Universität Konstanz

Greedy Sampling of Distributed Parameters in the Reduced-Basis Method by Numerical Optimization

Laura Iapichino
Stefan Volkwein

Konstanzer Schriften in Mathematik

Nr. 315, Juni 2013

ISSN 1430-3558

Greedy Sampling of Distributed Parameters in the Reduced-Basis Method by Numerical Optimization

Laura Iapichino

University of Konstanz, Department of Mathematics and Statistics, Universitätsstraße 10, D-78457 Konstanz, Germany

Stefan Volkwein[*]

University of Konstanz, Department of Mathematics and Statistics, Universitätsstraße 10, D-78457 Konstanz, Germany

Abstract

In the present paper the authors study second-order elliptic parametric partial differential equations ($\boldsymbol{\mu}$PDEs), where the parameters are scalars or distributed functions. By utilizing a modified greedy algorithm a reduced-basis approximation is derived. This new strategy combines the classical greedy algorithm with techniques from PDE constrained optimization. Numerical examples for the Graetz problem illustrate the efficiency of the strategy to handle not only scalar, but also distributed parameter functions.

Keywords: Second-order elliptic equations, reduced basis method, greedy algorithm, nonlinear optimization, Graetz problem
2010 MSC: 35J15, 49M37, 65K10, 90C30

1. Introduction

The reduced basis (RB) method is an efficient technique to solve parametric partial differential equations ($\boldsymbol{\mu}$PDEs) in a many query context, where the solution has to be computed for many different values of the parameters. The RB method drastically reduces the computational time for any additional solution (during the so-called *online* stage) once an initial set of basis functions has been computed (during the so-called *offline* stage) still retaining a certified level of accuracy. The Greedy algorithm represents the classical sampling strategy to select the parameters values that define the set of basis functions.

In this work we present an alternative and competitive approach, with respect to the classical greedy algorithm, for selecting the parameters values. The

[*] Corresponding author.
Email address: `Stefan.Volkwein@uni-konstanz.de` (Stefan Volkwein)

new approach consists in the introduction of an optimization problem into the parameters sampling procedure that allows to reduce the computational complexity of the offline stage of the RB method. The paper extends the earlier work [21] to distributed parameter functions.

Let us refer to the paper [3], where the authors computes reduced bases for high-dimensional input spaces. In contrast to our approach they do not utilize a-posteriori error estimates in their objectives, but the error between the truth solution and its reduced-basis approximation. Therefore, an efficient offline-online decomposition is not applicable. Moreover, the authors do not study distributed parameter functions. We also mention the recent work [6, 7, 8, 14], where adaptive strategies are suggested for the Greedy-training to overcome the problem with high-dimensional parameter spaces. In the context of the method of proper orthogonal decomposition (POD) nonlinear optimization is utilized in [13] to determine optimal snapshot locations in order to control the number of snapshots and minimize the error in the POD reduced-order model.

The paper is organized in the following manner: the RB method for elliptic equations is briefly reviewed in Section 2. The third section is devoted to the inroduction of our optimization greedy algorithm. In Section 4 we carry out four numerical experiments for the Graetz problem, where we consider physical, geometrical and distributed parameters. Finally, we make some conclusions in Section 5.

2. The reduced basis method for parametric elliptic equations

2.1. The parametric elliptic boundary value problem

Suppose that Ω is an open and bounded domain in $\mathbb{R}^d$, $d \in \{1, 2, 3\}$, with Lipschitz-continuous boundary $\Gamma = \partial\Omega$. Let V and H be real, separable Hilbert spaces defined on the spatial domain Ω. We identify H with its dual space denoted by H'. Furthermore, we suppose that V is dense in H with compact embedding, i.e.

$$V \hookrightarrow H \equiv H' \hookrightarrow V'.$$

By $\langle\cdot\,,\cdot\rangle_H$ and $\langle\cdot\,,\cdot\rangle_V$ we denote the inner products in H and V, respectively. Analogously, we consider two Hilbert spaces V_Γ and H_Γ on the boundary Γ and their duals such that $V_\Gamma \subset H_\Gamma \equiv H'_\Gamma \subset V'_\Gamma$.

Let us focus on linear parametric partial differential equations ($\boldsymbol{\mu}$PDEs) written in operator form:

$$\mathcal{A}(\tilde{y}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathcal{F}(\boldsymbol{\mu}) \text{ in } V', \qquad \mathcal{B}(\tilde{y}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathcal{G}(\boldsymbol{\mu}) \text{ in } V'_\Gamma. \tag{2.1}$$

In (2.1) the symbol $\boldsymbol{\mu}$ represents the set of scalar parameters or parametric functions or both. By $\mathcal{A}(\cdot\,;\boldsymbol{\mu}) : V \to V'_\Omega$ we denote the parametric linear second-order differential operator, by $\mathcal{B}(\cdot\ ;\boldsymbol{\mu}) : V \to V'_\Gamma$ the Dirichlet or Neumann boundary conditions operator, by $\mathcal{F}(\boldsymbol{\mu})$ and by $\mathcal{G}(\boldsymbol{\mu})$ the parametric source term and the boundary data, respectively. We assume that $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_R) \in \mathcal{D}_{\mathsf{ad}}$ holds, where $\mathcal{D}_{\mathsf{ad}}$ is a closed and convex subset of the Hilbert space $\mathcal{D} = \mathcal{D}^1 \times$

$\cdots \times \mathcal{D}^R$ with $\mathcal{D}^i \subset L^2(\Omega)$ or $\mathcal{D}^i \subset \mathbb{R}$. The space $\mathcal{D}$ is endowed with the usual product topology, i.e.,

$$\langle \boldsymbol{\mu}, \tilde{\boldsymbol{\mu}} \rangle_{\mathcal{D}} = \sum_{i=1}^{R} \langle \mu_i, \tilde{\mu}_i \rangle_{\mathcal{D}^i} \quad \text{for } \boldsymbol{\mu} = (\mu_1, \ldots, \mu_R),\ \tilde{\boldsymbol{\mu}} = (\tilde{\mu}_1, \ldots, \tilde{\mu}_R) \in \mathcal{D}$$

and the associated induced norm $\|\boldsymbol{\mu}\|_{\mathcal{D}} = (\langle \boldsymbol{\mu}, \boldsymbol{\mu} \rangle_{\mathcal{D}})^{1/2}$. In order to assure the well-posedness of (2.1) additional problem specific assumptions may be needed.

Remark 2.1. 1) For second-order elliptic equations we often choose the functions spaces $V = H^1(\Omega)$, $H = L^2(\Omega)$, $V_\Gamma = H^{1/2}(\Gamma)$ and $H_\Gamma = L^2(\Gamma)$; see the Graetz problem in Section 4.

2) Notice that — by transforming a parametric spatial domain $\Omega(\boldsymbol{\mu})$ to a certain reference domain $\Omega = \Omega(\hat{\boldsymbol{\mu}})$ with $\hat{\boldsymbol{\mu}} \in \mathcal{D}_{\mathsf{ad}}$ — the variable $\boldsymbol{\mu}$ may contain geometrical parameter as well; see our numerical example presented in Section 4.2. ◊

For any $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$ let us introduce the parameter-dependent bilinear form

$$a(u, v; \boldsymbol{\mu}) = \langle \mathcal{A}(u; \boldsymbol{\mu}), v \rangle_{V',V} \quad \text{for } u, v \in V \text{ and } \boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$$

which satisfies the following conditions.

Assumption 1. *There exist a constants $\beta \geq 0$ and $\alpha > 0$ independent of $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$ such that*

$$\begin{aligned} \big|a(u, v; \boldsymbol{\mu})\big| &\leq \beta \, \|u\|_V \|v\|_V && \textit{for all } u, v \in V \textit{ and } \boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}, \\ a(u, u; \boldsymbol{\mu}) &\geq \alpha \, \|u\|_V^2 && \textit{for all } u \in V \textit{ and } \boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}. \end{aligned}$$

Remark 2.2. We refer to the *successive constraint method* (SCM) for computing the coercivity constant α; see, e.g., [4, 10, 17]. ◊

Under Assumption 1 there is a unique solution by $\tilde{y}(\boldsymbol{\mu})$ to (2.1). Introducing a lift $y_\Gamma(\boldsymbol{\mu}) \in V$ of the Dirichlet boundary condition we may decompose $\tilde{y}(\boldsymbol{\mu})$ as:

$$\tilde{y}(\boldsymbol{\mu}) = y(\boldsymbol{\mu}) + y_\Gamma(\boldsymbol{\mu}),$$

where $y(\boldsymbol{\mu}) \in V^0 = \{v \in V \,|\, \mathcal{B}(v; \boldsymbol{\mu}) = 0 \text{ in } V_\Gamma'\}$.

In the variational form, the problem of finding $y(\boldsymbol{\mu})$ can be recasted in compact form: for $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$ find $y(\boldsymbol{\mu}) \in V^0$ such that

$$a(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in V^0 \tag{2.2}$$

with $f(v; \boldsymbol{\mu}) = \langle \mathcal{F}(\boldsymbol{\mu}) - \mathcal{A}(y_\Gamma(\boldsymbol{\mu}); \boldsymbol{\mu}), v \rangle_{V',V}$ for $v \in V^0$ and $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$. Note that, the formulation (2.2) is general enough to embrace second-order elliptic problem; see, e.g., in [5, 12, 22].

2.2. The finite element Galerkin approximation

The finite element (FE) method to numerically solve problem (2.2) consists in finding an approximate solution $y_{\mathcal{N}}(\boldsymbol{\mu}) \in V_{\mathcal{N}}$, where $V_{\mathcal{N}}$ is a set of subspaces of V^0 with finite dimension $\mathcal{N}$ (typically very large). Therefore, the approximated problem becomes: find $y_{\mathcal{N}}(\boldsymbol{\mu}) \in V_{\mathcal{N}}$ such that

$$a(y_{\mathcal{N}}(\boldsymbol{\mu}), v_{\mathcal{N}}; \boldsymbol{\mu}) = f(v_{\mathcal{N}}; \boldsymbol{\mu}) \quad \forall v_{\mathcal{N}} \in V_{\mathcal{N}}. \tag{2.3}$$

Approximating the resulting problem with the FE method consists in a particular choice for the subspace $V_{\mathcal{N}}$. In the FE method the space $V_{\mathcal{N}}$ is built from piecewise polynomial, linearly independent basis functions $\{\varphi_i^{\mathcal{N}}\}_{i=1}^{\mathcal{N}}$ defined on an admissible triangulation of the spatial domain Ω; see, e.g., [2].

