Optimization strategy for parameter sampling in the reduced basis method

L. Iapichino * **S. Volkwein** *

* *Department of Mathematics and Statistics, University of Konstanz, Universitätsstraβe 10, D-78457 Konstanz, Germany; (E-mail: {Laura.Iapichino, Stefan.Volkwein}@uni-konstanz.de).*

Abstract: The reduced basis (RB) method is an efficient technique to solve parametric partial differential equations in a multi-query context, where the solution has to be computed for many different parameter values. 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 is the classical sampling strategy to select parameter values that define the set of basis functions. Here, an alternative and competitive approach for choosing the parameter values is presented. The new approach is based on an optimization problem for the parameters that allows to reduce the computational complexity of the offline stage of the RB method and improve its effectiveness.

Keywords: PDE-constrained optimization, reduced-order models, reduced basis method, greedy algorithm, parametric distributed function

1. INTRODUCTION

Many physical problems can be described by parametric partial differential equations ($\boldsymbol{\mu}$PDEs) and, because of the usual large dimensionality of the numerical scheme used, there is a growing interest in reducing the computational time required to solve the $\boldsymbol{\mu}$PDEs. The reduced order methods are effective techniques to solve $\boldsymbol{\mu}$PDEs that can replace the classical numerical methods (e.g. finite element method), by reducing the computational time and keeping a suitable level of the solution accuracy.

In this work we deal with the reduced basis method (RB), a reduced order technique particularly suitable in a many query context, where the solution has to be computed for different values of the parameters. The RB solution of the $\boldsymbol{\mu}$PDE for any admissible parameter $\boldsymbol{\mu}$ is computed (during the *online* stage) as Galerkin projection into the RB space defined by a typically small set of solutions of the $\boldsymbol{\mu}$PDE computed during the *offline* stage and corresponding to a small subset of parameter values S_N accurately chosen.

The RB method is usually used for solving $\boldsymbol{\mu}$PDEs with a relative small dimension of the parameter space [10]. This dimension strongly affects the number of needed basis functions composing the RB space and the CPU times required by the so-called *greedy algorithm* used for the selection. In literature, the RB method has been developed for parametric elliptic PDEs [12] and successfully extended to nonlinear parabolic problems; see, e.g., [11]. In this paper we consider an alternative approach for selecting the basis functions, that allows to efficiently deal with large number of parameters and specially with parameters represented by distributed functions defined in the computational domain. The algorithm we use determines the appropriate set of parameters values (or functions) through a PDE-constrained optimization, and for this reason is called *optimization greedy*. Starting by showing its advantages with respect to the classical greedy algorithm even for a simple $\boldsymbol{\mu}$PDE involving only two parameters, we specially underline its efficiency for solving $\boldsymbol{\mu}$PDEs with parametric distributed functions. This paper basically extends the earlier work [15] to distributed parameter functions. Let us refer to [2], where similarly 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, in order to deal with a wider range of problems, even when the error bound is not available. Moreover, the authors do not study distributed parameter functions. We also mention the recent work [3, 4, 5, 9], where adaptive strategies are suggested for the greedy training to overcome the problem with high-dimensional parameter spaces. In the context of proper orthogonal decomposition (POD) nonlinear optimization is utilized in [8] 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 organised as follows. In Section 2 we provide a general description of the class of problems that can be solved with the proposed method. The RB method is introduced in Section 3. In Section 4 we explain the greedy and the optimization greedy algorithms to select the parameter values needed for the RB method. In Section 5 we show the numerical results carried out to prove the effectiveness of the optimization greedy approach. We end in Section 6 with some conclusions.

2. PROBLEM DESCRIPTION

We are interested in considering linear parametric partial differential equations ($\boldsymbol{\mu}$PDEs) in $\Omega(\boldsymbol{\mu})$ of the form:

$$\begin{cases} \mathcal{A}(y(\boldsymbol{\mu}), \boldsymbol{\mu}) = \mathcal{F}(\boldsymbol{\mu}) & \text{in } \Omega(\boldsymbol{\mu}), \\ \mathcal{B}(y(\boldsymbol{\mu}), \boldsymbol{\mu}) = \mathcal{G}(\boldsymbol{\mu}) & \text{on } \partial\Omega(\boldsymbol{\mu}), \end{cases} \tag{1}$$

