^p$ be the mapping defined by

$$H(\boldsymbol{\mu}) = \boldsymbol{\mu} - \nabla^2 J_h(\boldsymbol{\mu}_N^*)^{-1} \nabla J_h(\boldsymbol{\mu}). \tag{42}$$

We assume that the optimization problem $(\mathbb{P}_h)$ has an optimum $\boldsymbol{\mu}_h^*$ in $\mathcal{P}$. The solution $\boldsymbol{\mu}_h^*$ is a fixpoint of $H(\boldsymbol{\mu})$ as $\nabla J_h(\boldsymbol{\mu}_h^*) = 0$. Furthermore, a vector $\boldsymbol{\mu}$ can only be a fixpoint of H if $\nabla J_h(\boldsymbol{\mu}) = 0$.

For any $\boldsymbol{\mu} \in \bar{\mathcal{B}}(\boldsymbol{\mu}_N, 2\gamma\varepsilon)$ we can write

$$\begin{aligned} H(\boldsymbol{\mu}) - \boldsymbol{\mu}_N^* = {} & \nabla^2 J_h(\boldsymbol{\mu}_N^*)^{-1} \left[\nabla^2 J_h(\boldsymbol{\mu}_N^*)(\boldsymbol{\mu} - \boldsymbol{\mu}_N^*) - (\nabla J_h(\boldsymbol{\mu}) - \nabla J_h(\boldsymbol{\mu}_N^*))\right] \\ & - \nabla^2 J_h(\boldsymbol{\mu}_N^*)^{-1} \nabla J_h(\boldsymbol{\mu}_N^*). \end{aligned} \tag{43}$$

As the Hessian $\nabla^2 J_h(\boldsymbol{\mu})$ exists, we can write the Taylor expansion of $\nabla J(\boldsymbol{\mu})$

$$\nabla J_h(\boldsymbol{\mu}) = \nabla J_h(\boldsymbol{\mu}_N^*) + \int_0^1 \nabla^2 J_h(\boldsymbol{\mu}_N^* + t(\boldsymbol{\mu} - \boldsymbol{\mu}_N^*))(\boldsymbol{\mu} - \boldsymbol{\mu}_N^*) dt. \tag{44}$$

With (44) in (43) we obtain

$$\begin{aligned} H(\boldsymbol{\mu}) - \boldsymbol{\mu}_N^* = {} & \nabla^2 J_h(\boldsymbol{\mu}_N^*)^{-1} \Bigg[- \nabla J_h(\boldsymbol{\mu}_N^*) \\ & + \int_0^1 (\nabla^2 J_h(\boldsymbol{\mu}_N^*) - \nabla^2 J_h(\boldsymbol{\mu}_N^* + t(\boldsymbol{\mu} - \boldsymbol{\mu}_N^*))(\boldsymbol{\mu} - \boldsymbol{\mu}_N^*) \Bigg]. \end{aligned} \tag{45}$$

Consequently

$$\begin{aligned} \|H(\boldsymbol{\mu}) - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} \leq \|\nabla^2 J_h(\boldsymbol{\mu}_N^*)^{-1}\|_{\mathbb{R}^p} \Bigg[& \underbrace{\|\nabla J_h(\boldsymbol{\mu}_N^*)\|_{\mathbb{R}^p}}_{T_2} \\ & + \underbrace{\| \int_0^1 (\nabla^2 J_h(\boldsymbol{\mu}_N^*) - \nabla^2 J_h(\boldsymbol{\mu}_N^* + t(\boldsymbol{\mu} - \boldsymbol{\mu}_N^*))(\boldsymbol{\mu} - \boldsymbol{\mu}_N^*) \|_{\mathbb{R}^p}}_{T_1} \Bigg]. \end{aligned} \tag{46}$$

We can bound

$$T_1 \leq L(2\gamma\varepsilon) 2\gamma\varepsilon \leq \varepsilon \tag{47}$$

and

$$\begin{aligned} T_2 &\leq \|\nabla J_h(\boldsymbol{\mu}_N^*) - \nabla J_N(\boldsymbol{\mu}_N^*)\|_{\mathbb{R}^p} + \|\nabla J_N(\boldsymbol{\mu}_N^*)\|_{\mathbb{R}^p} \\ &\leq \Delta_{\nabla J}(\boldsymbol{\mu}_N^*) + \varepsilon_J \\ &= \varepsilon. \end{aligned} \tag{48}$$

Hence

$$\|H(\boldsymbol{\mu}) - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} \leq 2\gamma\varepsilon \tag{49}$$

So H maps the closed ball $\bar{\mathcal{B}}(\boldsymbol{\mu}_N^*, 2\gamma\varepsilon)$ into itself. Now we show that there is a unique fixpoint in $\bar{\mathcal{B}}(\boldsymbol{\mu}_N^*, 2\gamma\varepsilon)$.

Let $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ be in $\bar{\mathcal{B}}(\boldsymbol{\mu}_N^*, 2\gamma\varepsilon)$, then

$$H(\boldsymbol{\mu}_1) - H(\boldsymbol{\mu}_2) = \nabla^2 J_h(\boldsymbol{\mu}_N^*)^{-1} \int_0^1 \left(\nabla^2 J_h(\boldsymbol{\mu}_N^*) - \nabla^2 J_h(\boldsymbol{\mu}_1 + t(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1))\right)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \tag{50}$$

Hence

$$\begin{aligned} \|H(\boldsymbol{\mu}_1) - H(\boldsymbol{\mu}_2)\|_{\mathbb{R}^p} &\leq \gamma L(2\gamma\varepsilon)\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_{\mathbb{R}^p} \\ &\leq \frac{1}{2}\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_{\mathbb{R}^p}. \end{aligned} \tag{51}$$

This shows, that H is a strict contraction from $\bar{\mathcal{B}}(\boldsymbol{\mu}_N^*, 2\gamma\varepsilon)$ into itself. The Banach fixed-point theorem applies and states that there is a unique fixed point $\boldsymbol{\mu}_h^*$ of H in $\bar{\mathcal{B}}(\boldsymbol{\mu}_N^*, 2\gamma\varepsilon)$, hence, one unique point $\boldsymbol{\mu}_h^*$ where $\nabla J_h(\boldsymbol{\mu}) = 0$.