2.3. The reduced basis method

The reduced basis (RB) method efficiently computes an approximation of the FE solution $y_{\mathcal{N}}$ of problem (2.3) by using an approximation space with a significantly smaller dimension than the finite element one, made up of $N \ll \mathcal{N}$ proper solutions of the problem (2.3) corresponding to specific choices of the parameter values. The procedure to select the set of the specific parameter values is not unique and it is addressed in the next sections by involving the original contribute of this work. We consider a set of parameter values properly chosen $S_N = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N\}$ and the space $V_N = \text{span}\,\{\psi^1, \ldots, \psi^N\}$ made up of the corresponding FE solutions $\psi^i = y_{\mathcal{N}}(\boldsymbol{\mu}^i)$ of (2.3). We recall that in order to improve the choice of the basis, a convenient procedure is to orthonormalize the functions through, for instance, the classical Gram-Schmidt orthogonalization procedure; see, e.g., [19]. For every $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$, the reduced basis approximation $y_N(\boldsymbol{\mu})$ is computed through the Galerkin projection onto the reduced basis space $V_N(\boldsymbol{\mu})$ such that:

$$a(y_N(\boldsymbol{\mu}), \psi^n; \boldsymbol{\mu}) = f(\psi^n; \boldsymbol{\mu}), \quad n = 1, \ldots, N. \tag{2.4}$$

Since $y_N(\boldsymbol{\mu})$ belongs to V_N we can decompose it as linear weighted combination of the RB functions, moreover the latter are FE solutions, we can write:

$$y_N = \sum_{m=1}^{N} y_N^m \psi^m = \sum_{m=1}^{N} y_N^m \sum_{j=1}^{\mathcal{N}} \psi_j^m \varphi_j^{\mathcal{N}}.$$

The RB problem (2.4) can be rewritten as:

$$\sum_{m=1}^{N} y_N^m \sum_{j=1}^{\mathcal{N}} \psi_j^m \sum_{i=1}^{\mathcal{N}} \psi_i^n a(\varphi_j, \varphi_i^{;}\boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} \psi_i^n f(\varphi_i^{\mathcal{N}}; \boldsymbol{\mu}) \qquad n = 1, \ldots, N,$$

whence the corresponding linear system can be reformulated with respect to the original FE matrix $\mathbb{A}(\boldsymbol{\mu})$ and source term $\mathbf{f}(\boldsymbol{\mu})$ associated to (2.3) as follows:

$$[\Psi^\top \mathbb{A}(\boldsymbol{\mu}) \Psi]\, \mathbf{y}(\boldsymbol{\mu}) = \Psi^\top \mathbf{f}(\boldsymbol{\mu}), \tag{2.5}$$

with the matrices $\Psi = ((\psi_i^j)) \in \mathbb{R}^{\mathcal{N}\times N}$, $\mathbb{A} = ((a(\varphi_j^{\mathcal{N}}, \varphi_i^{\mathcal{N}}; \boldsymbol{\mu}))) \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$ and with the vectors $\mathbf{f}(\boldsymbol{\mu}) = (f(\varphi_i^{\mathcal{N}}; \boldsymbol{\mu})) \in \mathbb{R}^{\mathcal{N}}$, $\mathbf{y}(\boldsymbol{\mu}) = (y_N^i) \in \mathbb{R}^N$. The "pre" and "post" multiplications involved in (2.5) permit to deal with a reduced size N of the RB system (2.5) that is much smaller than the size $\mathcal{N}$ of the linear system corresponding FE problem (2.3). In general, as rule of thumb, the RB method becomes effective if the resolution time required for a new query of $\boldsymbol{\mu}$ is much smaller than the one required for the solution of the original FE system.

If the problem contains an affine parametric dependence, it is possible to decouple the linear and bilinear forms by the parametric dependence. In case of not affine parametric dependence, the affine decoupling can be recovered through the empirical interpolation method [1]. In the case of a problem involving a parameter distributed function, the affine decomposition is not recoverable and it is suitable to consider it in a part (small) of the domain in order to ensure the effectiveness and the reduction of the model complexity.

2.4. The greedy algorithm

The greedy algorithm is used in the reduced basis method in order to determine the RB space. During this step a (typically small) set of N parameter values is properly selected and, in correspondence of such set, a basis of N solutions of the problem is computed. The *a-posteriori error estimation* is a fundamental ingredient of the greedy selection, it permits an efficient, quick parameter space exploration and a reliable reduced space construction. The greedy algorithm is an efficient technique for the selection of the basis functions $\{\psi^1, \dots, \psi^N\}$ of the RB space V_N (as subset of the FE space $V_{\mathcal{N}}$). We suppose that we have defined the first N basis functions and we look for the value of $\boldsymbol{\mu}^\star$ that defines the next basis function. We distinguish two key computational tasks in the greedy algorithm that are described as follows:

- take the current set of N basis functions $V_N(\boldsymbol{\mu})$ as input and develop the *online dataset* needed to evaluate the RB approximation and the associated error bounds;

- choose a discrete parameter set Ξ_{train} of $\mathcal{D}_{\mathsf{ad}}$ as input and return the parameter $\boldsymbol{\mu}^\star$ which *maximizes* the prediction of the error between the RB solution (by using the previous selected RB bases) and the FE solution. This prediction is represented by the a-posteriori error bound $\Delta_N(\boldsymbol{\mu})$ introduced in Section 2.5. So that: $\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu}\in\Xi_{train}} \Delta_N(\boldsymbol{\mu})$.

Since the evaluation of the error bound is inexpensive, we are usually able to use relatively large training sets and to obtain good exploration of $\mathcal{D}_{\mathsf{ad}}$. Nevertheless, if the problem involves a large number of parameters or if the parameter dimension R is large, we need to choose very large training sets in order to obtain a reasonable exploration of the parameter domain and the error bound sampling tends to be very expensive in terms of computational costs and times. The goal of this work is to avoid this problem by proposing, in the next sections, an alternative approach in the greedy algorithm. Moreover, if the problem involves a parametric function distributed in the domain or in a part of it, the greedy algorithm becomes prohibitive, because the function can be seen as a vector containing the values of the function in the nodes of the mesh and its dimension may be very large. The new approach is efficient even in this case containing distributed parameter functions.

2.5. A-posteriori error estimates

A rigorous error estimation has two main roles in the RB method: to control the error between the approximate RB solution and the FE solution and to drive, during the greedy algorithm, the choice of the $(N+1)$-th basis function once the first N are already available. The calculation of the RB error bound admits an offline/online decomposition. The *offline* stage, performed once, is very expensive and $\mathcal{N}$-dependent, while the *online* evaluation, performed many times for each new desired $\boldsymbol{\mu}$, is very inexpensive and $\mathcal{N}$-independent. This efficient and reliable error estimation permits to predict the RB error with respect to the FE solution without computing the latter. It is crucial during the greedy algorithm to speed up the efficient selection of the snapshots. We recall here the main ingredients of the error estimation theory; see [15]. Let us denote by $y_{\mathcal{N}}(\boldsymbol{\mu})$ the FE solution of the problem (2.3) and by $y_N(\boldsymbol{\mu})$ its RB approximation (i.e., the solution of (2.4)), so that the error $e(\boldsymbol{\mu}) := y_{\mathcal{N}}(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu}) \in V_{\mathcal{N}}$ satisfies

$$a(e(\boldsymbol{\mu}), v_{\mathcal{N}}; \boldsymbol{\mu}) = \mathfrak{r}_N(v_{\mathcal{N}}; \boldsymbol{\mu}) \quad \forall v_{\mathcal{N}} \in V_{\mathcal{N}}, \tag{2.6}$$

where $\mathfrak{r}_N(\cdot\,; \boldsymbol{\mu}) \in V'_{\mathcal{N}}$ is the residual defined as follows

$$\mathfrak{r}_N(v_{\mathcal{N}}; \boldsymbol{\mu}) := f(v_{\mathcal{N}}; \boldsymbol{\mu}) - a(y_N(\boldsymbol{\mu}), v_{\mathcal{N}}; \boldsymbol{\mu}) \quad \text{for } v_{\mathcal{N}} \in V_{\mathcal{N}}.$$

We introduce $r_N(\boldsymbol{\mu}) \in V_{\mathcal{N}}$, the Riesz representation of $\mathfrak{r}_N(\cdot\,; \boldsymbol{\mu})$, satisfying

$$\langle r_N(\boldsymbol{\mu}), v_{\mathcal{N}} \rangle_V = \mathfrak{r}_N(v_{\mathcal{N}}; \boldsymbol{\mu}) \quad \forall v_{\mathcal{N}} \in V_{\mathcal{N}}.$$

This allows us to write the error residual equation (2.6) as

$$\mathcal{A}(e(\boldsymbol{\mu}), v_{\mathcal{N}}; \boldsymbol{\mu}) = \langle r_N(\boldsymbol{\mu}), v_{\mathcal{N}} \rangle_V \quad \forall v_{\mathcal{N}} \in V_{\mathcal{N}},$$

and it follows that the dual norm of the residual can be evaluated through the Riesz representation:

$$\|\mathfrak{r}_N(\cdot\,; \boldsymbol{\mu})\|_{V'_{\mathcal{N}}} := \sup_{v_{\mathcal{N}} \in V_{\mathcal{N}} \setminus \{0\}} \frac{\mathfrak{r}_N(v; \boldsymbol{\mu})}{\|v_{\mathcal{N}}\|_V} = \|r_N(\boldsymbol{\mu})\|_V. \tag{2.7}$$

Utilizing Assumption 1 we obtain the a-posteriori error estimate

$$\|e(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu}) := \frac{1}{\alpha^{1/2}} \|r_N(\boldsymbol{\mu})\|_V. \tag{2.8}$$

3. The greedy optimization algorithm

In this section we recall an alternative approach [21] to the second main task of the greedy algorithm explained in Section 2.5. In particular we want to avoid the error bound sampling over the whole parameter domain and replace this procedure by an optimization problem, in which we want to find the parameter (that plays the role of the optimal control) that minimize a proper cost functional defined through the dual norm of the residual.

To derive first-order necessary optimality conditions we have to suppose the following differentiability assumption.

Assumption 2. *The mapping $\mathcal{D}_{\mathsf{ad}} \ni \boldsymbol{\mu} \mapsto a(u, v; \boldsymbol{\mu})$ is continuously (Fréchet) differentiable for any $u, v \in V$. Its (Fréchet) derivative is denoted by $a_{\boldsymbol{\mu}}(u, v; \boldsymbol{\mu})$ which is a bounded and linear operator. Moreover, $\mathcal{D}_{\mathsf{ad}} \ni \boldsymbol{\mu} \mapsto f(v; \boldsymbol{\mu})$ is continuously (Fréchet) differentiable for any $v \in V$ as well. Its bounded and linear (Fréchet) derivative is denoted by $f_{\boldsymbol{\mu}}(v; \boldsymbol{\mu})$.*

Now, we consider the minimization problem:

$$\min \hat{J}(\boldsymbol{\mu}) \quad \text{subject to} \quad \boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}, \qquad (\hat{\mathbf{P}})$$

where the reduced cost functional is defined as follows:

$$\hat{J}(\boldsymbol{\mu}) = J(y_N(\boldsymbol{\mu}), \boldsymbol{\mu}) = -\frac{1}{2} \left\| a(y_N(\boldsymbol{\mu}), \cdot\,; \boldsymbol{\mu}) - f(\cdot\,; \boldsymbol{\mu}) \right\|^2_{V'_{\mathcal{N}}}. \qquad (3.1)$$

In (3.1) we denote by $y_N(\boldsymbol{\mu})$ the solution to the RB problem (2.4) defined with the already selected basis functions. We note that the first $\boldsymbol{\mu}$ defining the first basis function can be selected randomly and it define the first data set useful to deal with the minimization problem for selecting the second parameter value.