where $\boldsymbol{\mu}$ is the set of scalar parameters or parametric functions or both, related to either geometry and/or physical properties, $\boldsymbol{\mu} = (\mu_1, \dots, \mu_R) \in \mathcal{D} = \mathcal{D}^1 \times \dots \times \mathcal{D}^R$ with $\mathcal{D}^i \subset \mathbb{R}$ or $\mathcal{D}^i \subset L^2(\Omega)$. We denote by: $y(\boldsymbol{\mu})$ the solution of the problem, $\mathcal{A}(\cdot\,, \boldsymbol{\mu}) : V(\Omega(\boldsymbol{\mu})) \to V'(\Omega(\boldsymbol{\mu}))$ the linear differential operator, $\mathcal{B}(\cdot\,, \boldsymbol{\mu}) : V(\Omega(\boldsymbol{\mu})) \to X(\partial\Omega(\boldsymbol{\mu}))$ represents the Dirichlet or Neumann boundary conditions operator, $\mathcal{F}(\boldsymbol{\mu})$ and $\mathcal{G}(\boldsymbol{\mu})$ the source term and the boundary data, respectively. Here, $V(\Omega(\boldsymbol{\mu}))$ and $X(\partial\Omega(\boldsymbol{\mu}))$ denote appropriate Hilbert and Banach spaces, respectively.
In the variational form, (1) can be recasted in the compact form on a reference domain $\hat{\Omega} = \Omega(\hat{\boldsymbol{\mu}})$ (see, e.g, [7]): find $y(\boldsymbol{\mu}) \in V = V(\hat{\Omega})$ such that

$$a(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \qquad \forall v \in V, \tag{2}$$

The Galerkin method for (2) consists in finding an approximate solution $y_{\mathcal{N}}(\boldsymbol{\mu}) \in V_{\mathcal{N}}$, where $V_{\mathcal{N}}$ is a subspace of V with finite (typically very large) dimension $\mathcal{N}$. Then, the disrete problem becomes: find $y_{\mathcal{N}}(\boldsymbol{\mu}) \in V_{\mathcal{N}}$ such that

$$a(y_{\mathcal{N}}(\boldsymbol{\mu}), v_{\mathcal{N}}; \boldsymbol{\mu}) = f(v_{\mathcal{N}}; \boldsymbol{\mu}) \qquad \forall v_{\mathcal{N}} \in V_{\mathcal{N}}. \tag{3}$$

Utilizing the finite element (FE) method consists in a particular choice for $V_{\mathcal{N}}$. For instance, $V_{\mathcal{N}}$ can be represented by the space of piecewise linear functions defined on a triangulation $\mathcal{T}_h$ of Ω.

3. THE REDUCED BASIS METHOD

The RB method efficiently computes an approximation of the finite element solution $y_{\mathcal{N}}$ of problem (3) by replacing $V_{\mathcal{N}}$ with a smaller space V_N built by N proper solutions of the problem (3) corresponding to specific choices of the parameter values (or functions), we note that $N \ll \mathcal{N}$.
The method is basically split in two main stages:

- *Offline stage*: a set of parameter values (or functions) are properly chosen $S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$ (see Section 4) and the space composed by the corresponding solutions of the problem (3) $V_N = \text{span}\{\psi^1, \psi^2, \dots, \psi^N\}$, with $\psi^i = y_{\mathcal{N}}(\boldsymbol{\mu}^i)$ is built. We note that in order to improve the efficiency of the basis choice, a convenient procedure is to orthonormalize the basis functions through, for instance, the classical Gram-Schmidt orthogonalization procedure [13]. During this step, also the assembling of the method structures (matrices and vectors) can be done.
- *Online stage*: for every $\boldsymbol{\mu} \in \mathcal{D}$, the RB approximation $y_N(\boldsymbol{\mu})$ is computed through the Galerkin projection onto the RB space V_N such that:

$$a(y_N(\boldsymbol{\mu}), \psi^n; \boldsymbol{\mu}) = f(\psi^n; \boldsymbol{\mu}), \qquad n = 1, \dots, N. \tag{4}$$

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]. Due to this decoupling, it is possible to address all the high dimensional computations into the *offline step* and perform the small dimensional (parameter dependent) computations during the *online step*.