Now, for proving the error bound we put $\boldsymbol{\mu} = \boldsymbol{\mu}_h^*$ in (49) and obtain

$$\begin{aligned} \|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} &= \|H(\boldsymbol{\mu}_h^*) - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} \\ &\leq 2\gamma\varepsilon. \end{aligned} \tag{52}$$

□

Note that in order to apply the error estimator in practice, several constants (or the respective upper bounds) have to be calculated offline. In particular an upper bound for $\gamma(\boldsymbol{\mu}) \leq \bar{\gamma}$ and a constant $L_H \geq L(\alpha)$.

Remark 4 Note, that in the general case of a not strictly convex optimization problem there is no guarantee that the given bound around $\boldsymbol{\mu}_N^*$ contains the exact corresponding maximum or minimum in the detailed case. We can only state, that there is a stationary point $\boldsymbol{\mu}$ with $\nabla_\mu J_h(\boldsymbol{\mu}) = 0$ in the (calculated) vicinity of $\boldsymbol{\mu}_N^*$.

5.4.2 Incremental error estimator for a gradient descent method

Although the error estimator from the previous Section is a quite powerful and general tool to certify the results of the reduced optimization, it may be not applicable in some cases. For instance, it could be impracticable or undesired to satisfy the condition (40) in Proposition 4. It is impracticable, if the functional constants γ and L in Proposition 4 in combination with a reduced model of rather poor quality do not allow the fullfillement of the condition. Or it is undesired, if the objective is to have only a rough solution of the optimization problem and one does not want to push the optimization procedure far enough to obtain a small enough gradient norm ϵ_J in (40). In this case we need other error estimators.

In the following we propose an incremental error estimator designed for the gradient descent method. Hence, we assume the use of a gradient method to solve the optimization problem $\mathbb{P}_h$, starting the method with an initial parameter $\boldsymbol{\mu}_h^0$ and calculating the sequence $(\boldsymbol{\mu}_h^m)_{m=0}^M$ iteratively by

$$\boldsymbol{\mu}_h^m = \boldsymbol{\mu}_h^{m-1} + \alpha_h^{m-1} \cdot \boldsymbol{d}_h^{m-1}. \tag{53}$$

Here $\boldsymbol{d}_h^m = -\nabla_\mu J_h(\boldsymbol{\mu}_h^m) \in \mathbb{R}^p$ is the step direction and $\alpha_h^m \in \mathbb{R}$ is the stepsize. The stepsize is controlled by a stepsize rule, e.g. constant, as explicit function or adaptively with the Armijo-rule [11]. The stopping criterion for the algorithm is
$\|\nabla J_h(\boldsymbol{\mu}_h^M)\|_{\mathbb{R}^p} \leq \varepsilon_J$ for a suitable tolerance $\varepsilon_J > 0$. We will ensure in the applications, that all parameters $\boldsymbol{\mu}^m$ used during the gradient optimization are feasible, hence $\{\boldsymbol{\mu}^m\}_{m=0}^M \subset \mathcal{P}$. Note that this can be verified during the gradient descent.

Proposition 5 *Let $\boldsymbol{\mu}_h^M$ be the parameter vector from the detailed optimization problem $(\mathbb{P}_h)$ and $\boldsymbol{\mu}_N^M$ the parameter vector from the reduced optimization problem $(\mathbb{P}_N)$ obtained after M gradient descent steps. Assuming that both reduced and detailed optimization start at the same initial parameter $\boldsymbol{\mu}^0$ we can bound the approximation error of the optimal parameters by*

$$\|\boldsymbol{\mu}_h^M - \boldsymbol{\mu}_N^M\|_{\mathbb{R}^p} \leq \Delta_{\mu,incr}^M \tag{54}$$

with

$$\Delta_{\mu,incr}^M := \sum_{m=0}^{M-1} \left((1 + C_\alpha C_L)^{(M-1-m)} \left(\varepsilon^m \|\boldsymbol{d}_N^m\|_{\mathbb{R}^p} + C_\alpha \Delta_{\nabla J}(\boldsymbol{\mu}_N^m) \right) \right). \tag{55}$$

Here $\varepsilon^m \geq |\alpha_h^m - \alpha_N^m|$ is a bound for the stepsize error , C_α is an upper bound for the step size, C_L is the Lipschitz constant for the gradient $\nabla_\mu J_h$ and $\Delta_{\nabla J}(\boldsymbol{\mu}_N^m)$ is the error estimator for the output gradient from (39).

Proof We assume the initial parameters for the reduced and the detailed optimization to be equal $\|\boldsymbol{\mu}_h^0 - \boldsymbol{\mu}_N^0\|_{\mathbb{R}^p} = 0$. With (53) for the detailed and the reduced case we can state

$$\begin{aligned} \|\boldsymbol{\mu}_h^m - \boldsymbol{\mu}_N^m\|_{\mathbb{R}^p} &= \|\boldsymbol{\mu}_h^{m-1} - \boldsymbol{\mu}_N^{m-1} + \alpha_h^{m-1} \cdot \boldsymbol{d}_h^{m-1} - \alpha_N^{m-1} \cdot \boldsymbol{d}_N^{m-1}\|_{\mathbb{R}^p} \\ &\leq \|\boldsymbol{\mu}_h^{m-1} - \boldsymbol{\mu}_N^{m-1}\|_{\mathbb{R}^p} + \underbrace{\|\alpha_h^{m-1} \cdot \boldsymbol{d}_h^{m-1} - \alpha_N^{m-1} \cdot \boldsymbol{d}_N^{m-1}\|_{\mathbb{R}^p}}_{=T_1} \end{aligned} \tag{56}$$

The term T_1 can be rewritten as

$$\begin{aligned} T_1 &= \|\alpha_h^{m-1} \cdot \boldsymbol{d}_h^{m-1} - \alpha_h^{m-1} \cdot \boldsymbol{d}_N^{m-1} + \alpha_h^{m-1} \cdot \boldsymbol{d}_N^{m-1} - \alpha_N^{m-1} \cdot \boldsymbol{d}_N^{m-1}\|_{\mathbb{R}^p} \\ &\leq |\alpha_N^{m-1} - \alpha_h^{m-1}| \cdot \|\boldsymbol{d}_N^{m-1}\|_{\mathbb{R}^p} + |\alpha_h^{m-1}| \cdot \|\boldsymbol{d}_h^{m-1} - \boldsymbol{d}_N^{m-1}\|_{\mathbb{R}^p} \end{aligned} \tag{57}$$

and as the the stepsize α_h^m is bounded by $0 \leq \alpha_h^m \leq C_\alpha$ and $|\alpha_N^m - \alpha_h^m| \leq \varepsilon^m$

$$\Rightarrow T_1 \leq \varepsilon^{m-1} \|\boldsymbol{d}_N^{m-1}\|_{\mathbb{R}^p} + C_\alpha \underbrace{\|\boldsymbol{d}_h^{m-1} - \boldsymbol{d}_N^{m-1}\|_{\mathbb{R}^p}}_{T_2}. \tag{58}$$

