MOX–Report No. 40/2013

Stabilized reduced basis method for parametrized advection-diffusion PDEs

Pacciarini P.; Rozza G.

MOX, Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

Stabilized reduced basis method for parametrized advection-diffusion PDEs

Paolo Pacciarini[a], Gianluigi Rozza[b,c,*]

[a] *MOX - Modellistica e Calcolo Scientifico, Dipartimento di Matematica F. Brioschi, Politecnico di Milano, P.za Leonardo da Vinci 32, I-20133, Milano, Italy*
[b] *SISSA mathLab, International School for Advanced Studies, Via Bonomea 265, I-34136 Trieste, Italy*
[c] *CMCS - Modelling and Scientific Computing, MATHICSE - Mathematics Institute of Computational Science and Engineering, EPFL - Ecole Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland*

Abstract

In this work, we propose viable and efficient strategies for the stabilization of the reduced basis approximation of an advection dominated problem. In particular, we investigate the combination of a classic stabilization method (SUPG) with the Offline-Online structure of the RB method. We explain why the stabilization is needed in both stages and we identify, analytically and numerically, which are the drawbacks of a stabilization performed only during the construction of the reduced basis (i.e. only in the Offline stage). We carry out numerical tests to assess the performances of the "double" stabilization both in steady and unsteady problems, also related to heat transfer phenomena.

Keywords: reduced basis, advection dominated problems, stabilization methods.

1. Introduction

The aim of this work is to study and develop a *stabilized reduced basis method* suitable for the approximation of the solution of parametrized advection-diffusion PDEs with high Péclet number, that is, roughly, the ratio between the advection term and the diffusion one.

Advection-diffusion equations are very important in many engineering applications, because they are used to model, for example, heat transfer phenomena (with conduction and convection) [20] or the diffusion of pollutants in the atmosphere or in the water [7, 37]. In such applications, we often need very fast evaluations of the approximated solution, depending on some physical and/or geometrical input parameters. This happens, for example, in the case of *real-time* simulations. Moreover, we need rapid evaluations also if we have to perform repeated approximation of the solution, for different input parameters. An important case of this *many-query* situation is represented by some optimization problems, in which the objective function to optimize depends on the parameters through the solution of a PDE.

The reduced basis (RB) method [36, 40] meets our need for rapidity and it is also able to guarantee the *reliability* of the solution, thanks to sharp *a posteriori* error bounds. A crucial feature of the RB method is its decomposition into two computational steps. During the first expensive one, called *Offline* step, some high-fidelity approximated solutions are computed, which will become the global basis functions of the Galerkin projections performed during the second inexpensive phase, called *Online* step. A brief introduction to the RB method will be given in Section 2.

*Corresponding author
Email addresses: paolo.pacciarini@polimi.it (Paolo Pacciarini), grozza@sissa.it (Gianluigi Rozza)

As the advection-diffusion equations are often used to model heat transfer phenomena, we can find in literature many results about the RB approximation of heat transfer problems such as the Poiseuille-Graetz problem or the "thermal fin" problem [11, 30, 36, 39, 41, 42]. However, until now, only the case in which the Péclet number is reasonably low (i.e. $\sim 10^2$) was considered.

When the Péclet number is higher (i.e. $\sim 10^5$), it is very well known [38] that the Finite Element (FE) solution of the advection-diffusion equation - that the RB method aims to recover - can show significant instability phenomena. In order to fix this problem, in the RB framework, some solutions have been proposed for the steady case [7, 8, 35, 37]. The basic idea is to consider as *truth* solution a stabilized FE one, using some classical stabilization method (e.g. the SUPG method [38]), and then to perform the RB *Offline* and *Online* steps using the stabilized bilinear form instead of the original one. In the cited papers we can find some applications to environmental sciences and engineering problems concerning, in particular, air pollution. Very recently, also a Petrov-Galerkin based strategy has been proposed to deal with high Péclet number problems [6].

In our work we want to go further in the study of the *stabilized RB method*, proposing viable and efficient strategies to be used combined with the Offline-Online computational procedures and providing a deeper analysis on the need of stabilization for parametrized advection-diffusion problems. We start by studying steady problems and then we move to the time dependent case.

After having done, in Section 2, a short presentation of the RB method, in Section 3 we observe and analyse what happens when we "stabilize" only the Offline stage of the RB method, thus producing "stable" basis function to be interpolated in the Online stage by projecting with respect to the non-stabilized advection-diffusion operator. We will show that the latter strategy is not satisfactory because of "inconsistency" problems between the Offline and Online stages, arising from the use of two different bilinear forms. We will also prove an *a priori* error estimate (Proposition 3.1) which will allow us to quantify this inconsistency. After having determined which stabilization strategy gives better results and why, in Section 4 we will try to apply it to a test problem with a parameter dependent internal layer, using also a piecewise quadratic polynomial *truth* approximation space. Finally, in Section 5 we extend the investigation of the RB stabilization method to parabolic problems.

2. A brief review of the reduced basis method

The reduced basis (RB) method is a reduced order modelling (ROM) technique which provides rapid and reliable solutions for parametrized partial differential equations (PPDEs), in which the parameters can be either physical or geometrical [36, 40].