4. THE GREEDY ALGORITHM

The algorithm performing the selection of the parameter set, during the offline stage, represents a crucial step for the RB method and affects both the computational time and the accuracy of the method. In this section we first introduce the main ingredients of the *classical greedy algorithm* typically used in the RB method and then explain the alternative optimization approach.

4.1 The classical greedy algorithm

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 can be described as follows:

- taking the current set of N basis functions V_N as input and develop the *online dataset* needed to evaluate the RB approximation and associated error bounds (such as the matrix assembling and the normalization procedure through the projector Π_N onto V_N);
- taking a discrete parameter set Ξ_{train} of $\mathcal{D}$ as input and returns the parameter $\boldsymbol{\mu}^\star$ which *maximizes* the prediction of the error between the RB solution (by using the previous selected RB basis) and the FE solution. This prediction is represented by the posteriori error bound $\Delta_N(\boldsymbol{\mu})$ [12]. So that: $\boldsymbol{\mu}^\star = \arg\max_{\boldsymbol{\mu} \in \Xi} \Delta_N(\boldsymbol{\mu})$.

Since the evaluation of the error bound is inexpensive, we are usually able to use relatively large training set Ξ_{train} and obtain good exploration of $\mathcal{D}$. Nevertheless, if the problem presents a large number of parameters or even a parametric distributed function, this classical approach is prohibitive since we need a 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 times. The goal of this work is to avoid this problem by dealing with an alternative approach that, by involving an optimal control problem, can improve the efficiency of the classical approach.

4.2 The optimization greedy

With the use of the *optimization greedy*, we want to avoid the error bound sampling over a fine discretization of the parameter domain $\mathcal{D}$ and replace this procedure by an optimization problem. More in details, it means to replace the second task of the classical greedy algorithm (seen in the previous section) with the definition of the parameter $\boldsymbol{\mu}^\star$ as the optimal control of a minimization problem involving the cost functional defined through the dual norm of the residual. So that:

$$\hat{J}(\boldsymbol{\mu}^\star) = \inf_{\boldsymbol{\mu} \in \mathcal{D}} \hat{J}(\boldsymbol{\mu}),$$

where the cost functional is defined as follows:

$$\hat{J}(\boldsymbol{\mu}) = -\frac{1}{2} \left\| a(y_N(\boldsymbol{\mu}), \cdot\,; \boldsymbol{\mu}) - f(\cdot\,; \boldsymbol{\mu}) \right\|^2_{V'}. \tag{5}$$

In (5) we denote by $y_N(\boldsymbol{\mu})$ the solution of the RB problem (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 defines the first data set useful to deal with the minimization problem for selecting the second parameter value.
An iterative algorithm can be used to generate a sequence that converges to the optimal parameter and consequently selects a suitable parameters sample. We suppose that we have defined the first N basis functions and we describe the way to select the successive value of $\boldsymbol{\mu}$ that defines the $(N+1)$-th snapshot by using the optimization greedy. As already mentioned, the first computational task, that is the same of the classical greedy algorithm described in Section 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 an iterative algorithm able to define the desired parameter value $\boldsymbol{\mu}^{N+1} = \boldsymbol{\mu}^\star$ as minimum of the cost functional defined in (5) by using its derivative obtained with the Lagrangian approach [6, 14]. Starting with a suitable initial point $\boldsymbol{\mu}^{(0)}$, we use the *projected gradient method* for finding the $\boldsymbol{\mu}^{(k)}$ that converges to the minimum of the cost functional $\hat{J}$ and defines the value of $\boldsymbol{\mu}^\star$. Note that $\hat{J}$ possibly possesses several local minima (specially for large N), so that a good choice of the initial point is fundamental to reach the parameter value corresponding to the global minimum. In order to define a suitable starting value we apply the second main task of the greedy algorithm with a very coarse training set Ξ_{train} and we define the starting value of the gradient method as $\boldsymbol{\mu}^{(0)} = \arg\max_{\boldsymbol{\mu}\in\Xi} \Delta_N(\boldsymbol{\mu})$. 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 general, since the number of local minima increases with the number of basis functions already selected, several strategies can be applied in order to detect the global minimum of $\hat{J}$, e.g., by varying the dimension of the coarse space in dependence of N.