The term T_2 can be reformulated as

$$\begin{aligned} T_2 \leq \|\nabla_\mu J_h(\boldsymbol{\mu}_h^{m-1}) - \nabla_\mu J_h(\boldsymbol{\mu}_N^{m-1})\|_{\mathbb{R}^p} \\ + \| + \nabla_\mu J_h(\boldsymbol{\mu}_N^{m-1}) - \nabla_\mu J_N(\boldsymbol{\mu}_N^{m-1})\|_{\mathbb{R}^p}. \end{aligned} \tag{59}$$

Assuming that $\nabla_\mu J_h(\boldsymbol{\mu})$ is Lipschitz continuous in $\boldsymbol{\mu}$ we can bound

$$\|\nabla_\mu J_h(\boldsymbol{\mu}_h^{m-1}) - \nabla_\mu J_h(\boldsymbol{\mu}_N^{m-1})\|_{\mathbb{R}^p} \leq C_L \|\boldsymbol{\mu}_N^{m-1} - \boldsymbol{\mu}_h^{m-1}\|_{\mathbb{R}^p} \tag{60}$$

with the Lipschitz constant C_L.
The second term in equation (59) corresponds to the output gradient error estimator $\Delta_{\nabla J}(\boldsymbol{\mu}_N^{m-1})$ in (39). Consequently, by putting (60) and (39) in (59) we obtain

$$T_2 \leq C_L \|\boldsymbol{\mu}_N^{m-1} - \boldsymbol{\mu}_h^{m-1}\|_{\mathbb{R}^p} + \Delta_{\nabla J}(\boldsymbol{\mu}_N^{m-1}). \tag{61}$$

With (61) and (58) in (57) we obtain the result for the parameter error estimator in optimization step m as

$$\begin{aligned} \|\boldsymbol{\mu}_h^m - \boldsymbol{\mu}_N^m\|_{\mathbb{R}^p} &\leq \|\boldsymbol{\mu}_h^{m-1} - \boldsymbol{\mu}_N^{m-1}\|_{\mathbb{R}^p} \\ &\quad + \varepsilon^{m-1} \|d_N^{m-1}\|_{\mathbb{R}^p} + C_\alpha \left(\Delta_{\nabla J}(\boldsymbol{\mu}_N^{m-1}) + C_L \|\boldsymbol{\mu}_h^{m-1} - \boldsymbol{\mu}_N^{m-1}\|_{\mathbb{R}^p} \right). \end{aligned} \tag{62}$$

By iteratively applying (62) we obtain (55). □

Note that the upper bound for the stepsize error ε^m is zero when using a constant or an algebraic function as stepsize rule. Yet, finding this upper bound for adaptive stepsize rules, e.g. Armijo, is nontrivial. We assume, that the operators $\mathcal{L}_E$ and $\mathcal{L}_I$ are Lipschitz-continuous with respect to the parameter vector $\boldsymbol{\mu}$ with bounded Lipschitz-constants C_{L_E}, C_{L_I}.

Note also, that the error estimator has an incremental character, hence will increase over the iterations. We therefore expect that the values are mainly useful for a low number of iterations. Still they can be arbitrarily small depending on the quality of the RB-spaces, as shown in the following corollary:

Corollary 1 (Verification of zero error) *The error estimator for the optimal parameter is zero $\Delta^M_{\mu,incr} = 0$ if and only if $(u_h^k(\boldsymbol{\mu}_h^m))_{k=0}^K \in X_N$ and $(\partial_{\mu_i} u_h^k(\boldsymbol{\mu}_h^m))_{k=0}^K \in X_{N_{\partial_{\mu_i}}}$ for all $m = 0, \dots, M$. In this case the reduced and detailed parameters $\boldsymbol{\mu}_h^m = \boldsymbol{\mu}_N^m = \boldsymbol{\mu}^m$ are equal as well as reduced and detailed solutions $u_h(\boldsymbol{\mu}^m) = u_N(\boldsymbol{\mu}^m)$ and derivative solutions $\partial_{\mu_i} u_h(\boldsymbol{\mu}^m) = \partial_{\mu_i} u_N(\boldsymbol{\mu}^m), \forall i = 1, \dots, p$.*

Proof If $\Delta^M_{\mu,incr} = 0$ then $\Delta^m_{\mu,incr} = 0\ \forall m = 0, \dots, M$. Hence $\boldsymbol{\mu}_h^m = \boldsymbol{\mu}_N^m = \boldsymbol{\mu}^m\ \forall m = 0, \dots, M$. Due to positivity of error components in (55) $\Delta_u(\boldsymbol{\mu}^m) = \Delta_{\partial_{\mu_i}}(\boldsymbol{\mu}^m) = 0$. So $u_N(\boldsymbol{\mu}^m) = u_h(\boldsymbol{\mu}^m)$ and $\partial_{\mu_i} u_N(\boldsymbol{\mu}^m) = \partial_{\mu_i} u_h(\boldsymbol{\mu}^m)$.

On the other hand, if $(u_h^k(\boldsymbol{\mu}_h^m))_{k=0}^K \in X_N$ and $(\partial_{\mu_i} u_h^k(\boldsymbol{\mu}_h^m))_{k=0}^K \in X_{N_{\partial_{\mu_i}}}$ for all $m = 0, \dots, M$ then the residuals (31) and (35) are zero $\forall k = 0, \dots, K$ and $\forall m = 1, \dots, M$. Hence, $\Delta_u(\boldsymbol{\mu}^m) = \Delta_{\partial_{\mu_i} u}(\boldsymbol{\mu}^m) = 0$. Assuming a stepsize rule defined by an algebraic function, it is $\varepsilon^m = 0$ in (55) and all summands in (55) equal zero, consequently $\Delta^m_{\mu,incr} = 0\ \forall m = 0, \dots, M$. □

Corollary 2 *If both optimization procedures in the detailed as well as in the reduced case converge well towards an optimum $\boldsymbol{\mu}_h^*$ and $\boldsymbol{\mu}_N^*$ respectively, so that $\|\boldsymbol{\mu}_h^M - \boldsymbol{\mu}_h^*\|_{\mathbb{R}^p} \leq \varepsilon$ and $\|\boldsymbol{\mu}_N^M - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} \leq \varepsilon$ then the difference between the optimum in the detailed case and in the reduced case is*

$$\|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} \leq \Delta^M_{\mu,incr} + 2\varepsilon. \tag{63}$$