The need to solve this kind of problems arises in many engineering applications, in which the evaluation of some *output* quantities is required. These *outputs* are often function of the solution of a PDE, which can in turn depend on some *input* parameters. The aim of the RB method is to provide a very fast computation of this *input-output* evaluation.

Roughly speaking, given a value of the parameter, the (Lagrange) RB method consists in a Galerkin projection of the continuous solution on a particular subspace of a high-fidelity approximation space, e.g. a finite element (FE) space with a large number of degrees of freedom. This subspace is the one spanned by some pre-computed high-fidelity global solutions (*snapshots*) of the continuous parametrized problem, corresponding to some properly chosen values of the parameter.

For a complete presentation of the reduced basis method we refer to [36, 40], now we just recall its main features and we introduce some notations.

2.1. The continuous problem

Let $\boldsymbol{\mu}$ belong to the *parameter domain* $\mathcal{D}$, a subset of $\mathbb{R}^P$. Let Ω be a regular bounded open subset of $\mathbb{R}^d$ ($d = 1, 2, 3$) and X a suitable Hilbert space. Given a parameter value $\boldsymbol{\mu} \in \mathcal{D}$, let $a(\cdot, \cdot; \boldsymbol{\mu}) \colon X \times X \to \mathbb{R}$ be a bilinear form and let $F(\cdot; \boldsymbol{\mu}) \colon V \to \mathbb{R}$ be a linear functional. As we will focus on advection-diffusion equations, that are second order elliptic PDE, the space X will be such that $H^1_0(\Omega) \subset X \subset H^1(\Omega)$. Formally, our problem can be written as follows:

$$\begin{aligned}&\text{find } u(\boldsymbol{\mu}) \in X \text{ s.t.}\\ &a(u(\boldsymbol{\mu}), v; \mu) = F(v; \boldsymbol{\mu}) \quad \forall v \in X.\end{aligned} \tag{1}$$

The coercivity and continuity assumption on the form a can now be expressed by, respectively:

$$\exists\, \alpha_0 > 0 \quad \text{s.t.} \quad \alpha_0 \le \alpha(\boldsymbol{\mu}) = \inf_{v \in X} \frac{a(v, v; \boldsymbol{\mu})}{\|v\|^2_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{2}$$

and

$$+\infty > \gamma(\boldsymbol{\mu}) = \sup_{v \in X} \sup_{w \in X} \frac{|a(v, w; \boldsymbol{\mu})|}{\|v\|_X \|w\|_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{3}$$

We shall make now an important assumption: the *affine* dependency of a on the parameter $\boldsymbol{\mu}$. With *affine*, we mean that the form can be written in the following way:

$$a(v, w; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta^q_a(\boldsymbol{\mu}) a^q(v, w) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{4}$$

Here, $\Theta^q_a \colon \mathcal{D} \to \mathbb{R}$, $q = 1, \ldots, Q_a$, are smooth functions, while $a^q \colon X \times X \to \mathbb{R}$, $q = 1, \ldots, Q_a$, are $\boldsymbol{\mu}$-independent continuous bilinear forms. This assumption turns out to be crucial for performing the Offline-Online decoupling of the computation [36, 40]. At last we assume that also the functional F depends "affinely" on the parameter:

$$F(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_F} \Theta^q_F(\boldsymbol{\mu}) F^q(v) \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{5}$$

where, also in this case, $\Theta^q_F \colon \mathcal{D} \to \mathbb{R}$, $q = 1, \ldots, Q_F$, are smooth functions, while $F^q \colon X \to \mathbb{R}$, $q = 1, \ldots, Q_a$, are continuous linear functionals.

Recalling that $X^{\mathcal{N}}$ is a conforming finite element space with $\mathcal{N}$ degrees of freedom, we can now set the *truth* approximation of the problem (1):

$$\begin{aligned}&\text{find } u^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.}\\ &a(u^{\mathcal{N}}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = F(v^{\mathcal{N}}; \boldsymbol{\mu}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.\end{aligned} \tag{6}$$

As we are considering the conforming FE case, conditions similar to (2) and (3) are fulfilled by restriction. More precisely, as regards the coercivity of the restriction of a to $X^{\mathcal{N}} \times X^{\mathcal{N}}$, we define:

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) := \inf_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{a(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu})}{\|v\|^2_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{7}$$

and, as we are considering a restriction, it easily follows that:

$$\alpha(\boldsymbol{\mu}) \le \alpha^{\mathcal{N}}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{8}$$

Similarly, for the continuity, we can define

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) := \sup_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \sup_{w^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{|a(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu})|}{\|v^{\mathcal{N}}\|_X \|w^{\mathcal{N}}\|_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{9}$$

and it holds that:

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) \le \gamma(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{10}$$

In this work we will consider as *truth* approximation space $X^{\mathcal{N}}$ a classical finite element space [34].