5. NUMERICAL RESULTS

The Graetz problem is a classical problem in the 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 dealing with a steady convection-diffusion equation with two parameters: the first one indicates a geometrical transformation of the computational domain and the second one represents the ratio between convection and conduction terms. Then, we increase the complexity of the problem by dealing with a parametric distributed functions defined in the whole spatial domain in order to introduce the effectiveness of the new approach for this case. In all tests, the FE discretization deals with the accuracy of the size $\mathcal{O}(h)$ with mesh size $h \approx 0.0075$.

5.1 Steady convection-diffusion with two scalar parameters

In the first example we consider two scalar parameters: μ_1 describes the length of the hot segment of the duct (see the plot of Figure 1) and μ_2 represents the Peclet number as a measure of axial transport velocity field (modeling the physics of the problem).

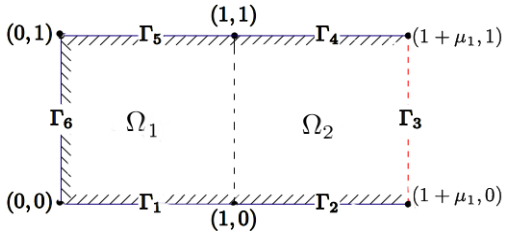

Figure 1. Spatial two-dimensional domain Ω.

All lengths are non-dimensionalized with respect to a unity length ℓ (dimensional channel width). 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, μ_2 is given by the Peclet number $\mathrm{Pe} = u\ell/\nu$, representing the ratio between convection and conduction terms. The parameter domain is $\mathcal{D} = [1, 10] \times [0.1, 100] \subset \mathbb{R}^2$ and we decompose the computational domain into two subdomains Ω_1 and $\Omega_2(\mu_1)$, such that $\Omega(\mu_1) = \Omega_1 \cup \Omega_2(\mu_1)$. The governing equations are described by

$$\begin{aligned} -\Delta\tilde{y}(\boldsymbol{\mu}) + \mu_2 \boldsymbol{k}\cdot\nabla\tilde{y}(\boldsymbol{\mu}) &= 0 \quad \text{in } \Omega(\mu_1),\\ \tilde{y}(\boldsymbol{\mu}) &= 1 \quad \text{on } \Gamma_2(\mu_1)\cup\Gamma_4(\mu_1),\\ \tilde{y}(\boldsymbol{\mu}) &= 0 \quad \text{on } \Gamma_1\cup\Gamma_6\cup\Gamma_5,\\ \frac{\partial\tilde{y}(\boldsymbol{\mu})}{\partial n} &= 0 \quad \text{on } \Gamma_3(\mu_1), \end{aligned} \tag{6}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D}$, $\boldsymbol{k}(\boldsymbol{x}) = (x_2(1-x_2), 0)^\top$ for $\boldsymbol{x} = (x_1, x_2)$ a point in Ω. 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 $y(\boldsymbol{\mu}) \in V^0 = \{v \in V \mid v = 0 \text{ on } \Gamma_D(\mu_1)\}$ with $V = H^1(\Omega(\mu_1))$ and $\Gamma_D(\mu_1) = \Gamma(\mu_1) \setminus \Gamma_3(\mu_1)$. Problem (6) can be written as follows:

$$\begin{aligned} -\Delta y(\boldsymbol{\mu}) + \mu_2 \boldsymbol{k}\cdot\nabla y(\boldsymbol{\mu}) &= f_\Gamma(\boldsymbol{\mu}) \quad \text{in } \Omega(\mu_1),\\ y(\boldsymbol{\mu}) &= 0 \quad \text{on } \Gamma_D(\mu_1),\\ \frac{\partial y(\boldsymbol{\mu})}{\partial n} &= 0 \quad \text{on } \Gamma_3(\mu_1) \end{aligned} \tag{7}$$

with $f_\Gamma(\boldsymbol{\mu}) = \Delta y_\Gamma(\boldsymbol{\mu}) - \mu_2 \boldsymbol{k}\cdot\nabla y_\Gamma(\boldsymbol{\mu})$. Problem (7) is reformulated on the reference domain $\hat{\Omega} = \Omega_1 \cup \hat{\Omega}_2$, where the following mapping $T : \hat{\Omega}_2 \to \Omega_2(\mu_1)$ is utilized:

$$T(\hat{\boldsymbol{x}}, \mu_1) = \mathbb{C}(\mu_1)\hat{\boldsymbol{x}} + \mathbf{d}(\mu_1)$$

$$\text{with } \mathbb{C}(\mu_1) = \begin{pmatrix} \mu_1/\mu_{ref} & 0 \\ 0 & 1 \end{pmatrix},\ \mathbf{d}(\mu_1) = \begin{pmatrix} -\mu_1/\mu_{ref} + 1 \\ 0 \end{pmatrix}$$

for $\hat{\boldsymbol{x}} \in \hat{\Omega}_2$, $\boldsymbol{\mu} \in \mathcal{D}$, where $\mu_{ref} = 4.5$. Utilizing T we define the integrals of the linear and bilinear forms in the reference domain. In Figure 2 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 (6) with $\boldsymbol{\mu}^1 = (1, 0.1)$. We choose a training set Ξ_{train} of randomly distributed points in $\mathcal{D}$ with $d = \mathrm{card}\,(\Xi_{train}) = 20$. Then we increase the cardinality d of Ξ_{train} every second step. Since, in this conditions, the optimization greedy is not able to compute the global minima for a

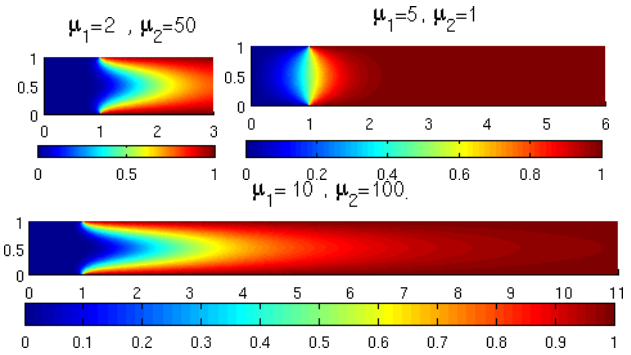

Figure 2. FE representative state solutions $y_{\mathcal{N}}(\mu)$ with $\boldsymbol{\mu} = (2, 50)$ (left plot), $\boldsymbol{\mu} = (5, 1)$ (middle plot) and $\boldsymbol{\mu} = (10, 100)$ (right plot).

large number of N, but only the local ones, as second test, we used a larger equidistributed initial test samples, starting with card $(\Xi_{train}) = 25$ (if $N \leq 10$) and increasing it until card $(\Xi_{train}) = 49$ (for larger values of N). In general for large number of parameter set, the classical greedy needs a very large parameter samples set in order to find the minimum value. The optimization greedy 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 RB method, if the number of parameters involved in the problem increases, we need a larger number of reduced functions to find a reliable approximated RB solution. We show in Figure 3 the errors between the RB approximation and the FE one, by varying the number of RB functions and by using three greedy algorithms: the one with bigger equidistribuited samples set (opt greedy 1), the one with the very coarse equidistribuited samples set (opt greedy 2) and the classical greedy. We report the minimum, maxi-

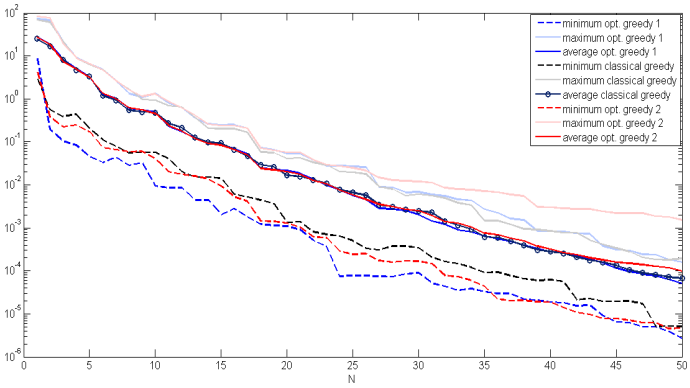

Figure 3. Error between RB and RE approximation by varying the number of basis functions for three variants of the greedy algorithms: the one with bigger equidistribuited samples set (opt greedy 1), the one with the very coarse equidistribuited samples set (opt greedy 2) and the classical greedy.

mum and averaged error over a train set of 100 parameter values randomly chosen. For the three approaches the averaged errors are 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 the computational times necessary for each step of the optimization greedy by using a very coarse initial samples set (optimization greedy 1), by increasing the dimension of the initial set (optimization greedy 2) and finally by using the classical greedy algorithm.