Proof A combination of Proposition 5 with the triangle inequality yields:

$$\begin{aligned} \|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p} &= \|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_h^M + \boldsymbol{\mu}_h^M - \boldsymbol{\mu}_N^* + \boldsymbol{\mu}_N^M - \boldsymbol{\mu}_N^M\|_{\mathbb{R}^p} \\ &\leq \|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_h^M\|_{\mathbb{R}^p} + \|\boldsymbol{\mu}_N^* - \boldsymbol{\mu}_N^M\|_{\mathbb{R}^p} + \|\boldsymbol{\mu}_h^M - \boldsymbol{\mu}_N^M\|_{\mathbb{R}^p} \\ &\leq \varepsilon + \varepsilon + \Delta^M_{\mu,incr}. \end{aligned} \tag{64}$$

□

6 Experiments

In the experiments we consider the PDE constrained optimization problem

$$\begin{cases} \max_{\mu \in \mathcal{P}} J(u(\boldsymbol{\mu})) \\ \text{s.t.} \\ \partial_t u(\mu) = \Delta\left(ku(\boldsymbol{\mu})\right) - \nabla \cdot (\boldsymbol{v}(\mu)u(\mu)) \text{ in } \Omega \times [0,T] \end{cases} \tag{65}$$

where the PDE represents a parametrized advection diffusion problem and J is a linear functional being specified later. We set $\Omega := [0,2] \times [0,1]$ and final time $T = 1$. We assume suitable initial conditions $u^0(\boldsymbol{\mu}) = u(\cdot, t = 0; \boldsymbol{\mu})$. Then, Dirichlet boundary conditions $u(\boldsymbol{\mu}) = u_{dir}(\boldsymbol{\mu})$ on $\Gamma_{dir} \times [0,T]$ with $\Gamma_{dir} := (\{\boldsymbol{x} = (x_1, x_2) | x_1 = 0, x_2 \in \mathbb{R}\} \cup \{\boldsymbol{x} = (x_1, x_2) | x_1 \in \mathbb{R}, x_2 = 1\}) \cap \partial\Omega$ as well as outflow boundary conditions $u(\boldsymbol{\mu}) = u_{out}(\boldsymbol{\mu})$ on $\Gamma_{out} \times [0,T]$ with $\Gamma_{out} := (\{\boldsymbol{x} = (x_1, x_2) | x_1 = 2, x_2 \in \mathbb{R}\} \cup \{\boldsymbol{x} = (x_1, x_2) |\ x_1 \in \mathbb{R}, x_2 = 0\}) \cap \partial\Omega$ are prescribed. The velocity $\boldsymbol{v}$ is supposed to be a divergence free parameter and time dependent velocity field of the form

$$\boldsymbol{v}(\boldsymbol{x}, t; \boldsymbol{\mu}) = \left(\mu_1(1-t) \cdot 5(1-x_2^2), -\mu_2(1-t)(4-x_1^2)\right)^T. \tag{66}$$

The diffusion is assumed to be homogeneous, hence k is a constant. The parameters $(\mu_1, \mu_2) = \boldsymbol{\mu}$ stem from a two dimensional parameter domain $\boldsymbol{\mu} \in \mathcal{P} = [0,1]^2$. They influence the strength of the x_1 and x_2 components of the velocity field. This problem can be discretized with cell-wise constant functions

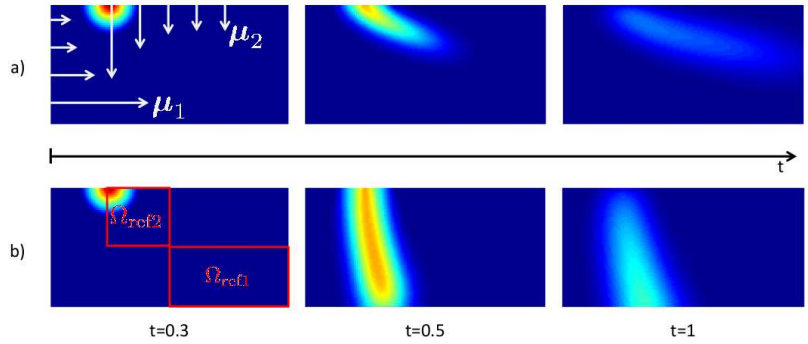


Fig. 1 Illustration of the problem setting and of two solution trajectories at chosen timesteps for exemplary parameters. a) Solution to the parameter $\boldsymbol{\mu} = (0.2, 1)$ at times $t = 0$, $t = 0.3$ and $t = 1$. In the first picture the velocity field is schematically illustrated. b) Solution to the parameter $\boldsymbol{\mu} = (1, 0.2)$ at times $t = 0$, $t = 0.3$ and $t = 1$. The first image shows the position of the two reference domains used in the optimization.

and a Finite Volume (FV) scheme using an Engquist–Osher flux, which results in a corresponding discretization space X_h and discretization operators $\mathcal{L}_I$ and $\mathcal{L}_E$ as well as in a discrete right hand side b_h. We chose a space discretization by a 128×64 rectangular grid leading to 8192 degrees of freedom. For satisfying the CFL conditions we discretized in time into 1024 time steps. In the Finite Volume discretization we realized an operator splitting with the implicit operator $\mathcal{L}_I$ containing the diffusion operator and the explicit part $\mathcal{L}_E$ containing the advection operator. Details concering the discretization can be found in [19]. Solutions for some chosen parameters at different time instances are shown in Figure 1.

We consider this FV model as high-dimensional "detailed" model. Using this discrete model we build up the low dimensional reduced basis model. The basis generation is conducted with a POD-Greedy algorithm [19] with T-partition [10] into ten intervals and a P-partition [17] with 10×10 partitions. Furthermore, the separate basis ansatz described in Section 4.2 is applied. Note, that the offline effort is considerable and can take several days. Hence, as typical in many RB methods this procedure is not recommended if only one optimization problem needs to be solved. It can be beneficial, if many optimization problems have to be solved (e.g. many functionals, parametric optimization problems, model predictive control, etc.) or other parametric or real-time tasks are to be expected for the reduced model.