As we have already said, the parameter $\boldsymbol{\mu}$ on which the equation depends can be *geometrical*, i.e. the domain of the equation depends on some parameters [19, 24, 26, 27, 28]. This means that, given $\boldsymbol{\mu} \in \mathcal{D}$, our problem can be expressed in the following way (that we call *original problem*):

$$\begin{aligned} &\text{find } u_o(\boldsymbol{\mu}) \in X_o(\boldsymbol{\mu}) \text{ s.t.} \\ &a_o(u_o(\boldsymbol{\mu}), v_o; \boldsymbol{\mu}) = F_o(v_o; \boldsymbol{\mu}) \quad \forall v_o \in X_o(\boldsymbol{\mu}) \end{aligned} \tag{11}$$

where $X_o(\boldsymbol{\mu})$ is a functional space on the *original domain* $\Omega_o(\boldsymbol{\mu})$, satisfying the condition $H_0^1(\Omega_o(\boldsymbol{\mu})) \subset X_o(\boldsymbol{\mu}) \subset H^1(\Omega_o(\boldsymbol{\mu}))$. Moreover $a_o(\cdot, \cdot; \boldsymbol{\mu})$ and $F_o(\cdot; \boldsymbol{\mu})$ are a bilinear and a linear form, respectively, on $X_o(\boldsymbol{\mu})$. We assume that the bilinear form a_o satisfies conditions (2) and (3).

In order to effectively apply the RB method to the problem (11), we need to map the parametric domain onto a reference one denoted with Ω, via a suitable parameter-dependent transformation T that is:

$$T(\cdot; \boldsymbol{\mu}) \colon \Omega \to \Omega_o(\boldsymbol{\mu}). \tag{12}$$

The reference domain can be defined by choosing the original domain corresponding to a particular value of the parameter. In this work we used only affine mappings [36, 40] that allow to easily recover the affinity assumptions (4) and (5). In [36, 40] it is possible to find, in particular, a detailed treatment of the advection-diffusion operators. For the sake of completeness, we have to recall that it would also be possible to use non-affine transformations (e.g. free-form deformation [24, 27], radial basis functions [26], transfinite maps [19, 25]) to describe parametrically the domain. Note that these approaches require the use of some interpolation techniques (e.g. empirical interpolation [1, 9, 12, 24]), in order to recover the affinity assumptions (4) and (5).

2.2. The reduced basis method: main features

Let us suppose that we are given a problem in the form (1) and its *truth* approximation (6). We recall that the dimension of the finite element space $X^{\mathcal{N}}$ is $\mathcal{N}$. We introduce now a set of N suitably chosen parameter values:

$$S = \{\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N\} \tag{13}$$

and we can define the *reduced basis space*:

$$X_N^{\mathcal{N}} = \text{span}\{u^{\mathcal{N}}(\boldsymbol{\mu}^n) \,|\, 1 \le n \le N\}^1. \tag{14}$$

Given a value $\boldsymbol{\mu} \in \mathcal{D}$ of the parameter we define the RB solution $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ such that:

$$a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = F(v_N; \boldsymbol{\mu}) \quad \forall v_N \in X_N^{\mathcal{N}}. \tag{15}$$

Recalling that $N \ll \mathcal{N}$, we emphasize the fact that to find the RB solution we need just to solve a $N \times N$ linear system, instead of the $\mathcal{N} \times \mathcal{N}$ one of the FE method.

The set S is built in the Offline stage, together with the particular solutions which span $X_N^{\mathcal{N}}$, using a Greedy algorithm [36, 40]. The latter chooses, at each step, the parameter value which maximizes a suitable *a posteriori* error estimator $\boldsymbol{\mu} \mapsto \Delta_N(\boldsymbol{\mu})$ such that

$$|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \le \Delta_N(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}^2 \tag{16}$$

and the algorithm stops when a prescribed tolerance ε_{tol}^* is reached, that is when

$$\Delta_N(\boldsymbol{\mu}) \le \varepsilon_{tol}^* \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{17}$$

[1] We do not actually consider this set of particular solution as a basis, but we perform a Gram-Schmidt orthonormalization process on it [36, 40].

[2] Here $|||\cdot|||_{\boldsymbol{\mu}}$ is the norm induced by the symmetric part of the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$. See (50) for the definition.

The error estimator has to be sharp, in order to avoid an unnecessarily high dimension N of the reduced basis space. Moreover, it must be computationally inexpensive in order to speed up the Greedy algorithm (within which it is computed many times) and to allow the certification of the RB solution during the Online stage. The estimator Δ_N is based on the residual and it requires the computation of a lower bound $\boldsymbol{\mu} \mapsto \alpha_{LB}(\boldsymbol{\mu})$ for the coercivity constant (2), which can be computed using the Successive Constraint Method (SCM) [18, 40].

We want to point out that all the expensive computations (i.e. those whose cost depend on the FE space dimension $\mathcal{N}$) are performed during the Offline stage.

As already mentioned before, the affinity assumptions (4) and (5) are crucial for the Offline-Online decoupling, as it is extensively shown in [36, 40]. If the latter assumptions are not fulfilled, it turns out to be necessary an interpolation strategy (e.g. empirical interpolation [1]) in order to recover them.