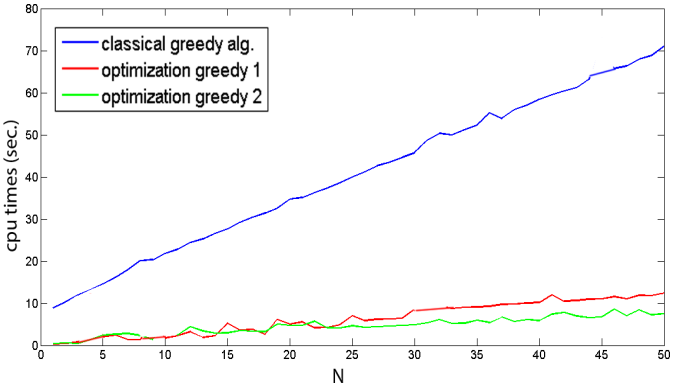

Figure 4. Computational times (in seconds) for computing the each step of the greedy algorithms.

We note that the optimization greedy is much more efficient with respect to the classical greedy even if we increase the dimension of the initial coarse grid. By using the optimization greedy 1 we perform all the 50 greedy steps in $\approx$ 240 seconds ($\approx$ 4 minutes), while with the classical greedy algorithm we need $\approx$ 2036 seconds ($\approx$ 35 minutes) for sampling the parameters space and define 50 basis functions.

5.2 Distributed parameter function

Now 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, 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 complete offline affine decomposition of the problem. The scope here is to show how a set of functions $\boldsymbol{\mu}^i(x)$ is selected by the optimization strategy and how this selection can generate a set of basis functions useful for a final online RB approximation involving every behavior of the parametric functions (possibly affine decomposable, i.e. with fast online computations). In general, the presence of a distributed parameter function can be successfully treated in the RB context with an optimization greedy even if it is not affine, since with this greedy approach the number of heavy computations is relative small (number of iterations of the descendant algorithm), so that the affine decomposition can be applied only online (with the desired parametric function) permitting to define anyway a reduced order method with competitive performance with respect to other full order techniques. We consider again the Graetz problem:

$$
\begin{aligned}
-\Delta\tilde{y}(\mu) + \boldsymbol{\mu}\cdot\nabla\tilde{y}(\boldsymbol{\mu}) &= 0 \quad \text{in } \Omega,\\
\tilde{y}(\mu) &= 1 \quad \text{on } \Gamma_2\cup\Gamma_4,\\
\tilde{y}(\mu) &= 0 \quad \text{on } \Gamma_1\cup\Gamma_6\cup\Gamma_5,\\
\frac{\partial\tilde{y}(\mu)}{\partial n} &= 0 \quad \text{on } \Gamma_3
\end{aligned}
\tag{8}
$$

for $\boldsymbol{\mu} \in \mathcal{D} = \{(\mu_1,\mu_2) : \Omega \to \mathbb{R}^2 \,|\, 0.1 \le \mu_i \le 100 \text{ in } \Omega \text{ for } i = 1,2\} \subset L^\infty(\Omega;\mathbb{R}^2)$. The domain is represented in Figure 5.

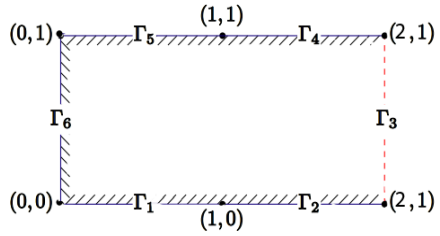

Figure 5. Spatial domain Ω.