6.1 Comparison of different RB approaches

In the following we will study the effect of separate reduced basis spaces in order to obtain an efficient evaluation of the parameter derivative solutions. For that purpose we generate reduced basis spaces in three different ways. All three use the POD-Greedy algorithm [19] using slight variations. The first reduced basis is constructed in the "classic" way considering only the error estimator for $\|u_h - u_N\|$ during the construction process. This single reduced space is later used for representing the solutions u_N as well as the derivative solutions $\partial_{\mu_i} u_N$. In the second reduced basis generation procedure, called "mixed" approach, we add in every interation of the POD-Greedy two new basis vectors. The first one is derived from the solution $u_h(\boldsymbol{\mu})$ for the parameter indicating the highest error estimator after the Greedy search and the second one is derived from the derivative solution $\partial_{\mu_i} u_h(\boldsymbol{\mu})$ for the parameter indicating the highest error estimator for all the derivative solutions over the same training set of parameters. Again this results in one single reduced space which we use to approximate the solution and the derivative solutions. The third basis generation process, as described in Section 4.4, produces "separate" reduced basis spaces for the solution and all the derivative solutions. Here, the POD-Greedy procedure is applied separately for the original and the derivative evolution problems. For the experiment we construct reduced spaces of different sizes using these three approaches. Then we conduct reduced simulations for a test set of 50 randomly chosen parameters with all of the reduced basis spaces and note the average online simulation time as well as the average error estimator for one derivative solution over the whole test set. The results are illustrated in Figure 2. We observe

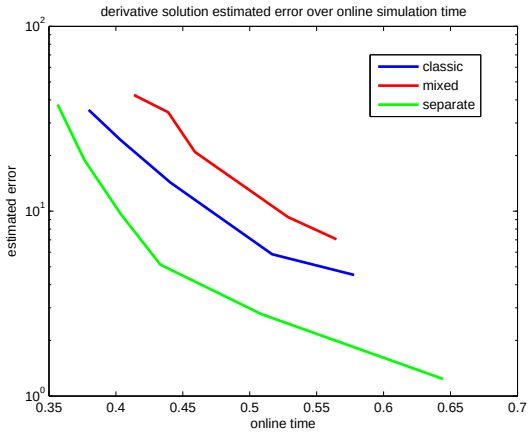


Fig. 2 Comparison of different approaches to approximate the derivative solution. Differently generated reduced models of varying sizes were tested over a sample of parameters and the average error estimator value is plotted over the average online simulation time.

that the online solves for the derivative solution for a desired error tolerance are always conducted faster by the separate approach than by the two others. The "classic" reduced space is well suited to approximate solutions but not necessarily to represent well the derivative solutions. Of course, the number of basis vectors in the reduced basis can be augmented and thereby also the derivative solutions can be approximated arbitrarily good. However, as the online simulation time grows with $\mathcal{O}(N^\beta)$ (with $\beta > 1$) the online procedure gets more and more inefficient. The same effect can be observed when using the "mixed" approach. Here we also approximate solution and derivative solutions by the same reduced basis and the same argumentation as above holds. When using separate bases the total number of basis vectors $N_s + \sum_{i=1}^{p} N_{\partial_{\mu_i}}$ (with N_s the number of basis vectors for the solution and $N_{\partial_{\mu_i}}$, $i = 1, \dots, p$ the number of basis vectors for the derivative solution) may be bigger than the basis size N in the other approaches, yet, the online simulation complexity of $\mathcal{O}(N_s^\beta + \sum_{i=1}^{p} N_{\partial_{\mu_i}}^\beta)$ is smaller than $\mathcal{O}(N^\beta)$. Consequently the separate bases approach is better suited to efficiently approximate u_h and the derivative solutions $\partial_{\mu_i} u_h$ simultaneously.

6.2 Parameter optimization with a reduced basis surrogate model

One of the advantages of the optimization approach presented in this work is the fact, that the costly build up of the reduced basis surrogate model has to be done only once in the offline phase and in the rapid online phase a functional for the optimization can be chosen freely. To illustrate this flexibility, we will solve two optimization problems with the setting as in (65) for two different linear functionals J_1 and J_2. For both optimization problems we use the same reduced basis surrogate model generated in advance without being obliged to perform any time consuming adaptations when switching between the functionals.

The objective of the first optimization problem is to maximize the functional

$$J_1(u(\boldsymbol{\mu})) := \frac{1}{|\Omega_{ref1}|} \int_{\Omega_{ref1}} u(\cdot, T; \boldsymbol{\mu})$$

returning the average concentration at the final time step in the first reference domain Ω_{ref1}. The second optimization problem with the functional

$$J_2(u(\boldsymbol{\mu})) := \frac{1}{|\Omega_{ref2}|0.25} \int_{t=0.5}^{0.75} \int_{\Omega_{ref2}} u(\cdot, \cdot; \boldsymbol{\mu})$$

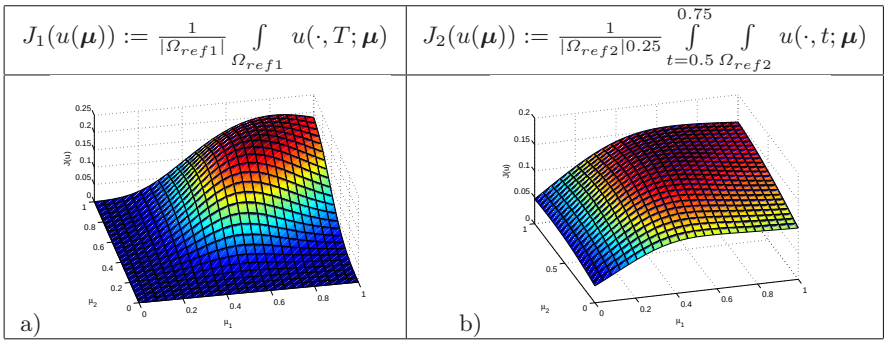


Fig. 3 The values of the functionals a) $J_1(u(\boldsymbol{\mu}))$ and b) $J_2(u(\boldsymbol{\mu}))$ plotted for various values of μ_1 and μ_2.

aims to maximize the average concentration in the time interval $[0.5, 0.75]$ in the second reference domain Ω_{ref2}. The reference domains Ω_{ref1} and Ω_{ref2} are indicated in Figure 1.

In a first step we plot the functionals' landscapes by solving the PDE and calculating the functional values for different values of $\boldsymbol{\mu}$. This is illustrated in Figure 3. Note, that the use of a reduced basis model rather than that of the original discrete model is already beneficial for "exploring" the functional landscape of a PDE-constrained optimization problem. By this quick exploration one can obtain a first impression of the behaviour of the optimization problem, see how many maxima and minima the problem probably offers and where they are approximately situated.