Remark 3.1. 1) To ensure that the objective $\hat{J}$ is continuously differentiable we do not utilize directly the estimator $\Delta_N(\mu)$ from (2.8), but a quadratic upper bound. If $\hat{J}(\boldsymbol{\mu}) \geq -\alpha\varepsilon^2/2$ holds at $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$ for some chosen tolerance $\varepsilon > 0$, then it follows from (2.8)

$$\left\| y_{\mathcal{N}}(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu}) \right\|_V \leq \frac{1}{\sqrt{\alpha}} \left\| f(\cdot\,; \boldsymbol{\mu}) - a(y_N(\boldsymbol{\mu}), \cdot\,; \boldsymbol{\mu}) \right\|_{V'_{\mathcal{N}}} \leq \varepsilon$$

for the error between the corresponding FE and RB solutions $y_{\mathcal{N}}(\boldsymbol{\mu})$ and $y_N(\boldsymbol{\mu})$, respectively. Often ε is chosen from a FE a-priori error analysis for (2.2)

2) Sometimes it is indeed straightforward to compute a lower bound for the coercivity constant α. However, this is not the case in general. Methods like the SCM have been developed to compute a lower bound for the coercivity constant and such a lower bound is normally used in the standard error estimators. In favor of a more general statement we decide to neglect the lower bound in our cost functional. ◊

Thanks to Assumption 2 the reduced cost functional $\hat{J}$ is Fréchet-differentiable. Thus, we can derive first-order necessary optimality conditions in order to characterize (local) optimal solutions to ($\hat{\mathbf{P}}$); see, e.g., [9]. Suppose that $\bar{\boldsymbol{\mu}} \in \mathcal{D}_{\mathsf{ad}}$ is a local solution to ($\hat{\mathbf{P}}$). Let $\bar{y}_N = y_N(\bar{\boldsymbol{\mu}}) \in V_N$ denote the associated optimal state. We introduce the Lagrange functional by

$$\begin{aligned}\mathcal{L}(y_N, \boldsymbol{\mu}, \lambda_N) &= J(y_N, \boldsymbol{\mu}) + \langle a(y_N, \cdot\,; \boldsymbol{\mu}) - f(\cdot\,; \boldsymbol{\mu}), \lambda_N\rangle_{V',V} \\ &= J(y_N, \boldsymbol{\mu}) + a(y_N, \lambda_N; \boldsymbol{\mu}) - f(\lambda_N; \boldsymbol{\mu})\end{aligned}$$

for $(y_N, \boldsymbol{\mu}, \lambda_N) \in V_N \times \mathcal{D}_{\mathsf{ad}} \times V_N$. Since the linearization of (2.4) at $(\bar{y}_N, \bar{\boldsymbol{\mu}})$, given by

$$a(y_N, v_N; \bar{\boldsymbol{\mu}}) = -a_{\boldsymbol{\mu}}(\bar{y}_N, v_N; \bar{\boldsymbol{\mu}})\boldsymbol{\mu} \quad \forall v_N \in V_N,$$

has a unique solution $y_N \in V$, for any $\boldsymbol{\mu} \in \mathcal{D}$ a constraint qualification condition holds; see, e.g., [20]. Thus, we can infer the existence of a (unique) Lagrange multiplier $\bar{\lambda}_N$ satisfying — together with the primal variables $(\bar{y}_N, \bar{\boldsymbol{\mu}})$ — the first-order optimality conditions

$$\begin{aligned}\langle \mathcal{L}_{y_N}(\bar{y}_N, \bar{\boldsymbol{\mu}}, \bar{\lambda}_N), v_N\rangle_{V',V} &= 0 \quad \forall v_N \in V_N, \\ \langle \mathcal{L}_{\lambda_N}(\bar{y}_N, \bar{\boldsymbol{\mu}}, \bar{\lambda}_N), v_N\rangle_{V',V} &= 0 \quad \forall v_N \in V_N, \\ \langle \mathcal{L}_{\boldsymbol{\mu}}(\bar{y}_N, \bar{\boldsymbol{\mu}}, \bar{\lambda}_N), \boldsymbol{\mu} - \bar{\boldsymbol{\mu}}\rangle_{\mathcal{D}',\mathcal{D}} &\geq 0 \quad \forall \boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}.\end{aligned} \tag{3.2}$$

where $\mathcal{L}_{y_N}$, $\mathcal{L}_{\boldsymbol{\mu}}$ and $\mathcal{L}_{\lambda_N}$ denote the Fréchet derivatives with respect to the arguments y_N, $\boldsymbol{\mu}$ and λ_N. For $\mathcal{L}_{y_N}$ we obtain:

$$\begin{aligned}&\langle \mathcal{L}_{y_N}(\bar{y}_N, \bar{\boldsymbol{\mu}}, \bar{\lambda}_N), v_N\rangle_{V',V} \\ &\quad = -\langle a(\bar{y}_N, \cdot\,; \bar{\boldsymbol{\mu}}) - f(\cdot\,; \bar{\boldsymbol{\mu}}), a(v_N, \cdot\,; \bar{\boldsymbol{\mu}})\rangle_{V'_{\mathcal{N}}} + a(v_N, \bar{\lambda}_N\,; \bar{\boldsymbol{\mu}}) \\ &\quad = \langle \bar{\mathfrak{r}}_N(\cdot\,; \bar{\boldsymbol{\mu}}), a(v_N, \cdot\,; \bar{\boldsymbol{\mu}})\rangle_{V'_{\mathcal{N}}} + a(v_N, \bar{\lambda}_N; \bar{\boldsymbol{\mu}}) = a(v_N, \bar{r}_N(\bar{\boldsymbol{\mu}}); \bar{\boldsymbol{\mu}}) + a(v_N, \bar{\lambda}_N; \bar{\boldsymbol{\mu}}),\end{aligned}$$

where $\bar{\mathfrak{r}}_N(\cdot\,; \bar{\boldsymbol{\mu}}) = f(\cdot\,; \bar{\boldsymbol{\mu}}) - a(\bar{y}_N, \cdot\,; \bar{\boldsymbol{\mu}})$ is the residual and $r_N(\bar{\boldsymbol{\mu}})$ denotes its Riesz representation in $V_{\mathcal{N}}$; compare Section 2.5. Thus, the adjoint variable $\bar{\lambda}_N \in V$ solves the *adjoint equation*

$$a(v_N, \bar{\lambda}_N; \bar{\boldsymbol{\mu}}) = -a(v_N, \bar{r}_N(\bar{\boldsymbol{\mu}}); \bar{\boldsymbol{\mu}}) \quad \forall v_N \in V_N.$$

From the second line in (3.2) we obtain:

$$a(\bar{y}_N, v_N; \bar{\boldsymbol{\mu}}) = f(v_N; \bar{\boldsymbol{\mu}}) \quad \forall v_N \in V_N,$$

which is the *state equation*. Finally, from

$$\begin{aligned}&\langle \mathcal{L}_{\boldsymbol{\mu}}(\bar{y}_N, \bar{\boldsymbol{\mu}}, \bar{\lambda}_N), \boldsymbol{\mu}\rangle_{\mathcal{D}',\mathcal{D}} \\ &\quad = -\langle a(\bar{y}_N, \cdot\,; \bar{\boldsymbol{\mu}}) - f(\cdot\,; \bar{\boldsymbol{\mu}}), a_{\boldsymbol{\mu}}(\bar{y}_N, \cdot\,; \bar{\boldsymbol{\mu}})\boldsymbol{\mu} - f_{\boldsymbol{\mu}}(\cdot\,; \bar{\boldsymbol{\mu}})\boldsymbol{\mu}\rangle_{V'_{\mathcal{N}}} \\ &\qquad + \langle a_{\boldsymbol{\mu}}(\bar{y}_N, \bar{\lambda}_N; \bar{\boldsymbol{\mu}}) - f_{\boldsymbol{\mu}}(\bar{\lambda}_N; \bar{\boldsymbol{\mu}}), \boldsymbol{\mu}\rangle_{\mathcal{D}',\mathcal{D}} \\ &\quad = \langle a_{\boldsymbol{\mu}}(\bar{y}_N, \bar{r}_N(\bar{\boldsymbol{\mu}}) + \bar{\lambda}_N; \bar{\boldsymbol{\mu}}) - f_{\boldsymbol{\mu}}(\bar{r}_N(\bar{\boldsymbol{\mu}}) + \bar{\lambda}_N; \bar{\boldsymbol{\mu}}), \boldsymbol{\mu}\rangle_{\mathcal{D}',\mathcal{D}}\end{aligned}$$

for all $\boldsymbol{\mu} \in \mathcal{D}$ and the third line of (3.2), we conclude the *variational inequality*

$$\left\langle a_{\boldsymbol{\mu}}(\bar{y}_N, \bar{r}_N(\bar{\boldsymbol{\mu}}) + \bar{\lambda}_N; \bar{\boldsymbol{\mu}}) - f_{\boldsymbol{\mu}}(\bar{r}_N(\bar{\boldsymbol{\mu}}) + \bar{\lambda}_N; \bar{\boldsymbol{\mu}}), \boldsymbol{\mu} - \bar{\boldsymbol{\mu}} \right\rangle_{\mathcal{D}',\mathcal{D}} \geq 0 \quad \forall \boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}.$$

It is known that the derivative $\hat{J}'$ of the reduced cost functional at a parameter $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$ is given, as in [9], by

$$\begin{aligned} \hat{J}'(\boldsymbol{\mu}) &= \mathcal{L}_{\boldsymbol{\mu}}(y_N, \boldsymbol{\mu}, \lambda_N) \\ &= a_{\boldsymbol{\mu}}(y_N, r_N(\boldsymbol{\mu}) + \lambda_N; \boldsymbol{\mu}) - f_{\boldsymbol{\mu}}(r_N(\boldsymbol{\mu}) + \lambda_N; \boldsymbol{\mu}) \in \mathcal{D}', \end{aligned} \tag{3.3}$$

where y_N solves (2.4) and λ_N is the solution to the adjoint equation utilizing the state y_N and the parameter $\boldsymbol{\mu}$.