We follow the same procedure of Section 5.1. Then, (8) can be expressed in the form (7) with $f_\Gamma(\boldsymbol{\mu}) = \Delta y_\Gamma - \boldsymbol{\mu}\cdot\nabla y_\Gamma$. 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

$$\mathcal{D}_H = \{\boldsymbol{\mu} \equiv s \text{ in } \Omega \,|\, s \in \{s_1,\dots,s_{30}\} \subset [0.1,100],\ s_i \neq s_j\}.$$

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 $\boldsymbol{\mu}^1 \equiv 0.1$ on Ω and the first basis function ψ^1 is the normalized FE solution of the problem in correspondence of this parameter function. Then we compute the classical greedy algorithm in the very coarse set $\mathcal{D}_H$ in order to select the constant function in $\mathcal{D}_H$ which maximizes the error bound and represents the initial value for the gradient method.
The optimal control found by using the optimization algorithm at the first step is the same selected by the coarse greedy algorithm ($\boldsymbol{\mu} \equiv 100$ in Ω), it means that the minimum of the cost functional in $\mathcal{D}_{\mathsf{ad}}$ coincides with the minimum in $\mathcal{D}_H$.
For selecting the third basis function, we start again with a coarse greedy for defining the initial value $\boldsymbol{\mu}^{(0)}$ of the projected gradient method. The initial value of the gradient method, this time, is the constant function of value about 32 and the minimum parameter function found by the optimal control problem is shown in Figure 6 together with the correspondent FE solution of the problem.

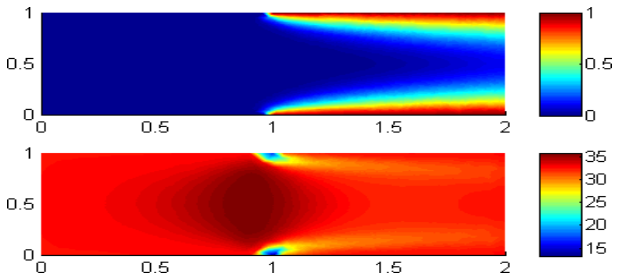

Figure 6. Parameter function $\boldsymbol{\mu}^3$ selected by the optimization greedy (down plot) and the corresponding FE solution $y_{\mathcal{N}}(\boldsymbol{\mu}^3)$ (up plot).

Continuing in the same way, we show in Figure 7 some of the first parameter functions selected by the optimization greedy and needed for composing the RB space.

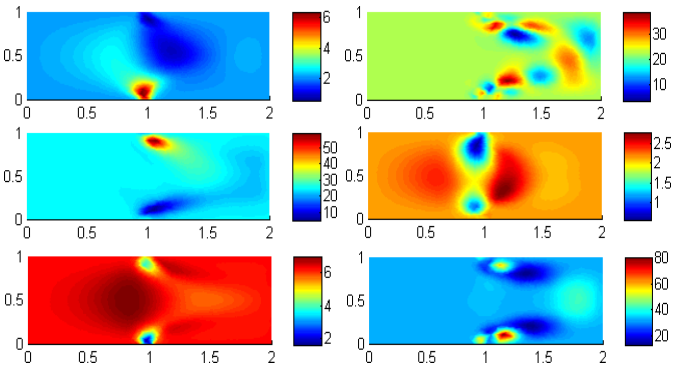

Figure 7. Some parameter functions selected by the optimization greedy.

In order to compare this approach with a possible classical greedy one, we consider the function $\boldsymbol{\mu}(x)$ piecewise constant having eight possible values in eight subdomains of Ω (see the first plot of Figure 8), such that it can be represented by a vector $\boldsymbol{\mu} = (\mu_1,\dots,\mu_8)$ and the classical greedy approach can be performed and a second RB space can be computed.
We have considered, then, four sets of possible parameter functions $\boldsymbol{\mu}(x)$ usable online to test the RB spaces obtained with the two different greedy algorithms. The first set is composed by constant functions in $\boldsymbol{\mu}(x)$ in Ω. The other three sets are piecewise constant functions defined in different subdivisions of Ω (see some example in the second, third and fourth plot of Figure 8 where we deal with 2, 40 and 5 subdomains).