We see in Figure 3 that both opimization problems have (probably) a unique maximum. To demonstrate once more the flexibility of our method in the sense that it is applicable with any numerical optimization technique, we conduct a solve of the two problems, first using a gradient descent method with the Armijo-rule for stepsize control and secondly the Nelder-Mead algorithm. In order to obtain the gradient of the functional we calculate a solution to the according sensitivity PDE as described in Section 3. For verification purposes we solve the optimization problem numerically using both, the detailed high dimensional model and the reduced basis surrogate model. One difficulty in numerical optimization is the fact, that the results can depend on the choice of the initial parameters in the optimization procedure. However, the best choice for initial paramters (for fast convergence and convergence to the right optimum) is usually not known a priori. To account for that and show the overall performance of our method, we perform optimizations for a sample of 30 randomly chosen initial parameters and average all results over this sample. In this experiment we set the tolerance ε_J to 0.0005. Note, that the quality of the error estimator depends on the value chosen for ε_J. We will focus on this dependence in a subsequent experiment. The experiments were conducted on a machine with two AMD Quad-Core processors and 32GB RAM.

When comparing the time needed for a detailed optimization (t_{opt} det.) with the time needed for a reduced optimization (t_{opt} red.) in Table 1 we see that the optimization using the reduced basis surrogate model is about ten times faster. The relatively high computation time for the reduced optimization is due to the fact that the problem is linear time variant and therefore the reduced operators have to be assembled by the linear combination (26) of coefficients and components in every time step of the simulations. The online time for reduced simulations is much lower when treating time invariant problems. More important than the reduction in calculation time, however, is the reduction in dimension. The results in Table 1 make clear that in order to solve the PDE-constrained optimization problem accurately we only need a reduced basis surrogate model of an average dimension of about $N \approx N_{\partial_{\mu 1}} \approx N_{\partial_{\mu 2}} \approx 60$. Compared with the original discrete model's dimension of 8192 this is a reduction of about 99.3%. Note, that due to the fact that we chose a rather low dimensional example and implemented the numerical schemes in Matlab, the huge dimension reduction in our experiment does not reflect in an equally large reduction in computation time. However, when applying the method to discrete models of larger dimension the gain in computation time is expected to be considerably higher. As already mentioned, the accuracy of the optimization with a reduced basis surrogate model turns out to be very good, but also the general error estimator Δ_μ provides useful upper bounds for the optimization error in order to certify the reduced optimization. Regarding results in Table 1 for both numerical optimization algorithms applied, it gets obvious that the good results are not due to the

Table 1 Results of the parameter optimization with reduced and detailed models for two different functionals J_1 and J_2. Each optimization problem was solved using a gradient descent method (lines 1 and 2) and the Nelder-Mead (NM) algorithm (lines 3 and 4). The values are averaged ovar a sample of 30 optimization procedures for randomly chosen starting parameters. In the table are indicated: The average size of dimensions used for the PDE-solves with the reduced model (in brackets the dimensions used for the derivative solutions), time needed to conduct an optimization using the detailed model (t_{opt} det.) and the reduced model (t_{opt} red.), the true error in the parameters found by the detailed and the reduced optimization ($\|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p}$), the general error estimator (Δ_μ).

Meth.	Funct.	dim(RB) used	t_{opt} det.	t_{opt} red.	$\|\boldsymbol{\mu}_h^* - \boldsymbol{\mu}_N^*\|_{\mathbb{R}^p}$	Δ_μ
gradient	J_1	59.2(57.4; 59.5)	8.06 h	1.39 h	$7.37 \cdot 10^{-8}$	0.0063
	J_2	60.0(59.0; 61.6)	6.99 h	0.623 h	$7.62 \cdot 10^{-8}$	0.0121
NM	J_1	-	4.53 h	0.464 h	$4.22 \cdot 10^{-6}$	0.0046
	J_2	-	4.92 h	0.461 h	0.00161	0.00908

Table 2 Results of a parameter optimization using a gradient method conducted for both optimization problems with functionals J_1 and J_2. For every gradient step the norm of the functional gradient $\|\nabla J_i\|$, the value of the general error estimator Δ_μ, the value of condition (40) (which should be ≤ 1 to provide applicability of the general error estimator) and the value of the incremental error estimator $\Delta_{\mu,incr}^m$ are noted.

	step# (m)	0	1	2	3	4	10
J_1	$\|\nabla J_1(\boldsymbol{\mu}^m)\|$	0.396	0.106	0.0724	0.0410	0.0151	$9.93 \cdot 10^{-4}$
	Δ_μ	-	-	-	0.0885	0.0345	0.00436
	$2\gamma L(2\gamma\varepsilon)$	26.8	1.14	1.28	0.812	0.322	0.0423
	$\Delta_{\mu,incr}^m$	0	$4.03 \cdot 10^{-4}$	0.00190	0.00589	0.0167	1.39
	step# (m)	0	1	2	3	4	8
J_2	$\|\nabla J_2(\boldsymbol{\mu}^m)\|$	0.125	0.0644	0.0102	0.0057	0.00330	$3.92 \cdot 10^{-4}$
	Δ_μ	-	-	0.0232	0.0241	0.00915	0.00223
	$2\gamma L(2\gamma\varepsilon)$	2.84	13.6	0.678	0.413	0.282	0.0717
	$\Delta_{\mu,incr}^m$	0	$5.72 \cdot 10^{-4}$	0.0029	0.0102	0.0335	3.282

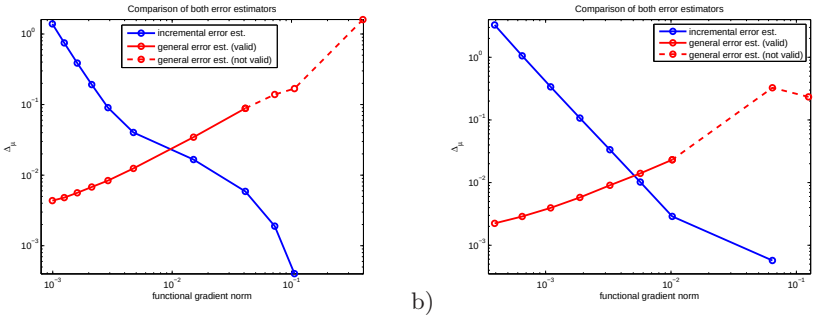
use of a specific optimization technique. The reduced optimization itself as well as the error estimator work equally well for the gradient method as well as for the Nelder-Mead algorithm.