Hereinafter, we describe our greedy optimization method. Let suppose that we have defined the first N basis functions. Then the successive value of $\boldsymbol{\mu}$ that defines the $(N+1)$-th snapshot is selected by using the optimization greedy paradigm. As already mentioned, the first computational task, that is the same of the classical greedy algorithm described in Section 2.4, provides the online dataset needed to evaluate with the current space V_N the RB approximation and the associated ingredients to compute the dual norm of the residuals. As second main task, now we apply a variant of the projected gradient method utilizing the derivative (3.3) for the search direction and a line search based on the Armijo rule (see [11, Section 5.4]). Starting with a suitable initial guess $\boldsymbol{\mu}^{(0)}$ we compute the desired parameter value $\boldsymbol{\mu}^{N+1}$ as a (local) approximated solution $\boldsymbol{\mu}^{(k)}$ of $(\hat{\mathbf{P}})$, by using the following stopping criterion:

$$\|\hat{J}'(\boldsymbol{\mu}^{(k)})\|_{\mathcal{D}'} \leq \tau_{abs} + \tau_{rel} \, \|\hat{J}'(\boldsymbol{\mu}^{(0)})\|_{\mathcal{D}'}$$

with $0 < \tau_{abs} \leq \tau_{rel}$. We note that, the cost functional $\hat{J}$ defined in (3.1) has several local minima (specially for large N), so that a good choice of the initial point is fundamental to reach the global minimum parameter value. In order to define a suitable starting value $\boldsymbol{\mu}^{(0)}$, we apply the second main task of the greedy algorithm (see Section 2.4) by using a very coarse training set $\Xi_{train} \subset \mathcal{D}_{\mathsf{ad}}$ and we define the starting value of the gradient projection method by $\boldsymbol{\mu}^{(0)} = \arg\max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_N(\boldsymbol{\mu})$. Due to the small size of Ξ_{train}, this initial sampling is inexpensive and permits to select a suitable initial value of the iterative method. In Algorithm 1 we summarize our method.

4. Numerical experiments for the Graetz problem

The Graetz problem is a classical problem in literature dealing with forced steady heat convection combined with heat conduction in a duct with walls at different temperature. The first segment of the duct has “cold” walls, while the second segment has “hot” walls. The flow has an imposed temperature at the inlet and a known convection field (i.e., a given parabolic velocity profile). In order to explain the detailed strategy, we start presenting the Graetz problem

Algorithm 1 (Greedy optimization method)

```
Require: tolerance $\varepsilon$, (coarse) training set $\Xi_{train} \subset \mathcal{D}_{\mathsf{ad}}$, initial parameter $\boldsymbol{\mu}^1$,
    maximal number $N_{\max}$ of RB functions.
Compute the FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^1)$ of (2.3).
Define the space $V_1 = \mathrm{span}\{y_{\mathcal{N}}(\boldsymbol{\mu}^1)\}$ and set $N = 2$.
repeat
    Compute $\boldsymbol{\mu}^{(0)} = \arg\max_{\boldsymbol{\mu}\in\Xi_{train}} \Delta_{N-1}(\boldsymbol{\mu})$.
    Determine a solution $\bar{\boldsymbol{\mu}}$ to $(\hat{\mathbf{P}})$ by applying the projected gradient method
        with starting value $\boldsymbol{\mu}^{(0)}$ and set $\boldsymbol{\mu}^N = \bar{\boldsymbol{\mu}}$.
    Compute the FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^N)$ of (2.3).
    Define $V_N = V_{N-1} \cup \mathrm{span}\{y_{\mathcal{N}}(\boldsymbol{\mu}^N)\}$ and $N = N + 1$.
until $\Delta_N(\bar{\boldsymbol{\mu}}) < \varepsilon$ or $N = N_{\max} + 1$
```

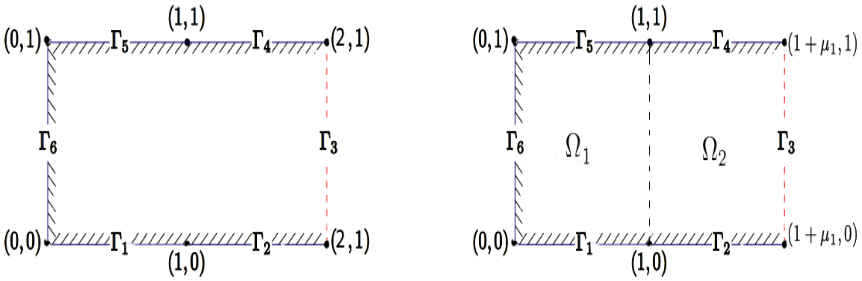

Figure 4.1: Spatial two-dimensional domain Ω for Run 4.1 (left plot) and Run 4.2 (right plot).

dealing with a steady convection-diffusion equation with a single parameter representing the ratio between convection and conduction terms. Then, we increase the complexity of the problem by introducing a second parameter describing the geometrical deformation of the domain. In the last two examples we deal with a parametric distributed functions defined in all or a part of the spatial domain in order to introduce the effectiveness of the new approach for these cases. In all tests, the FE discretization deals with the accuracy of the size $\mathcal{O}(h)$ with mesh size $h \approx 0.0075$.

4.1. Steady convection-diffusion equation with a single parameter

In the first example we deal with a scalar parameter represented by the Peclet number as a measure of axial transport velocity field (modeling the physics of the problem). We consider the physical domain Ω shown in the left plot of Figure 4.1. Here $\boldsymbol{x} = (x_1, x_2)$ denotes a point in Ω. All lengths are non-dimensionalized with respect to a unity length ℓ (dimensional channel width). Moreover, ν is the dimensional diffusivity; and u is a reference dimensional velocity for the convective field (defined as four times the maximum velocity). As already mentioned the scalar real parameter $\boldsymbol{\mu} = \mu$ is given by the Peclet number $\mathrm{Pe} = u\ell/\nu$, representing the ratio between convection and conduction terms and the parameter set is the bounded, closed interval $\mathcal{D}_{\mathsf{ad}} = [0.1, 100] \subset \mathcal{D} = \mathbb{R}$. The

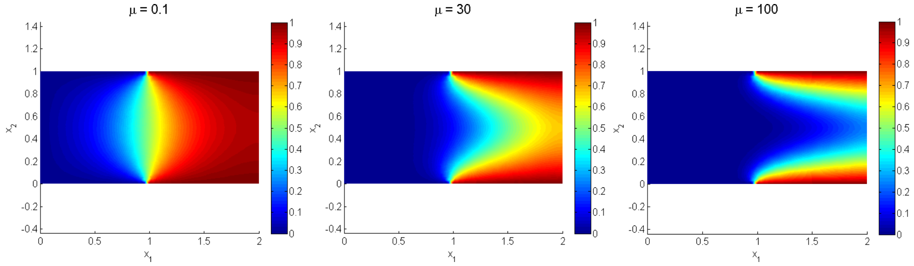

Figure 4.2: FE representative state solutions $y_{\mathcal{N}}(\mu)$ with $\mu = 0.1$ (left plot), $\mu = 30$ (middle plot) and $\mu = 100$ (right plot) for Run 4.1.

governing equations are described by the following steady advection-diffusion problem:

$$\begin{aligned} -\Delta\tilde{y}(\mu) + \mu\boldsymbol{k}\cdot\nabla\tilde{y}(\mu) &= 0 \quad \text{in } \Omega, \\ \tilde{y}(\mu) &= 1 \quad \text{on } \Gamma_2 \cup \Gamma_4, \\ \tilde{y}(\mu) &= 0 \quad \text{on } \Gamma_1 \cup \Gamma_6 \cup \Gamma_5, \\ \frac{\partial\tilde{y}(\mu)}{\partial n} &= 0 \quad \text{on } \Gamma_3, \end{aligned} \tag{4.1}$$

where the dimensional conductivity coefficient for the air flowing in the duct is chosen to be $\boldsymbol{k}(\boldsymbol{x}) = (x_2(1-x_2), 0)^\top$ for $\boldsymbol{x} = (x_1, x_2) \in \Omega$. By introducing the lift function y_Γ for the inhomogeneous Dirichlet boundary conditions, we can define the function $\tilde{y}(\mu) = y(\mu) + y_\Gamma$, where $y(\mu) \in V^0 = \{v \in V \,|\, v = 0 \text{ on } \Gamma_D\}$ with $V = H^1(\Omega)$ and $\Gamma_D = \Gamma \setminus \Gamma_3$. Problem (4.1) can be written as follows:

$$\begin{aligned} -\Delta y(\mu) + \mu\boldsymbol{k}\cdot\nabla y(\mu) &= \Delta y_\Gamma - \mu\boldsymbol{k}\cdot\nabla y_\Gamma \quad && \text{in } \Omega, \\ y(\mu) &= 0 && \text{on } \Gamma_D, \\ \frac{\partial y(\mu)}{\partial n} &= 0 && \text{on } \Gamma_3. \end{aligned} \tag{4.2}$$

The weak formulation of problem (4.2) is given by finding a $y(\mu) \in V^0$ such that

$$a(y(\mu), \varphi; \mu) = f(\varphi; \mu) \qquad \forall \varphi \in V^0,$$

where

$$a(u, v; \mu) = \int_\Omega \nabla u\cdot\nabla v + \mu\boldsymbol{k}\cdot\nabla u v\,\mathrm{d}\boldsymbol{x}, \quad f(v; \mu) = -a(y_\Gamma, v; \mu)$$

for $u, v \in V^0$ and $\mu \in \mathcal{D}_{\mathsf{ad}}$.

Run 4.1. In Figure 4.2 we present some representative FE solutions in correspondence of three parameter values $\mu \in \mathcal{D}_{\mathsf{ad}}$. We consider two different (randomly chosen) coarse training sets Ξ_{train} to find the initial guess $\mu^{(0)}$ for the projected gradient method. As a first parameter value we choose $\mu^1 = 0.1$ to fix the first RB function $\psi^1 = y_{\mathcal{N}}(\mu^1)$; see the left plot of Figure 4.2. In

N	$\mu^{(0)}$	$\bar{\mu}$	μ_g	iter.	$\hat{J}(\mu^{(0)})$	$\hat{J}(\bar{\mu})$	$\hat{J}(\mu_g)$
1	95.32	100.00	99.97	7	-1.34e+00	-1.48e+00	-1.47e+00
2	33.88	42.94	43.02	19	-1.53e−02	-1.65e−02	-1.65e−02
3	31.73	15.19	15.19	24	-8.89e−05	-3.65e−04	-3.65e−04
4	85.71	73.86	73.98	26	-1.28e−05	-2.01e−05	-2.01e−05
5	31.73	26.00	5.74	44	-1.39e−07	-2.12e−07	-4.70e−07
6	85.71	89.37	4.80	21	-3.93e−09	-4.67e−09	-2.48e−08

Table 4.1: Results obtained by using Ξ_{train} of cardinality 6 for Run 4.1. Here, N denotes the number of RB functions already selected for the RB approximation, $\mu^{(0)}$ is the initial guess for the projected gradient method, $\bar{\mu}$ denotes the optimal solution of $(\hat{\mathbf{P}})$ obtained by the projected gradient method and μ_g stands for the parameter value, where $\Delta_N(\boldsymbol{\mu})$ is maximal over a set of 100 equidistant points in $\mathcal{D}_{\mathsf{ad}}$. Moreover, 'iter.' denotes the number of iterations needed by the projected gradient method to converge to $\bar{\mu}$.