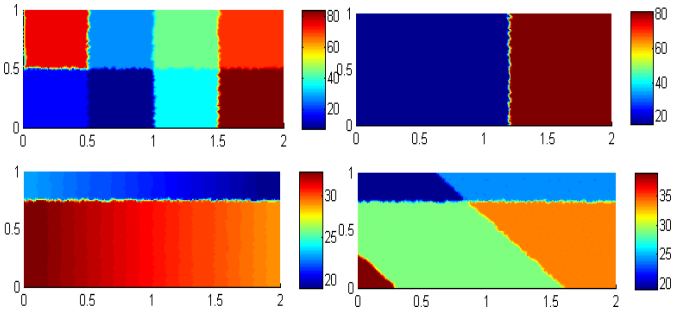

Figure 8. Piecewise constant parametric functions used for the classical greedy algorithm (first plot) and for some online tests (second, third and fourth plot).

In order to show the effectiveness of the basis selection performed by the optimization greedy, for the explained sets of online parametric functions we have found the RB approximation of the problem by using both RB spaces (the one built by the optimization greedy (OG) and the one by the classical greedy (CG)). We have compared the results obtained with the RB method, by varying the number N of basis functions, with the ones obtained with the FE method. Figure 9 plots the absolute errors between the two approximations (RB and FE) obtained by choosing a set of 10 random piecewise constant parameter functions $\boldsymbol{\mu}$ defined in the different decompositions of the domain Ω earlier mentioned.

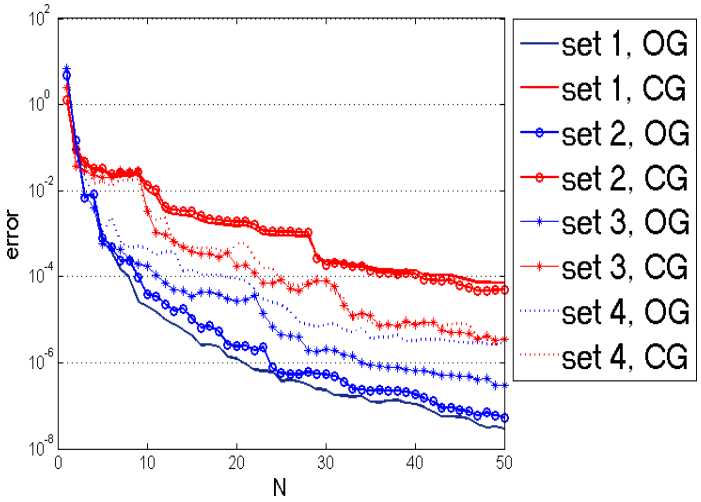

Figure 9. Averaged errors between the RB solutions (obtained with classical greedy (CG) and optimization greedy (OG)) and FE approximations by using 10 random functions of 4 set of piecewise constant parametric functions.

We note that the gap between the results can be very large (4 orders of magnitude) or small depending on the function $\boldsymbol{\mu}(x)$ considered online in the problem. In general the optimization greedy is able to select always a better space of basis functions with respect to the classical greedy algorithm, due to the distributed parameter functions used offline.

6. CONCLUSIONS

In this paper we have introduced an alternative procedure for selecting the parameter set needed for the RB method, particularly effective when the problem involves both large number of parameters and distributed parameter functions. 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 or distributed parameter functions. The proposed optimization procedure requires, with respect to the classical 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. The considered 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. This approach improves the efficiency of the method, by searching the parameter values of S_N in the continuous parameter space $\mathcal{D}$ and not in its discretization Ξ_{train} and it is able to perform a better selection of the parameters sample. 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 [3].

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] 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.

[3] 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

[4] 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.

[5] 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.

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

[7] L. Iapichino. *Reduced basis methods for the solution of parametrized PDEs in repetitive and complex networks with application to CFD*. Ph.D. thesis, N. 5529, École Polytechnique Fédérale de Lausanne, 2012.

[8] O. Lass and S. Volkwein. Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location. *Computational Optimization and Applications*. 58:645–677, 2014.

[9] Y. Maday and B. Stamm. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM J. Sci. Comput.*, 35(6): A2417–A2441, 2013.

[10] 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`.

[11] A.T. Patera and K. Veroy. *Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds*. International Journal for Numerical Methods in Fluids. 47 (8-9):773-788,2005.

[12] 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.

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

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

[15] K. Urban, S. Volkwein, and O. Zeeb. Greedy sampling using nonlinear optimization. *Reduced Order Methods for Modeling and Computational Reduction MS&A - Modeling, Simulation and Applications*. volume 9, pp 137–157,2014.