The quality and the applicability of the general error estimator from Proposition 4 depend (amongst other quantities) on the choice of the optimization tolerance ε_J. Hence, in the next experiment we examine this dependence by starting a gradient optimization from a given initial parameter $\boldsymbol{\mu}^0 = (0.5, 0.5)$ and note at every gradient step on the way to the optimum $\boldsymbol{\mu}^* = (0.812, 0.696)$ the norm of the gradient $\|\nabla J_1\|$ (which links directly to the optimization tolerance ε_J), the value of the general error estimator Δ_μ and the value of the applicability condition (40) $2\gamma L(2\gamma\varepsilon) \leq 1$. The same is done for the second optimization problem with J_2 starting from $\boldsymbol{\mu}^0 = (0.2, 0.2)$ going to $\boldsymbol{\mu}^* = (0.417, 0.327)$. Note, that for this optimization run we do not use the Armijo stepsize control, but apply a quotient rule ($\alpha^m = \frac{1}{m+2}$). The results can be found in Table 2.

We see that in the case of high optimization tolerances the general error estimator can not be applied due to the fact that the condition (40) is not fulfilled. In this case it could be helpful to use the iterative error estimator from Proposition 5 as an alternative. Hence, we conduct the same experiment starting from the same initial parameters and tracking the values of the incremental error estimator. The results of this experiment can also be found in Table 2. It turns out that for high optimization tolerances, or more specific, in cases where not many gradient iteration steps are needed, the incremental error estimator can be useful, while the non-incremental general error estimator may not be applicable. However, when a high accuracy in the solution of the optimization problem is required, the general error estimator gives far better results, due to the fact that the value of the iterative error estimator augments with every gradient step. This is illustrated for both functionals in Figure 4.

7 Conclusion

We presented an approach for solving PDE-constrained parameter optimization problems rapidly using reduced basis surrogate models. The method consists of an offline-phase where a reduced basis model



a) b)

Fig. 4 Both, the general estimator Δ_μ as well as the incremental error estimator $\Delta_{\mu,incr}$ are plotted over the norm of the functional gradient for the first optimization problem with J_1 a) and for the second optimization problem with J_2 b). The dots mark the gradient steps (iterations from right to left). Furthermore, it is marked by dashed lines where the general error estimator is not valid due to the nonfulfillment of condition (40).

for the PDE-constraint is generated. In the online phase the parameter optimization problem can be solved using the low dimensional reduced basis surrogate model. This leads to a considerably lower online computational complexity and thereby less computation time. As our scheme provides derivative information, the approach can be used with a wide range of numerical optimization techniques. By using a sensitivity PDE to obtain this derivative information, the reduced basis surrogate model is not linked to a given optimization problem with a given functional but the method provides the possibility to choose freely a linear output functional in the online phase or solve parametrized optimization problems.

However, using a surrogate model during the optimization can lead to a suboptimal solution. We showed, that this suboptimality can be controlled by a rigorous a-posteriori error bound for the optimal parameters. We presented a general error estimator which can be used in combination with any numerical optimization scheme. If, however, the conditions for a valid application of this error estimator are not given, one has to use error estimators specific to the current problem. We presented such an error estimator for the suboptimality in a gradient descent method. Furthermore, we presented error estimators for the reduced solution and derivative solutions as well as for the reduced functional and the functional gradient. All these error bounds can be computed rapidly during the online phase. In the experiments we studied the application of our approach for two optimization problems with a parametrized advection diffusion problem as PDE-constraint.

So far our method is limited to linear functionals. We will extend our approach to nonlinear functionals (particularly quadratic functionals) in future work. Furthermore it could be interesting to investigate alternatives for the suboptimality error estimators (e.g. for SQP methods).

Acknowledgements The authors would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart and the Baden-Württemberg Stiftung gGmbH.

References

1. A.C. Antoulas. An overview of approximation methods for large-scale dynamical systems. *Annu. Rev. Contr.*, 29:181–190, 2005.
2. 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. Math. Acad. Sci. Paris Series I*, 339:667–672, 2004.
3. L.T. Biegler. *Large-scale PDE-constrained optimization.* Springer, 2003.
4. J. Borggaard and J. Burns. A PDE sensitivity equation method for optimal aerodynamic design. *Journal of Computational Physics*, 136:366–384, 1997.
5. J. Borggaard, J.A. Burns, A Surana, and L. Zietsman. Control, Estimation and Optimization of Energy Efficient Buildings. In *American Control Conference*, 2009.

6. T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.
7. G. Caloz and J. Rappaz. *Numerical analysis for nonlinear and bifurcation problems*, volume V of *Techiques of Scientific Computing (Part2)*. Elsevier, 1997.
8. J. Cea. Conception optimale ou identification de formes calcul rapide de la derive directionelle de la fonction cout. *Mathematical Modelling and Numerical Analysis*, 20(3):371 – 402, 1986.
9. L. Dede. Reduced basis method and a posteriori error estimation for parametrized optimal control problems with control constraints. *Journal of Scientific Computing*, 2011.
10. M. Dihlmann, M. Drohmann, and B. Haasdonk. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. In *Proc. of ADMOS 2011*, 2011.
11. C. Geiger and C. Kanzow. *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*. Springer, 1999.
12. M. Grepl and M. Kärcher. Reduced basis a posteriori error bounds for parametrized linear-quadratic elliptic optimal control problems. *C.R. Acad Sci. Paris, Ser.1*, 349:873–877, 2011.
13. M.A. Grepl. *Reduced-basis Approximations and a Posteriori Error Estimation for Parabolic Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, May 2005.
14. M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN, Math. Model. Numer. Anal.*, 39(1):157–181, 2005.
15. R. Griesse and B. Vexler. Numerical sensitivity analysis for the quantity of interest in PDE-constrained optimization. *SIAM J. of Scientific Computing*, 29(1):22 – 48, 2007.
16. B. Haasdonk. Convergence rates of the POD-Greedy method. *ESAIM: Mathematical Modelling and Numerical Analysis*, Accepted., 2012.
17. B. Haasdonk, M. Dihlmann, and M. Ohlberger. A training set and multiple basis generation approach for parametrized model reduction based on adaptive grids in parameter space. *Mathematical and Computer Modelling of Dynamical Systems*, 17:423–442, 2011.
18. B. Haasdonk and M. Ohlberger. Adaptive basis enrichment for the reduced basis method applied to finite volume schemes. In *Proc. 5th International Symposium on Finite Volumes for Complex Applications*, pages 471–478, 2008.
19. B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *M2AN, Math. Model. Numer. Anal.*, 42(2):277–302, 2008.
20. B. Haasdonk, M. Ohlberger, and G. Rozza. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *ETNA, Electronic Transactions on Numerical Analysis*, 32:145–161, 2008.
21. A. Hay, I. Akhtar, and J. Borggaard. On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions. *International Journal for Numerical Methods in Fluids*, 2011.
22. M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer, 2009.
23. M. Hinze and F. Tröltzsch. Discrete concepts versus error analysis in PDE-constrained optimization. *GAMM-Mitteilungen*, 33:148 – 162, 2010.
24. F. Jarre and J. Stoer. *Optimierung*. Springer, 2004.
25. T. Lassila and G. Rozza. Parametric free-form shape design with PDE models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 199:1583–1592, 2010.
26. R.M. Lewis, A.T. Patera, and J. Peraire. A posteriori finite element bounds for sensitivity derivatives of partial-differential-equation outputs. *Finite Elements in Analysis and Design*, 34:271–290, 2000.
27. N.C. Nguyen, G. Rozza, D.B.P. Huynh, and A. Patera. Reduced basis approximation and a posteriori error estimation for parametrized parabolic PDEs; application to real-time Bayesian parameter estimation. *John Wiley & Sons*, 2010.
28. I.B. Oliveira and A.T. Patera. Reduced-basis techniques for rapid reliable optimization of systems described by affinely parametrized coercive elliptic partial differential equations. *Optim Eng*, 8:43–65, 2007.
29. A. Quarteroni, G. Rozza, and A. Quaini. Reduced basis methods for optimal control of advection-diffusion problems. In *Advances in Numerical Mathematics*, number MOX 79, pages 193–216,