N	$\mu^{(0)}$	$\bar{\mu}$	μ_g	iter.	$\hat{J}(\mu^{(0)})$	$\hat{J}(\bar{\mu})$	$\hat{J}(\mu_g)$
1	99.58	100.00	99.97	2	-1.46e+00	-1.48e+00	-1.47e+00
2	43.35	42.96	43.02	2	-1.65e−02	-1.65e−02	-1.65e−02
3	15.14	15.23	15.19	2	-3.65e−04	-3.65e−04	-3.65e−04
4	73.87	73.87	73.98	2	-2.00e−05	-2.01e−05	-2.01e−05
5	8.77	5.63	5.74	10	-3.19e−07	-4.74e−07	-4.74e−07
6	28.69	28.66	28.65	4	-9.45e−09	-9.45e−09	-9.45e−09

Table 4.2: Analogous results as in Table 4.1, but obtained by using Ξ_{train} with cardinality 25 for Run 4.1.

Table 4.1 we report the minimum parameter values ($\bar{\mu}$) obtained through the projected gradient method by using a training set with card $(\Xi_{train}) = 6$, that define the initial guess ($\mu^{(0)}$). For the projected gradient method we utilize the tolerances $\tau_{abs} = 10^{-6}$ and $\tau_{rel} = 10^{-2}$ for the stopping criterion.

In order to compare our approach with the classical greedy sampling, we show, together with the optimal solution $\bar{\mu}$, the parameter value μ_g obtained by maximizing the a-posteriori error $\Delta_N(\mu)$ over an equidistant parameter grid in the interval $\mathcal{D}_{\mathsf{ad}}$ with 100 points. Furthermore, we present also the values of the reduced cost functional at the different parameter values $\mu^{(0)}$, $\bar{\mu}$, and μ_g. It turns out that for $N = 5$ and $N = 6$ the parameter value $\bar{\mu}$ selected by the projected gradient method does not correspond to the selected parameter value μ_g, due to a not suitable initial point, i.e., $\bar{\mu}$ is a local minimum, but not a global one. To overcome this problem we increase the cardinality of Ξ_{train} from 6 to 25. The results are shown in Table 4.2.

We observe that the larger training set avoids the problem to determine local, but not global solutions to $(\hat{\mathbf{P}})$. The projected gradient method is able to recover the global minimum for each step of the greedy algorithm. In Table 4.3 we show the computational time needed for applying the optimization greedy for the two different training sets (T_{opt}) and for applying the classical greedy method (T_g).

N	iter.	$T^{(0)}$	T_{gr}	T_{opt}	iter.	$T^{(0)}$	T_{gr}	T_{opt}	T_g
1	7	0.05	0.08	0.13	2	0.21	0.02	0.24	8.33
2	19	0.05	0.23	0.29	2	0.26	0.02	0.28	9.10
3	24	0.06	0.41	0.47	2	0.25	0.03	0.28	10.10
4	26	0.06	0.39	0.46	2	0.27	0.12	0.40	11.07
5	44	0.07	2.05	2.12	10	0.66	0.30	0.35	11.96
6	21	0.07	0.45	0.53	4	0.32	0.25	0.58	12.76
tot.				4.00				2.13	63.32

Table 4.3: Computational times (in seconds) needed for applying the projected gradient method with two different training sets Ξ_{train} with $\text{card}(\Xi_{train}) = 6$ (left column block) as well as $\text{card}(\Xi_{train}) = 25$ (middle column block) and the classical greedy sampling over a training set with 100 equidistant points for Run 4.1 (right column). Here, $T^{(0)}$ stands for the time needed to compute the starting guess $\mu^{(0)}$ for the projected gradient method, T_{gr} is the time for the optimization and $T_{opt} = T^{(0)} + T_{gr}$.

N	$\text{card}(\Xi_{train})$	$\mu^{(0)}$	$\bar{\mu}$	μ_g	iter.	$T^{(0)}$	T_{gr}	T_{opt}
1	3	77.43	100.00	99.99	31	0.03	0.30	0.34
2	3	73.05	43.03	42.92	88	0.03	1.00	1.04
3	6	19.55	15.28	15.42	11	0.06	0.15	0.21
4	6	69.26	73.91	73.98	15	0.06	0.25	0.31
5	15	2.08	5.64	5.58	13	0.18	0.79	0.97
6	20	30.81	28.70	28.82	8	0.25	0.28	0.54
tot.								3.41

Table 4.4: Results obtained by increasing $\text{card}(\Xi_{train})$ of the training set Ξ_{train} and the corresponding computational times (in seconds) for Run 4.1.

In particular, we report in detail the computational times for finding the initial guess $\mu^{(0)}$ by sampling over the coarse training set and the times needed for the projected gradient method ($T^{(0)} + T_{gr} = T_{opt}$). We note that, by using a larger initial training set, does not always mean to do less gradient iterations, but it helps to find the global minimum instead of a local one. Compared to the classical greedy approach, the CPU times are much smaller for our combined strategy.

Motivated by the results in Table 4.1 and 4.2 a good strategy could be to enrich the initial training set in each step of the greedy algorithm, due to the fact that the number of local minimum increases with the number of basis functions of the reduced space. We present in Table 4.4 results obtained by using different initial train sets Ξ_{train} depending on the step of the greedy algorithm. It turns out that this approach works well and the CPU times are comparable to the ones we presented in Table 4.3.

In Figure 4.3 we show the error between the RB solution and the FE one (minimum, maximum and average over a sample set of 100 values) by using different

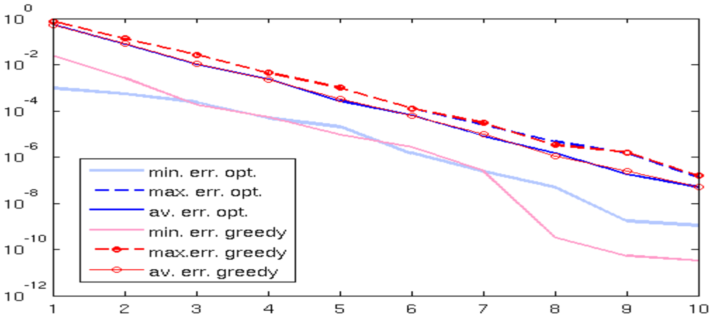

Figure 4.3: Error between the RB solution and the FE one (minimum, maximum and average over a sample set of 100 values) by using different number of basis functions for Run 4.1.

numbers of basis functions for both the classical greedy algorithm and the optimization greedy (with card $(\Xi_{train}) = 6$).
We note that the results are comparable even if the optimization greedy does not capture all the global minimum. Finally, let us mention that the results of the gradient computation has been compared with the MATLAB routine `fmincon.m` by observing that, in all the test cases, they are almost the same. ◊

4.2. Steady convection-diffusion equation with two parameters

In this subsection we consider a geometrical deformation of the domain by introducing a second parameter μ_2 that describes the length of the hot segment of the duct, see in the right plot of Figure 4.1. In this example the parameter domain is $\mathcal{D}_{\mathsf{ad}} = [0.1, 100] \times [1, 10] \subset \mathcal{D} = \mathbb{R}^2$ and we decompose the computational domain into two subdomains Ω_1 and $\Omega_2 = \Omega_2(\mu_2)$, such that $\Omega(\mu_2) = \bar{\Omega}_1 \cup \bar{\Omega}_2$. The governing equations are described by the following steady advection-diffusion problem:

$$\begin{aligned} -\Delta\tilde{y}(\boldsymbol{\mu}) + \mu_1 \boldsymbol{k}\cdot\nabla\tilde{y}(\boldsymbol{\mu}) &= 0 \quad \text{in } \Omega(\mu_2),\\ \tilde{y}(\boldsymbol{\mu}) &= 1 \quad \text{on } \Gamma_2(\mu_2) \cup \Gamma_4(\mu_2),\\ \tilde{y}(\boldsymbol{\mu}) &= 0 \quad \text{on } \Gamma_1 \cup \Gamma_6 \cup \Gamma_5,\\ \frac{\partial\tilde{y}(\boldsymbol{\mu})}{\partial n} &= 0 \quad \text{on } \Gamma_3(\mu_2). \end{aligned} \tag{4.3}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D}_{\mathsf{ad}}$, $\boldsymbol{k}(\boldsymbol{x}) = (x_2(1 - x_2), 0)^\top$ for $\boldsymbol{x} = (x_1, x_2)$. As in Section 4.1 we introduce the lift y_Γ of the Dirichlet boundary conditions and define the function $\tilde{y}$ as $\tilde{y}(\boldsymbol{\mu}) = y(\boldsymbol{\mu}) + y_\Gamma(\boldsymbol{\mu})$, where $y(\boldsymbol{\mu}) \in V^0 = \{v \in V \,|\, v = 0 \text{ on } \Gamma_D(\mu_2)\}$ with $V = H^1(\Omega(\mu_2))$ and $\Gamma_D(\mu_2) = \Gamma(\mu_2) \setminus \Gamma_3(\mu_2)$. Problem (4.1) can be written as follows:

$$\begin{aligned} -\Delta y(\boldsymbol{\mu}) + \mu_1 \boldsymbol{k}\cdot\nabla y(\boldsymbol{\mu}) &= \Delta y_\Gamma(\boldsymbol{\mu}) - \mu_1 \boldsymbol{k}\cdot\nabla y_\Gamma(\boldsymbol{\mu}) && \text{in } \Omega(\mu_2),\\ y(\boldsymbol{\mu}) &= 0 && \text{on } \Gamma_D(\mu_2),\\ \frac{\partial y(\boldsymbol{\mu})}{\partial n} &= 0 && \text{on } \Gamma_3(\mu_2). \end{aligned} \tag{4.4}$$

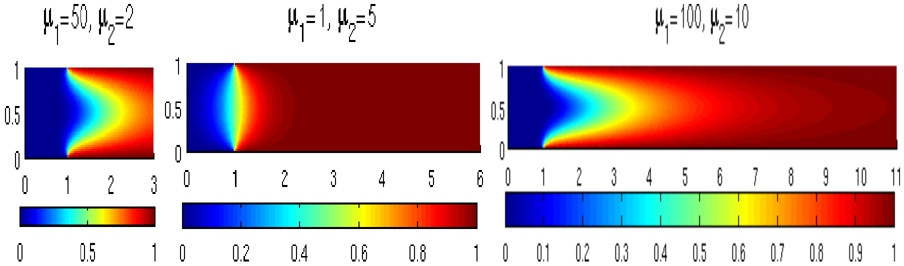

Figure 4.4: FE representative state solutions $y_{\mathcal{N}}(\mu)$ with $\boldsymbol{\mu} = (50, 2)$ (left plot), $\boldsymbol{\mu} = (1, 5)$ (middle plot) and $\boldsymbol{\mu} = (100, 10)$ (right plot) for Run 4.2.