The affinity assumptions allow the storage, during the Offline stage, of the matrices corresponding to the parameter independent forms a_q, $q = 1, \dots, Q_a$, restricted to $X_N^{\mathcal{N}}$. Thanks to this fact, during the Online stage the assembly of the reduced basis system only consists in a linear combination of these pre-computed matrices. A similar strategy can also be applied to the computation of the error estimator [36, 40].

3. Stabilized reduced basis: introduction and numerical tests

The main goal of this section is to design an efficient stabilization procedure for the RB method. More specifically, we will make a comparison between an *Offline-Online stabilized* method and an *Offline-only stabilized* one, when used to approximate the solution of a parametric advection-diffusion problem:

$$-\varepsilon(\boldsymbol{\mu})\Delta u(\boldsymbol{\mu}) + \boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla u(\boldsymbol{\mu}) = 0 \quad \text{on } \Omega_o(\boldsymbol{\mu}). \tag{18}$$

given a parameter value $\boldsymbol{\mu} \in \mathcal{D}$ and suitable Dirichlet, Neumann or mixed boundary conditions. Here $\Omega_o(\boldsymbol{\mu})$ is an open subset of $\mathbb{R}^2$, while $\varepsilon(\boldsymbol{\mu})$ and $\boldsymbol{\beta}(\boldsymbol{\mu})$ are functions $\Omega_o(\boldsymbol{\mu}) \to [0, +\infty)$ and $\Omega_o(\boldsymbol{\mu}) \to \mathbb{R}^2$, respectively.

It is well known, from the general theory of the numerical approximation of advection-diffusion equations, that the FE solution of such equations can show significant instability phenomena when the advective term dominates the diffusive one. Let us try to give a more detailed explanation. Let $\mathcal{T}_h$ be a triangulation of Ω and let K be an element of $\mathcal{T}_h$. We say that a problem is *advection dominated* if the following condition holds:

$$\mathbb{P}\mathsf{e}_K(\boldsymbol{\mu})(x) := \frac{|\boldsymbol{\beta}(\boldsymbol{\mu})(x)| h_K}{2\varepsilon(\boldsymbol{\mu})(x)} > 1 \quad \forall \boldsymbol{x} \in K \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{19}$$

where h_K is the diameter of K. It is very well known from literature (e.g. [38]) the FE approximation of advection dominated problems can show significant instability phenomena, e.g. spurious oscillations near the boundary layers. Several ways have been proposed to fix these problems. We choose to resort to a strongly consistent stabilization method: the *Streamline/Upwind Petrov-Galerkin* (SUPG) [3, 16, 22, 23]. For a detailed presentation of the stabilization method for the FE approximation of advection dominated problems, we refer to [38].

Let us now explain the basic ideas of the two RB stabilization methods mentioned before. As regards the *Offline-Online stabilized* method, the choice of the name reveals that the Galerkin projections are performed, in both Offline and Online stage, with respect to the SUPG stabilized bilinear form that is

$$\begin{aligned} a_{stab}(w^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) = & \int_\Omega \varepsilon(\boldsymbol{\mu}) \nabla w^{\mathcal{N}} \cdot \nabla v^{\mathcal{N}} + (\boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla w^{\mathcal{N}}) v^{\mathcal{N}} \\ & + \sum_{K \in \mathcal{T}_h} \delta_K \int_K L^{\boldsymbol{\mu}} v^{\mathcal{N}} \left(\frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L^{\boldsymbol{\mu}}_{SS} v^{\mathcal{N}} \right) \end{aligned} \tag{20}$$

with $w^{\mathcal{N}}, v^{\mathcal{N}}$ chosen in a suitable piecewise polynomial space $X^{\mathcal{N}}$. In (20) $L^{\boldsymbol{\mu}}$ is the parameter dependent advection-diffusion operator, that is

$$L^{\boldsymbol{\mu}} v^{\mathcal{N}} = -\varepsilon(\boldsymbol{\mu})\Delta v^{\mathcal{N}} + \boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla v^{\mathcal{N}}, \tag{21}$$

which can be splitted in its symmetric part $L_S^{\boldsymbol{\mu}}$ and in its skew-symmetric one $L_{SS}^{\boldsymbol{\mu}}$ [38]

$$\begin{aligned} L_S^{\boldsymbol{\mu}} v^{\mathcal{N}} &= -\varepsilon(\boldsymbol{\mu})\Delta v^{\mathcal{N}} - \frac{1}{2}(\operatorname{div} \boldsymbol{\beta}(\boldsymbol{\mu}))v^{\mathcal{N}} \\ L_{SS}^{\boldsymbol{\mu}} v &= \boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla v^{\mathcal{N}} + \frac{1}{2}(\operatorname{div} \boldsymbol{\beta}(\boldsymbol{\mu}))v^{\mathcal{N}}. \end{aligned} \tag{22}$$

Note that in the case of divergence free advection field $\boldsymbol{\beta}$ (as in our numerical tests), we have

$$L_S^{\boldsymbol{\mu}} v^{\mathcal{N}} = -\varepsilon(\boldsymbol{\mu})\Delta v^{\mathcal{N}}, \quad L_{SS}^{\boldsymbol{\mu}} v = \boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla v^{\mathcal{N}}. \tag{23}$$

The bilinear form a_{stab} is coercive, so we can apply the already developed theory in order to use the reduced basis method. The alternative method we want to study - the *Offline-only stabilized* method - consists in using the stabilized form (20) only during the Offline stage and then projecting, during the Online stage, with respect to the standard advection-diffusion bilinear form. The underlying heuristic idea is to be able to build stabilized basis and avoid the Online stabilization.