2006.
30. G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced Basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Meth. Eng.*, 15(3):229–275, 2008.
31. E. Sachs and S. Volkwein. POD-Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen*, 33, No.2:194–208, 2010.
32. F. Tröltzsch and S. Volkwein. POD a-posteriori error estimation for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44(1):83–115, 2009.
33. K. Veroy, C. Prud'homme, D. V. Rovas, and A. T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *In Proceedings of 16th AIAA computational fluid dynamics conference*, 2003. Paper 2003-3847.

A Matrices for the error estimator of $\partial_{\mu_i} u_N$

$$\left(\boldsymbol{K}^k_{IIi}\right)_{n,m} = \left\langle \mathcal{L}^{k+1}_I(\psi_{n,i}), \mathcal{L}^{k+1}_I(\psi_{m,i}) \right\rangle \tag{67}$$

$$\left(\boldsymbol{K}^k_{EEi}\right)_{n,m} = \left\langle \mathcal{L}^{k}_E(\psi_{n,i}), \mathcal{L}^{k}_E(\psi_{m,i}) \right\rangle \tag{68}$$

$$\left(\boldsymbol{K}^k_{\partial E\partial Ei}\right)_{n,m} = \left\langle (\partial_{\mu_i}\mathcal{L})^k_{h,Ex}(\varphi_n), (\partial_{\mu_i}\mathcal{L})^k_{h,Ex}(\varphi_m) \right\rangle \tag{69}$$

$$\left(\boldsymbol{K}^k_{\partial I\partial Ii}\right)_{n,m} = \left\langle (\partial_{\mu_i}\mathcal{L})^{k+1}_{h,Im}(\varphi_n), (\partial_{\mu_i}\mathcal{L})^{k+1}_{h,Im}(\varphi_m) \right\rangle \tag{70}$$

$$\left(\boldsymbol{K}^k_{\partial b\partial bi}\right) = \left\langle (\partial_{\mu_i} b)^k_h, (\partial_{\mu_i} b)^k_h \right\rangle \tag{71}$$

$$\left(\boldsymbol{K}^k_{IEi}\right)_{n,m} = \left\langle \mathcal{L}^{k+1}_I(\psi_{n,i}), \mathcal{L}^{k}_E(\psi_{m,i}) \right\rangle \tag{72}$$

$$\left(\boldsymbol{K}^k_{I\partial Ei}\right)_{n,m} = \left\langle \mathcal{L}^{k+1}_I(\psi_{n,i}), (\partial_{\mu_i}\mathcal{L}^k_E)(\varphi_m) \right\rangle \tag{73}$$

$$\left(\boldsymbol{K}^k_{I\partial Ii}\right)_{n,m} = \left\langle \mathcal{L}^{k+1}_I(\psi_{n,i}), (\partial_{\mu_i}\mathcal{L}^{k+1}_I)(\varphi_m) \right\rangle \tag{74}$$

$$\left(\boldsymbol{K}^k_{I\partial bi}\right)_{n} = \left\langle \mathcal{L}^{k+1}_I(\psi_{n,i}), (\partial_{\mu_i} b^k_h) \right\rangle \tag{75}$$

$$\left(\boldsymbol{K}^k_{E\partial Ei}\right)_{n,m} = \left\langle \mathcal{L}^{k}_E(\psi_{n,i}), (\partial_{\mu_i}\mathcal{L}^k_E)(\varphi_m) \right\rangle \tag{76}$$

$$\left(\boldsymbol{K}^k_{E\partial Ii}\right)_{n,m} = \left\langle \mathcal{L}^{k}_E(\psi_{n,i}), (\partial_{\mu_i}\mathcal{L}^{k+1}_I)(\varphi_m) \right\rangle \tag{77}$$

$$\left(\boldsymbol{K}^k_{E\partial bi}\right)_{n,m} = \left\langle \mathcal{L}^{k}_E(\psi_{n,i}), (\partial_{\mu_i} b^k_h) \right\rangle \tag{78}$$

$$\left(\boldsymbol{K}^k_{\partial E\partial Ii}\right)_{n,m} = \left\langle (\partial_{\mu_i}\mathcal{L}^k_E)(\varphi_n), (\partial_{\mu_i}\mathcal{L}^{k+1}_I)(\varphi_m) \right\rangle \tag{79}$$

$$\left(\boldsymbol{K}^k_{\partial E\partial bi}\right)_{n} = \left\langle (\partial_{\mu_i}\mathcal{L}^k_E)(\varphi_n), (\partial_{\mu_i} b^k_h) \right\rangle \tag{80}$$

$$\left(\boldsymbol{K}^k_{\partial I\partial bi}\right)_{n} = \left\langle (\partial_{\mu_i}\mathcal{L}^{k+1}_I)(\varphi_n), (\partial_{\mu_i} b^k_h) \right\rangle \tag{81}$$