Problem (4.4) may be reformulated on the reference domain $\hat{\Omega} = \bar{\hat{\Omega}}_1 \cup \bar{\hat{\Omega}}_2$, where $\hat{\Omega}_1 = \Omega_1$ holds and the following mapping $T : \hat{\Omega}_2 \to \Omega_2(\mu_2)$ is utilized:

$$T(\hat{\boldsymbol{x}}, \mu_2) = \mathbb{C}(\mu_2)\hat{\boldsymbol{x}} + \mathbf{d}(\mu_2) \text{ with } \mathbb{C}(\mu_2) = \begin{pmatrix} \frac{\mu_2}{\mu_{ref}} & 0 \\ 0 & 1 \end{pmatrix}, \ \mathbf{d}(\mu_2) = \begin{pmatrix} -\frac{\mu_2}{\mu_{ref}} + 1 \\ 0 \end{pmatrix}$$

for $\hat{\boldsymbol{x}} \in \hat{\Omega}_2$, $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$, where $\mu_{ref} = 4.5$ describes the length of the hot segment in the reference domain. Utilizing the mapping T we can define the integrals of the linear and bilinear forms in the reference domain.

Run 4.2. In Figure 4.4 we show representative FE solutions for particular choices of the parameters. We select as first RB function ψ^1 the FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^1)$ for (4.3) with $\boldsymbol{\mu}^1 = (0.1, 1)$. We choose a training set Ξ_{train} of randomly distributed points in $\mathcal{D}_{\mathsf{ad}}$ with $d = \text{card}\,(\Xi_{train}) = 20$. Then we increase the cardinality d of Ξ_{train} every second step. The results are presented in Table 4.5.

As second test, we used an equidistributed initial test samples, starting with $\text{card}\,(\Xi_{train}) = 25$ (if $N \leq 10$) and increasing it until $\text{card}\,(\Xi_{train}) = 49$ (for larger values of N), we report in Table 4.6 the results until $N = 10$.
Since in this example we have two parameters in the problem, we can observe that the value of the cost function is bigger with respect to the example with only one parameter. Due to this typical behavior of the RB method we can do the following consideration. For a larger number of parameter set, the classical greedy algorithm needs a very large parameter samples set in order to find the minimum value. The greedy optimization process is affected by the same problem, but the initial sample set is always much more coarse with respect to the one of the classical greedy algorithm. Moreover, as usual in the reduced basis method, if the number of parameters involved in the problem increase, we need a larger number of reduced functions to find a reliable approximated RB solution. We show in Figure 4.5 the errors between the RB approximation and the FE one, by varying the number of RB functions and by using the three greedy algorithms: the one with bigger equidistribuited samples set (indicated as optimization greedy 1), the one with the very coarse equidistribuited samples set (indicated as optimization greedy 2) and the classical greedy algorithm.

d	N	$\boldsymbol{\mu}^{(0)}$	$\bar{\boldsymbol{\mu}}$	$\boldsymbol{\mu}_g$	it.	$\hat{J}(\boldsymbol{\mu}^{(0)})$	$\hat{J}(\bar{\boldsymbol{\mu}})$	$\hat{J}(\boldsymbol{\mu}_g)$
20	1	(98.2,9.3)	(100,10)	(100,10)	4	-6.0e+0	-6.4e+0	-6.4e+0
20	2	(0.3,9.0)	(0.1,10)	(100,1.4)	7	-3.6e+0	-4.6e+0	-6.1e+0
23	3	(97.1,1.4)	(100,1)	(100,1)	3	-9.4e+0	-1.2e+1	-1.2e+1
23	4	(98.8,4.2)	(100,3.8)	(100,4)	5	-8.8e−1	-9.3e−1	-9.3e−1
27	5	(30.9,9.9)	(38,10)	(37.6,10)	75	-1.4e−1	-1.5e−1	-1.5e−1
27	6	(18.1,1.3)	(31.7,1)	(33.4,1)	22	-3.1e−2	-5.6e−2	-5.6e−2
32	7	(33.6,4.8)	(30.8,3.9)	(29.2,4)	16	-2.0e−2	-2.1e−2	-2.1e−2
32	8	(1.1,3.6)	(0.1,4.3)	(0.1,4.4)	8	-1.1e−2	-1.2e−2	-1.2e−2
37	9	(11.4,9.9)	(14.5,10)	(100,1.8)	10	-4.2e−3	-4.6e−3	-9.2e−3
37	10	(77.4,5.3)	(77.1,5.1)	(100,1.8)	4	-3.6e−3	-3.6e−3	-8.3e−3

Table 4.5: Results obtained by increasing cardinality d of the randomly chosen training set Ξ_{train} for Run 4.2. Here, N denotes the number of RB functions already selected for the RB approximation, $\boldsymbol{\mu}^{(0)}$ is the initial guess for the projected gradient method, $\bar{\boldsymbol{\mu}}$ denotes the optimal solution to $(\hat{\mathbf{P}})$ obtained by the projected gradient method and $\boldsymbol{\mu}_g$ stands for the parameter values, where $\Delta_N(\boldsymbol{\mu})$ is maximal over an equidistant sampling in $\mathcal{D}_{\mathsf{ad}}$ with 625 equidistant points. Moreover, 'it.' denote the number of iterations needed by the projected gradient method to converge to $\bar{\mu}$.

N	$\boldsymbol{\mu}^{(0)}$	$\bar{\boldsymbol{\mu}}$	$\boldsymbol{\mu}_g$	it.	$\hat{J}(\boldsymbol{\mu}^{(0)})$	$\hat{J}(\bar{\boldsymbol{\mu}})$	$\hat{J}(\boldsymbol{\mu}_g)$
1	(100,10)	(100,10)	(100,10)	1	-6.4e+0	-6.4e+0	-6.4e+0
2	(100,1)	(100,1)	(100,1.4)	1	-6.0e+0	-6.0e+0	-6.1e+0
3	(0.1,10)	(0.1,10)	(0.1,10)	1	-2.4e+0	-2.4e+1	-2.4e+1
4	(100,3.3)	(100 3.8)	(100,4)	5	-8.9e−1	-9.3e−1	-9.3e−1
5	(50.1,10)	(38.1,10)	(37.6,10)	104	-1.3e−1	-1.5e−1	-1.5e−1
6	(25.1,1)	(31.7,1)	(22.4,1)	18	-5.2e−2	-5.6e−2	-5.6e−2
7	(25.1,3.3)	(27,3.8)	(29.2,4)	17	-2.0e−2	-2.1e−2	-2.1e−2
8	(0.1,3.3)	(0.1,4.3)	(0.1,4.4)	7	-1.0e−2	-1.2e−2	-1.2e−2
9	(75.0,5.5)	(76,5.1)	(100,1.8)	5	-4.2e−3	-4.4e−3	-8.7e−3
10	(25.1,10)	(14.1,1)	(100,1.8)	26	-1.8e−3	-4.0e−3	-6.2e−3

Table 4.6: Analogous results as in Table 4.5, but obtained with $\mathrm{card}\,(\Xi_{train}) = 25$ of the equidistantly chosen training set Ξ_{train} for Run 4.2.

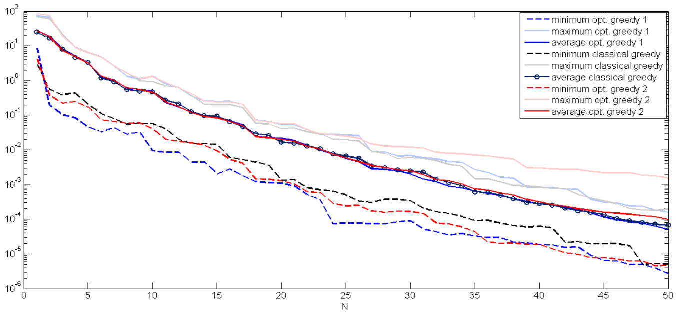

Figure 4.5: Error between the RB approximation and the FE one, by varying the number of basis functions for Run 4.2. We apply three variants of the greedy algorithms: the one with bigger equidistribuited samples set (indicated as optimization greedy 1), the one with the very coarse equidistribuited samples set (indicated as optimization greedy 2) and the classical greedy algorithm.

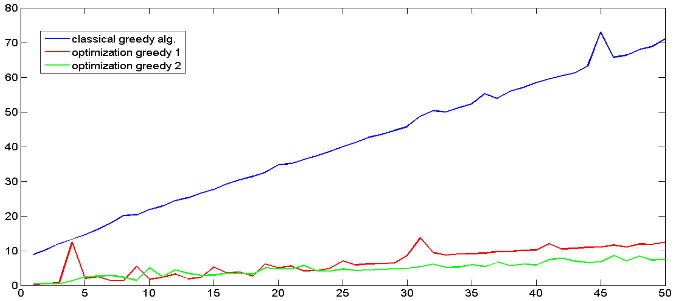

Figure 4.6: Computational times (in seconds) for computing the each step of the greedy algorithms for Run 4.2.

We report the minimum, maximum and averaged error over a train set of 100 parameter values randomly chosen. For the three approaches the averaged error is similar, in particular, we observe for $N > 30$ that the very coarse case presents a larger maximum errors, corresponding to parameters belonging to the region close to the global minimum of the cost functional that is not recovered. Moreover, in order to conclude the comparisons between the different approaches we show in Figure 4.6 the computational times necessary for each step of the optimization greedy by using a very coarse initial samples set (indicated in figure with optimization greedy 1), by increasing the dimension of the initial set (indicated in Figure 4.6 with optimization greedy 2) and finally by using the classical greedy algorithm.