As we always need an affine expansion of the bilinear form, as in (4), the advantage of using the *Offline-only stabilized* method would be a certain reduction of the computational cost, that could be significant if the number of affine stabilization terms is very high.

We will start from the study of some quite simple test problems, for which is straightforward to obtain the affine expansion. The first one, in Section 3.1, is a problem that shows strong instability effect that can be effectively fixed by the *Offline-Online stabilized* method. The second test case, shown in Section 3.2, is a Poiseuille-Graetz problem [20, 36].

From here on, we will write explicitly the FE space dimension $\mathcal{N}$ only when it will be necessary.

3.1. A first test case

We begin by studying a problem depending only on one "physical" parameter, actually the global Péclet number. Let Ω be the unit square in $\mathbb{R}^2$, that is $(0,1) \times (0,1)$. The domain is sketched in Figure 1. The problem is the following one:

$$\begin{cases} -\dfrac{1}{\mu}\Delta u(\mu) + (1,1) \cdot \nabla u(\mu) = 0 & \text{in } \Omega \\ u(\mu) = 0 & \text{on } \Gamma_1 \cup \Gamma_2 \\ u(\mu) = 1 & \text{on } \Gamma_3 \cup \Gamma_4 \end{cases} \tag{24}$$

with $\mu > 0$. Note that μ is the Péclet number of our problem, so we will be interested in the case in which μ is high (as $\boldsymbol{\mu}$ is now a scalar, in this Section we identify $\boldsymbol{\mu}$ and μ).

In order to pursue a finite element approximation, we need to write a suitable weak formulation of the problem:

$$\begin{aligned} &\text{find } u(\mu) \in V := \{v \in H^1(\Omega) \mid v|_{\Gamma_1 \cup \Gamma_2} = 0, v|_{\Gamma_3 \cup \Gamma_4} = 1\} \text{ s.t.} \\ &a(u(\mu), v; \mu) = 0 \quad \forall v \in H_0^1(\Omega) \end{aligned} \tag{25}$$

where

$$a(w, v; \mu) := \int_\Omega \frac{1}{\mu}\nabla w \cdot \nabla v + (\partial_x u + \partial_y u)\, v. \tag{26}$$

We know from the general theory of PDEs that the problem (25) admits a unique solution.

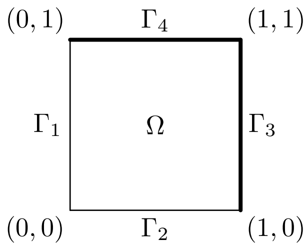


Figure 1: First test case: domain. On the bold sides we impose $u = 1$, while on the rest of the boundary $u = 0$.

Let $\mathcal{T}_h$ be a proper triangulation of Ω. The finite element approximation of the problem turns out to be:

$$\begin{aligned} &\text{find } u_h(\mu) \in V_h := \{v_h \in \mathbb{P}^r(\mathcal{T}_h) \mid v_h|_{\Gamma_1 \cup \Gamma_2} = 0, v_h|_{\Gamma_3 \cup \Gamma_4} = 1\} \text{ s.t.} \\ &a(u_h(\mu), v^{\mathcal{N}}; \mu) = 0 \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \end{aligned} \tag{27}$$

with $X^{\mathcal{N}}$ defined as the subspace of $\mathbb{P}^r(\mathcal{T}_h)$ made up by the functions that vanish on the boundary of Ω. Here $\mathbb{P}^r(\mathcal{T}_h)$ is defined by

$$\mathbb{P}^r(\mathcal{T}_h) = \{v_h \in H^1(\Omega) \,|\, v_h|_K \in \mathbb{P}^r(K), K \in \mathcal{T}_h\} \tag{28}$$

Finally, let us define the function g_h as a lifting in $\mathbb{P}^r(\mathcal{T}_h)$ of the Dirichlet boundary condition. We can now define $u^{\mathcal{N}}(\mu) = u_h(\mu) - g_h$, that belongs to $X^{\mathcal{N}}$. Thus we obtain the final FE formulation of our problem:

$$\begin{aligned} &\text{find } u^{\mathcal{N}}(\mu) \in X^{\mathcal{N}} \text{ s.t.} \\ &a(u^{\mathcal{N}}(\mu), v^{\mathcal{N}}; \mu) = F(v^{\mathcal{N}}; \mu) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}. \end{aligned} \tag{29}$$

where

$$F(v^{\mathcal{N}}; \mu) := -a(g_h, v^{\mathcal{N}}; \mu). \tag{30}$$

When the parameter μ takes "small" values we do not have instability problems. More precisely, we can obtain stable solutions if

$$\mathbb{P}\mathsf{e}_K := \frac{\mu\, h_K}{\sqrt{2}} < 1 \quad \forall K \in \mathcal{T}_h. \tag{31}$$

In Figure 2 the approximated $\mathbb{P}^1$-FE solution obtained for $\mu = 6$ is shown. We can use the RB method to approximate the solution of the problem (24) for a parameter range from 1 to 10. In Figure 3 we show some representative RB solutions computed in correspondence of some value of the parameter μ. The dimension of the RB space is $N = 8$. In Figure 4 we report the energy norm of the difference between the RB solution and the FE solution (RB approximation error) as a function of the parameter μ. More precisely, we show the linear interpolation of the RB approximation error computed for 50 equispaced parameter values between 1 and 10. The vertical dashed lines are plotted in correspondence of the parameter values selected by the greedy algorithm [36]. It is evident that the RB approximation error tends to vanish in correspondence of the parameter values selected by the greedy algorithm, as expected by definition of RB solution to guarantee the consistency of the method.

A more interesting case is when the Péclet number assumes higher values, thus fulfilling condition (19). In Figure 5 the solution obtained by using a non-stabilized FE approximation with

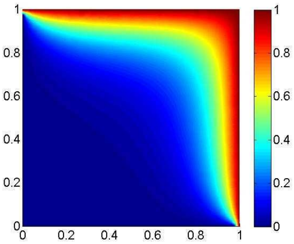
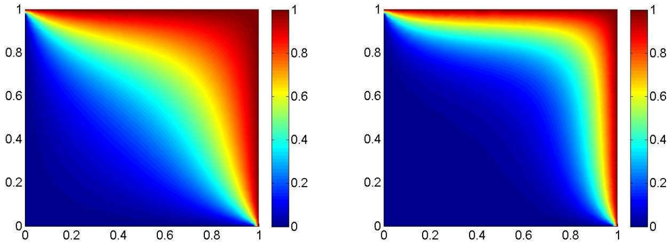


Figure 2: First test case, low Péclet number. FE solution for $\mu = 6$.



(a) $\mu = 2$ (b) $\mu = 6$

Figure 3: First test case, low Péclet number. Representative RB solutions for different values of the parameter.

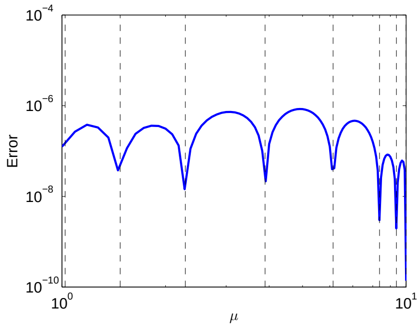


Figure 4: First test case, low Péclet number. RB approximation error as a function of the parameter.

$\mu = 600$ is represented. Even in this case we can perform a RB approximation of the solution, but the RB solutions reflect all the instability problems of the FE solution, as we can see in Figure 6. For this simple case, if we let the parameter range from 100 to 1000 the greedy algorithm converges and the energy norm of the difference between the RB solution and the FE solution behaves as for lower values of the Péclet number, as we can see in Figure 7, but this procedure would lead us to wrong physical results. This happens because the target of the RB approach is to approximate the exact continuous solution of the problem by trying to recover the FE solution using a significantly lower number of degrees of freedom. The point is now that the FE solution is not a good approximation of the exact one.

A possible way to fix this instability problems could be to use some stabilization methods. We chose to use the SUPG stabilization method. First of all, we have to impose the stabilization correction to the weak formulation (27). We thus define the following bilinear form:

$$s(w_h, v_h; \mu) := \sum_{K \in \mathcal{T}_h} \delta_K \int_K \left(-\frac{1}{\mu} \Delta w_h + (1,1) \cdot \nabla w_h \right) \left(\frac{h_K}{\sqrt{2}} (1,1) \cdot \nabla v_h \right) \tag{32}$$

with $w_h, v_h \in \mathbb{P}^r(\mathcal{T}_h)$. We chose, as before, to use $\mathbb{P}^1$ finite elements, that is $r = 1$. As piecewise linear functions have null Laplacian, the latter form reduces to:

$$s(w_h, v_h; \mu) = \sum_{K \in \mathcal{T}_h} \frac{\delta_K h_K}{\sqrt{2}} \int_K (\partial_x w_h + \partial_y w_h)(\partial_x v_h + \partial_y v_h) \tag{33}$$

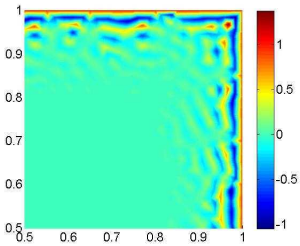
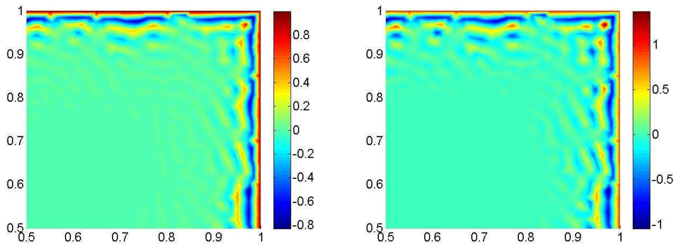


Figure 5: First test case, high Péclet number. FE solution for $\mu = 600$ (zoom on $[0.5, 1] \times [0.5, 1]$).