We note that the optimization greedy is much more efficient with respect to the classical greedy, by using the optimization greedy 1 we perform all the 50 greedy steps in ≈ 240 seconds (≈ 4 minutes), while with the classical greedy

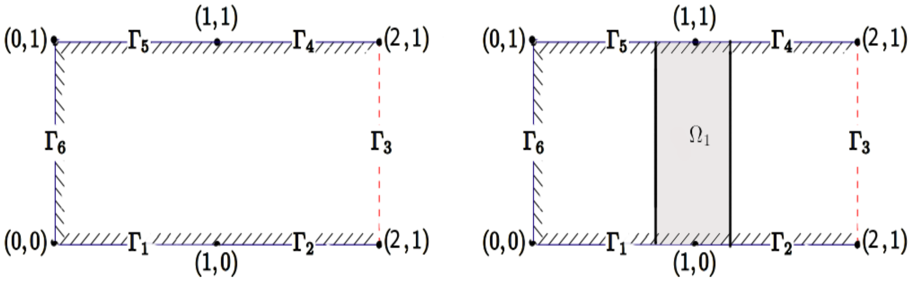

Figure 4.7: Spatial domain Ω for Run 4.3 (left plot) and Run 4.4 (right plot).

algorithm we need ≈ 2036 seconds (≈ 35 minutes) for sampling the parameters space and define 50 basis functions. ♢

4.3. Steady convection-diffusion equation with a single parameter

In this section we want to apply the proposed strategy when the problem involves a distributed parameter function. In this case the classical greedy algorithm is prohibitive because the parameter assumes different values in every node of the computational domain, or in a subdomain, so that it could be represented as a vector of large dimension. We would like to introduce here the case of a parameter function distributed in the whole computational domain, even if it is not suitable for the affine decomposition of the problem and for the final computational scope of the reduced basis method. The scope here is to show how the function $\boldsymbol{\mu}$ is selected by the optimization strategy and how this selection affects the final reduced basis approximation. In general, the presence of a distributed parameter function can be successfully treated in the RB context if it is defined in a relatively small part of the domain, so that the affine decomposition can be applied in the main part of the domain, permitting to define anyway a reduced order method with competitive performance with respect to other full order techniques. We consider again the Graetz problem:

$$\begin{aligned} -\Delta \tilde{y}(\mu) + \boldsymbol{\mu} \cdot \nabla \tilde{y}(\boldsymbol{\mu}) &= 0 \quad \text{in } \Omega, \\ \tilde{y}(\mu) &= 1 \quad \text{on } \Gamma_2 \cup \Gamma_4, \\ \tilde{y}(\mu) &= 0 \quad \text{on } \Gamma_1 \cup \Gamma_6 \cup \Gamma_5, \\ \frac{\partial \tilde{y}(\mu)}{\partial n} &= 0 \quad \text{on } \Gamma_3. \end{aligned} \tag{4.5}$$

where $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}} = \{(\mu_1, \mu_2) : \Omega \to \mathbb{R}^2 \,|\, 1 \leq \mu_i \leq 10 \text{ in } \Omega \text{ and } i = 1, 2\} \subset \mathcal{D} = L^\infty(\Omega; \mathbb{R}^2)$ and the domain is represented in the left plot of Figure 4.7. We follow the same procedure of the last sections, by introducing the lift function y_Γ of the Dirichlet boundary conditions, we can define the function $\tilde{y}(\boldsymbol{\mu}) = y(\boldsymbol{\mu}) + y_\Gamma$, where $y(\mu) \in V^0 = \{v \in V \,|\, v = 0 \text{ on } \Gamma_D\}$ with $V = H^1(\Omega)$ and $\Gamma_D = \Gamma \setminus \Gamma_3$.

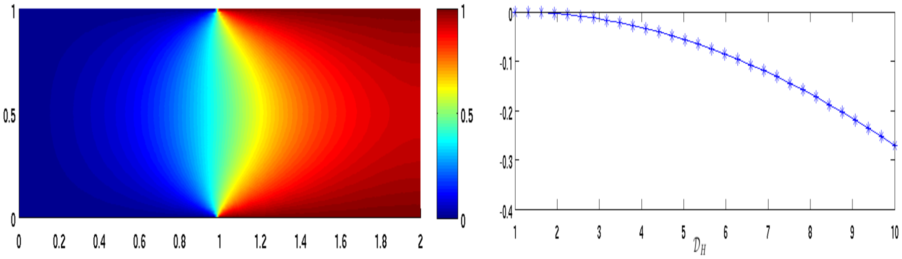

Figure 4.8: First RB function ψ^1 as the FE solution $y_{\mathcal{N}}(\mu^1)$ with $\mu \equiv 1$ (left plot) and reduced cost functional $\hat{J}(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{D}_H$ at the first step of the RB sampling (right plot) for Run 4.3.

The problem (4.5) can be written as follows:

$$\begin{aligned} -\Delta y(\boldsymbol{\mu}) + \boldsymbol{\mu} \cdot \nabla y(\boldsymbol{\mu}) &= \Delta y_\Gamma - \boldsymbol{\mu} \cdot \nabla y_\Gamma && \text{in } \Omega, \\ y(\boldsymbol{\mu}) &= 0 && \text{on } \Gamma_D, \\ \frac{\partial y(\boldsymbol{\mu})}{\partial n} &= 0 && \text{on } \Gamma_3. \end{aligned} \tag{4.6}$$

The weak formulation of problem (4.6) reads as: find $y(\boldsymbol{\mu}) \in V^0$ such that

$$a(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in V^0,$$

where

$$a(u, v; \boldsymbol{\mu}) = \int_\Omega \nabla u \cdot \nabla v + \boldsymbol{\mu} \cdot \nabla u v \, \mathrm{d}\boldsymbol{x}, \quad f(v; \boldsymbol{\mu}) = -a(y_\Gamma, v; \boldsymbol{\mu})$$

for $u, v \in V^0$ and $\boldsymbol{\mu} \in \mathcal{D}_{\mathsf{ad}}$.

Run 4.3. The selection of the initial guess for the iterative algorithm of the optimal control problem is obtained by applying a greedy algorithm in the 'discrete' set $\mathcal{D}_H$ defined as

$$\mathcal{D}_H = \big\{\boldsymbol{\mu} \equiv s \text{ in } \Omega \text{ with } s \in \{s_1, \ldots, s_3\} \subset [1, 10],\ s_i \neq s_j\big\}.$$

Since $\mathcal{N} = 5401$ holds, the classical greedy algorithm is really unfeasible with a parameter vector of such dimension. As first parameter function we choose $\boldsymbol{\mu}^1 \equiv 1$ on Ω and, as first basis function ψ^1 the normalized FE solution of the problem in correspondence of this parameter function, see Figure 4.8. Then we compute the classical greedy algorithm in $\mathcal{D}_H$ in order to select the constant function in $\mathcal{D}_H$ which maximize the error bound and represent the initial value for the gradient method. We show in the right plot of Figure 4.8 the value of the reduced cost functional $\hat{J}(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{D}_H$. The optimal control found by using the optimization algorithm is the same selected by the coarse greedy

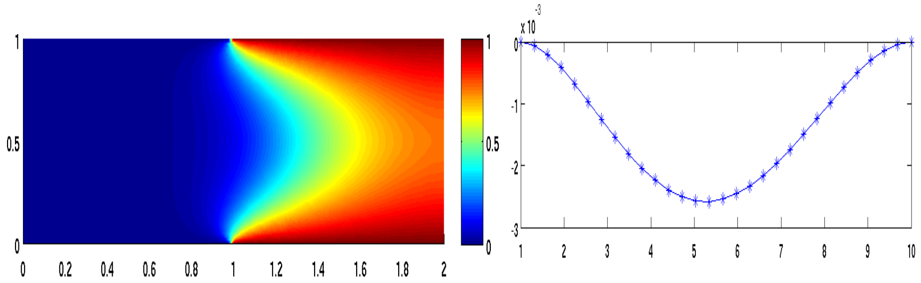

Figure 4.9: Second RB function ψ^2 as the FE solution $y_{\mathcal{N}}(\mu^2)$ with $\mu \equiv 10$ (left plot) and reduced cost functional $\hat{J}(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{D}_H$ at the first step of the RB sampling (right plot) for Run 4.3.

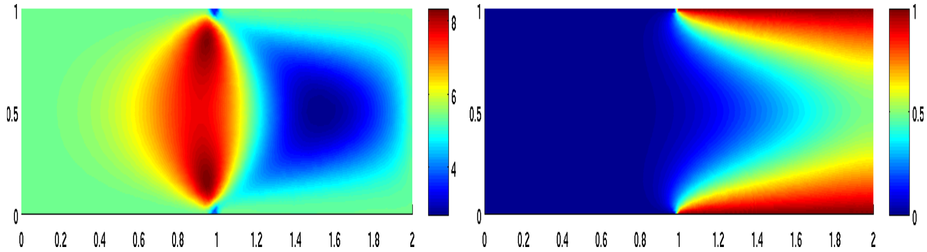

Figure 4.10: Parameter function $\boldsymbol{\mu}^3$ selected by the optimization greedy (left plot) and the corresponding FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^3)$ (right plot) for Run 4.3.

algorithm ($\boldsymbol{\mu} \equiv 10$ in Ω), it means that the minimum of the cost functional in $\mathcal{D}_{\mathsf{ad}}$ coincides with the minimum in $\mathcal{D}_H$. In the left plot of Figure 4.9 we show the corresponding FE solution, which is used to get the second basis function ψ^2. As for the previous selection, we start with a coarse greedy for defining the initial value $\boldsymbol{\mu}^{(0)}$ of the projected gradient method. In the right plot of Figure 4.9 the values of the cost functional for $\boldsymbol{\mu}$ in $\mathcal{D}_H$. The initial value of the gradient method, this time, is the constant function of value about 5.3 and the minimum parameter function found by the optimal control problem is shown in Figure 4.10 together with the correspondent problem solution. Continuing in the same way, we present the parameter function $\boldsymbol{\mu}^4$ and the associated FE solution in Figure 4.11. In order to show the effectiveness of the bases selection performed by the new proposed strategies, we have found the RB approximation of the problem and we have compared the results with the FE approximation by varying the number of basis functions. Figure 4.12 plot the absolute errors between the two approximations (RB and FE) obtained by choosing a set of 100 random parameter functions $\boldsymbol{\mu}$. ◊

4.4. Steady convection-diffusion equation with a parameter function distributed in a part of the computational domain

As last example we consider the Graetz problem involving a distributed parametric function as the previous one, this time defined in a part of the domain,

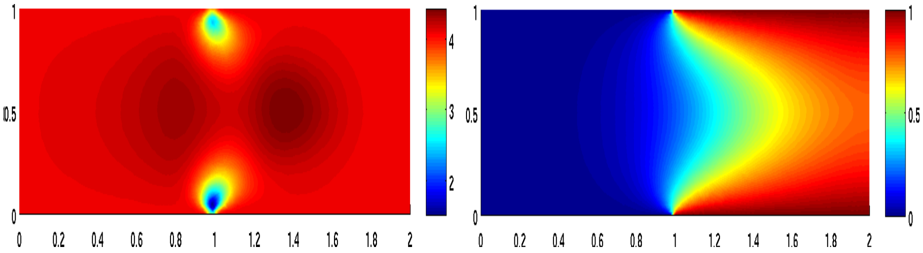

Figure 4.11: Parameter function $\boldsymbol{\mu}^4$ selected by the optimization greedy (left plot) and the corresponding FE solution $y_\mathcal{N}(\boldsymbol{\mu}^4)$ (right plot) for Run 4.3.