(a) $\mu = 400$ (b) $\mu = 600$

Figure 6: First test case, high Péclet number. representative RB solutions for different values of the parameter (zoom on $[0.5, 1] \times [0.5, 1]$).

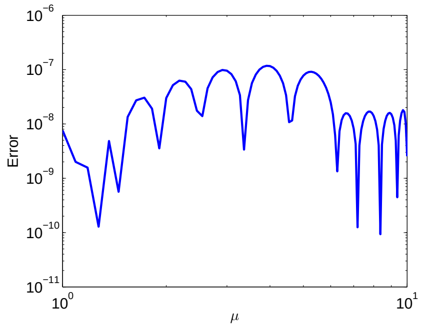


Figure 7: First test case, high Péclet number. RB approximation error expressed as a function of the parameter

again with $w_h, v_h \in \mathbb{P}^1(\mathcal{T})$. We can now define the final formulation of the stabilized FE problem:

$$\begin{aligned}
&\text{find } u^{s\mathcal{N}}(\mu) \in X^{\mathcal{N}} \text{ s.t.} \\
&a(u^{s\mathcal{N}}(\mu), v^{\mathcal{N}}; \mu) + s(u^{s\mathcal{N}}(\mu), v^{\mathcal{N}}; \mu) = F(v^{\mathcal{N}}; \mu) + F^s(v^{\mathcal{N}}; \mu) \quad \forall v \in X^{\mathcal{N}}.
\end{aligned} \tag{34}$$

where F is the same as in (30) and F^s is

$$F^s(v^{\mathcal{N}}) := -s(g_h, v^{\mathcal{N}}). \tag{35}$$

We finally define $u_h^s(\mu) = u^{s\mathcal{N}}(\mu) + g_h$, that is the FE stabilized solution which satisfies the Dirichlet boundary conditions. Let us call a_{stab} the bilinear form and f_{stab} the right-hand side, that is

$$\begin{aligned}
a_{stab} &= a + s, \\
F_{stab} &= F + F^s.
\end{aligned} \tag{36}$$

It is straightforward to prove that, for our choice of polynomial approximation space, we do not need to fulfil any requirement on the weights δ_K to guarantee the stability of the discrete problem (34) with respect to the SUPG norm [38]:

$$\begin{aligned}
\|v^{\mathcal{N}}\|^2_{SUPG,\boldsymbol{\mu}} = |||v^{\mathcal{N}}|||^2_{\boldsymbol{\mu}} + \sum_{K \in \mathcal{T}_h} \delta_K \left(L_{SS} v^{\mathcal{N}}, \frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L^{\boldsymbol{\mu}}_{SS} v^{\mathcal{N}} \right)_K \\
\forall v^{\mathcal{N}} \in V^{\mathcal{N}}, \quad \forall \boldsymbol{\mu} \in \mathcal{D},
\end{aligned} \tag{37}$$

simply because this is actually the norm induced by $a_{stab}(\cdot,\cdot;\boldsymbol{\mu})$ on $V^{\mathcal{N}}$. In (37) $|||\cdot|||_{\boldsymbol{\mu}}$ is the norm induced by the symmetric part of the advection-diffusion operator that is

$$|||v|||_{\boldsymbol{\mu}}^2 = \int_{\Omega_o(\boldsymbol{\mu})} |\nabla v|^2, \quad \forall v \in H^1(\Omega_o(\boldsymbol{\mu})), \tag{38}$$

while $L_{SS}^{\boldsymbol{\mu}}$ is the skew-symmetric part of the advection-diffusion operator, defined in (22). We then set $\delta_K = 1$ for each element $K \in \mathcal{T}_h$. In Figure 8 is shown a SUPG stabilized FE solution for $\mu = 600$.

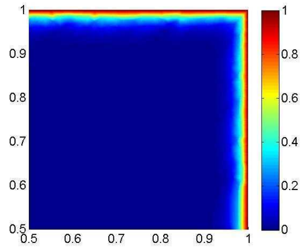


Figure 8: First test case, high Péclet number. SUPG FE solution for $\mu = 600$ (zoom on $[0.5, 1] \times [0.5, 1]$).

Now we can try the two different approaches described before: the *Offline-Online stabilized* and the *Offline-only stabilized* methods. As regards the first one, we have just to perform the whole RB standard method simply using a_{stab} instead of a. The *Offline-only stabilized* approach consists in using the form a_{stab} during the Offline stage, in order to obtain stable reduced basis, and to perform the Online Galerkin projection with respect to the form a. Formally, denoted by $X_N^{\mathcal{N}}$ the space spanned by the reduced basis, the *Offline-Online stabilized* solution $u_N^s(\mu) \in X_N^{\mathcal{N}}$ satisfies

$$a_{stab}(u_N^s(\mu), v_N; \mu) = F_{stab}(v_N; \mu) \quad \forall\, v_N \in X_N^{\mathcal{N}} \tag{39}$$