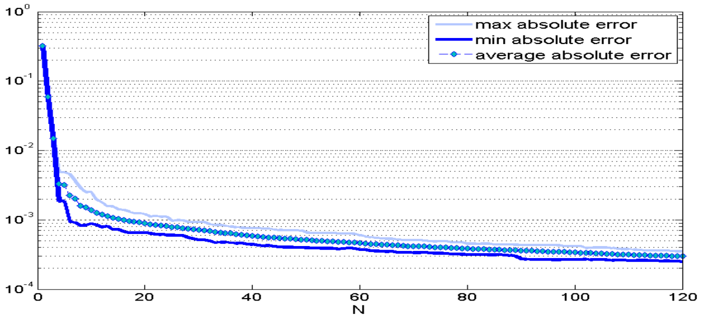

Figure 4.12: Minimum, maximum and averaged errors between the RB and FE approximations by using 100 random parametric functions for Run 4.3.

the central one. The problem is described by the following equations:

$$\begin{aligned} -\Delta\tilde{y}(\boldsymbol{\mu}) + \tilde{y}_{x_1}(\boldsymbol{\mu}) + \tilde{y}_{x_2}(\boldsymbol{\mu}) &= 0 \quad \text{in } \Omega\backslash\Omega_1, \\ -\Delta\tilde{y}(\boldsymbol{\mu}) + \boldsymbol{\mu}\cdot\nabla\tilde{y}(\mu) &= 0 \quad \text{in } \Omega_1, \\ \tilde{y}(\boldsymbol{\mu}) &= 1 \quad \text{on } \Gamma_2\cup\Gamma_4, \\ \tilde{y}(\boldsymbol{\mu}) &= 0 \quad \text{on } \Gamma_1\cup\Gamma_6\cup\Gamma_5, \\ \frac{\partial\tilde{y}(\boldsymbol{\mu})}{\partial n} &= 0 \quad \text{on } \Gamma_3, \end{aligned} \tag{4.7}$$

where $\boldsymbol{\mu}\in\mathcal{D} = \{(\mu_1,\mu_2):\Omega_1\to\mathbb{R}^2\,|\,1\le\mu_i\le 10 \text{ on } \Omega_1 \text{ for } i=1,2\}$. The spatial domain is represented in the right plot of Figure 4.7. The weak formulation of problem (4.7) reads as: find $y(\boldsymbol{\mu})\in V^0$ such that

$$a(y(\boldsymbol{\mu}),v;\boldsymbol{\mu}) = f(v;\boldsymbol{\mu}) \quad \forall v\in V^0,$$

where

$$\begin{aligned} a(u,v;\boldsymbol{\mu}) &= \int_\Omega \nabla u\cdot\nabla v\,\mathrm{d}\boldsymbol{x} + \int_{\Omega\backslash\Omega_1}(u_{x_1}+u_{x_2})v\,\mathrm{d}\boldsymbol{x} + \int_{\Omega_1}\boldsymbol{\mu}\cdot\nabla uv\,\mathrm{d}\boldsymbol{x}, \\ f(v;\boldsymbol{\mu}) &= -a(y_\Gamma,v;\boldsymbol{\mu}) \end{aligned}$$

for $u,v\in V^0$ and $\boldsymbol{\mu}\in\mathcal{D}_{\mathsf{ad}}$.

Run 4.4. As before, the selection of the initial guess for the iterative algorithm of the optimal control problem is obtained by applying a greedy algorithm in the 'discrete' set

$$\mathcal{D}_H = \big\{\boldsymbol{\mu}\equiv s \text{ in } \Omega_1 \text{ with } s\in\{s_1,\ldots,s_{30}\}\subset[1,10],\ s_i\ne s_j\big\}.$$

We consider as first parameter function $\boldsymbol{\mu}^1\equiv 1$ on Ω_1 and, as first basis, the normalized FE solution of the problem corresponding to this parameter function; see left plot in Figure 4.8. As for the previous selection we start with a coarse greedy for defining the initial value of the gradient method and we show in the right plot of Figure 4.8 the values of the cost functional for $\boldsymbol{\mu}$ in $\mathcal{D}_H$. The procedure for the selection of the basis follows the same steps of the last example. In Figures 4.13, 4.14 and 4.15 we present some selected parameter functions and their corresponding FE solutions. In order to show the effectiveness of the bases selection performed by the new proposed strategies, we have found the RB approximation of the problem and we have compared the results with the FE approximation by varying the number of basis functions. Figure 4.16 shows the errors between the two approximations (RB and FE) obtained by choosing a set of 100 random parameter functions. It turns out that the greedy optimization method works fine for this test example with a distributed parameter $\boldsymbol{\mu}$ on the subdomain Ω_1. The selected RB spaces behave as one usually expects in the classical greedy algorithm for scalar parameter (vectors). ◊

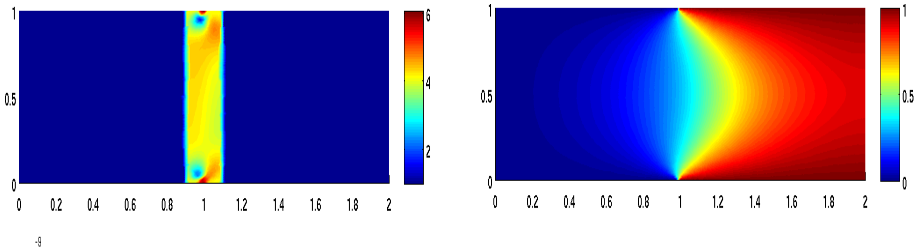

Figure 4.13: Selected parameter function $\boldsymbol{\mu}^8$ and the corresponding FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^8)$ for Run 4.4.

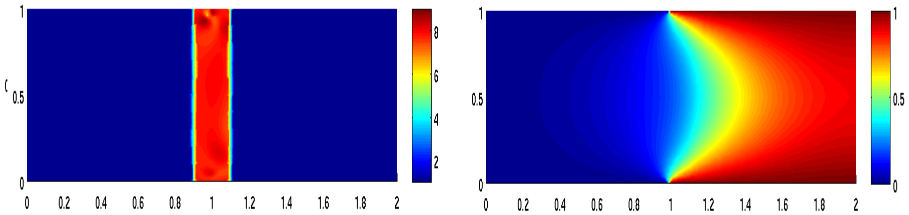

Figure 4.14: Selected parameter function $\boldsymbol{\mu}^{23}$ and the corresponding FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^{23})$ for Run 4.4

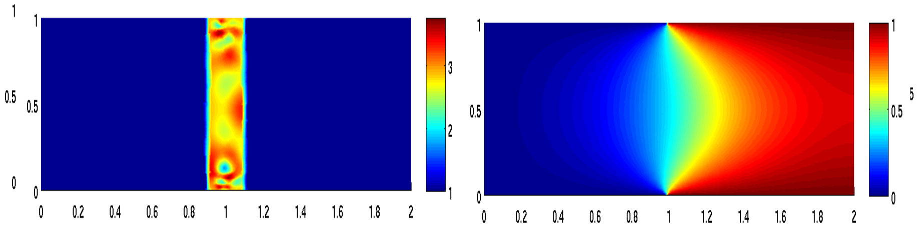

Figure 4.15: Selected parameter function $\boldsymbol{\mu}^{33}$ and the corresponding FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^{33})$ for Run 4.4

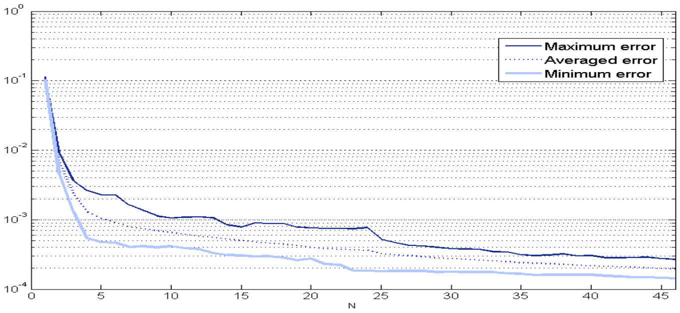

Figure 4.16: Minimum, maximum and averaged errors between the RB and FE approximations by using 100 random parametric functions for Run 4.4.

5. Conclusions

In this paper we have introduced an alternative procedure for selecting the parameter set needed for the RB. It has been successfully applied to $\boldsymbol{\mu}$PDEs in which geometrical and physical features of the computational domain are addressed to a set of parameters and with distributed parameter functions. The critical issue of this procedure is to avoid the selection of a local minimum of the cost functional during the parameters selection. We have proposed a strategies to solve this aspect of the methodology.

The proposed optimization procedure requires, with respect to the greedy algorithm, a smaller computational time during the offline stage of the RB method. Moreover, it is able to capture a better approximation of the best parameter value to select, due to the fact that it plays the role of the optimal control in a minimization problem. Moreover, the proposed approach has been proved to be particularly effective for solving PDEs involving parametric distributed functions ($\boldsymbol{\mu}(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$), for which the classical RB method is prohibitive.

Finally, the proposed strategy can also be applied to the nonlinear problems provided with a rigorous a-posteriori error bound is available for the RB approximation, see [6].

References

[1] 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*, 339(9):667–672, 2004.

[2] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag, 2008.

[3] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM J. Sci. Comput.*, 30:3270-3288, 2008.

[4] Y. Chen, J. Hesthaven, Y. Maday, and J. Rodriguez. A monotonic evaluation of lower bounds for inf-sup stability constants in the frame of reduced basis approximations. *C. R. Acad. Sci. Paris, Ser. I*, 346:1295–1300, 2008.

[5] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North Holland, January 1978.

[6] M. Drohmann, B. Haasdonk, and M. Ohlberger. Adaptive reduced basis methods for nonlinear convection-diffusion equations. Finite Volumes for Complex Applications VI – Problems & Perspectives, J. Fort et al. (eds.), Springer Proceedings in Mathematics 4, Volume 1, page 369-377, 2011

[7] 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*, June 8-13, 2008, Aussois, France, pp. 471-478, 2008.

[8] J.S. Hesthaven, B. Stamm, and S. Zhang. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. Submitted, 2012.

[9] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints.* Springer, 2009.

[10] D.B.P Huynh, G. Rozza, S. Sen, and A.T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Acad. Sci. Paris. Sér. I Math.*, 345:473–478, 2007.

[11] C.T. Kelley. *Iterative Methods for Optimization.* Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1999.

[12] N.V. Krylov. *Lectures on Elliptic and Parabolic Equations in Sobolev Spaces.* American Mathematical Society, September 2008.

[13] O. Lass and S. Volkwein. Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location. Submitted, 2012.

[14] Y. Maday and B. Stamm. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. Submitted to *SIAM J. Sci. Comput.*, 2012.

[15] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations.* Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering. Available at `http://augustine.mit.edu`.

[16] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: reduced-basis output bounds methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

[17] 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. *Arch. Comput. Methods Engrg.*, 15:229–275, 2008.

[18] W. Rudin. *Functional analysis.* McGraw-Hill Science/Engineering/Math, 1991.

[19] L.N. Trefethen and D. Bau III. *Numerical Linear Algebra.* SIAM: Philadelphia, PA, 1997.

[20] F. Tröltzsch. *Optimal Control of Partial Differential Equations. Theory, Methods and Applications.* American Mathematical Society, Providence, 2010.

[21] K. Urban, S. Volkwein, and O. Zeeb. Greedy sampling using nonlinear optimization. Submitted 2012.

[22] V. Volpert. *Elliptic Partial Differential Equations. Volume 1: Fredholm Theory of Elliptic Problems in Unbounded Domains.* Birkhäuser, 2011.

[23] K. Yoshida. *Functional Analysis.* Springer Verlag, 1980.