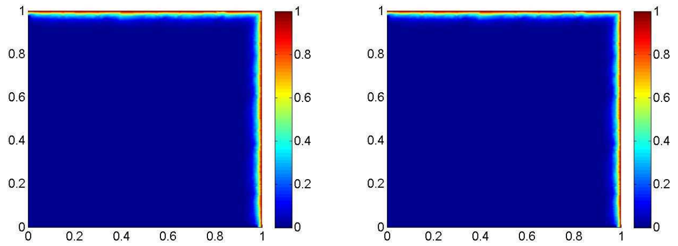
while the *Offline-only stabilized* solution $u_N(\mu) \in X_N^{\mathcal{N}}$ is such that

$$a(u_N(\mu), v_N; \mu) = F(v_N; \mu) \quad \forall\, v \in X_N^{\mathcal{N}}. \tag{40}$$

By using the norm induced by a_{stab} to carry out the Offline stage, we are actually taking the SUPG stabilized FE solution $u^{s\,\mathcal{N}}(\mu)$ as the "exact" one. So it makes sense to measure the performance of the method by evaluating the difference between the RB solution and the stabilized FE one.

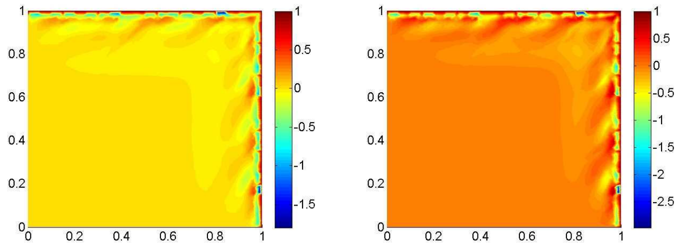
The *Offline-Online stabilized* method, as expected, produces stable RB solutions, as shown in Figure 9, and the actual error, with respect to the stabilized FE solution, is smaller than the tolerance guaranteed by the greedy algorithm ($\varepsilon_{tol}^* = 10^{-5}$), as we can see in Figure 11. On the contrary, the behaviour of the *Offline-only stabilized* approach is very unsatisfactory. As we can see in Figure 10, even though the reduced basis are stable, the *Offline-only stabilized* RB solutions show large oscillations. *We have actually shown that a Galerkin projection on a subspace spanned by stable functions does not guarantee a stable solution for large values of the Péclet number.*

In order to better understand which are the causes of the instability of the *Offline-only stabilized* method, we try to use a locally refined mesh to build the reduced basis, instead of the SUPG stabilization method, during the Offline stage. In this case, "locally" means that we refine the mesh in the area in which we expect that the boundary layer will arise. Acting in this way, we can obtain Offline stable reduced basis without resorting to any stabilization method, because the condition (31) is now satisfied, at least where we previously had instability problems. Obviously, by



(a) $\mu = 400$ (b) $\mu = 600$

Figure 9: First test case, high Péclet number. Representative *Offline-Online stabilized* RB solutions for different values of the parameter (zoom on $[0.5, 1] \times [0.5, 1]$).



(a) $\mu = 400$ (b) $\mu = 600$

Figure 10: First test case, high Péclet number. Representative *Offline-only stabilized* RB solutions for different values of the parameter (zoom on $[0.5, 1] \times [0.5, 1]$).

increasing the number of degrees of freedom, we quite increase the computational cost. The Offline algorithm, performed using the refined mesh and the original bilinear form, produces 14 basis and it takes 711 seconds while the Offline algorithm, carried out with the coarser mesh and the stabilized bilinear form, takes only 114 seconds and builds 8 basis. The RB solutions obtained for the same parameter range as before (Figure 12) do not show instability phenomena, so an explanation of the behaviour of the *Offline-only stabilized* method tested before has to be found analysing the use of different bilinear forms in the two stages of the RB method, as we will do further. The distance in energy norm[3] between the FE solution and the RB one is showed in Figure 13 (we recall that $\varepsilon^*_{tol} = 10^{-5}$). Comparing Figure 12 and Figure 9 we can also see how the stabilization method tends to "smooth" the boundary layer.

Before going on, in Table 1, we report informations about the computations performed in this section. In all the numerical tests we used a tolerance $\varepsilon^*_{tol} = 10^{-5}$ on the greedy algorithm. With T_{off} and T_{on} we mean the time elapsed during the Offline and Online stage respectively. We can underline how a stabilization technique performs computationally better than a local refinement technique.

[3] i.e. the norm $|||\cdot|||_{\boldsymbol{\mu}}$ induced by the symmetric part of the original advection-diffusion bilinear form (see (50)).

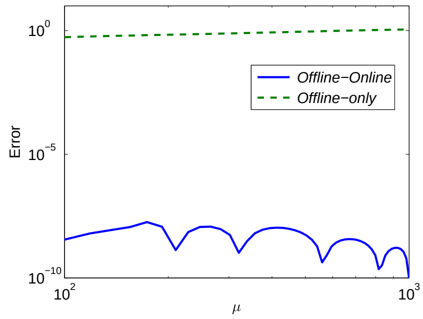


Figure 11: First test case, high Péclet number. comparison of the RB approximation error obtained for the two different stabilization strategies; here the error is expressed as a function of the parameter.

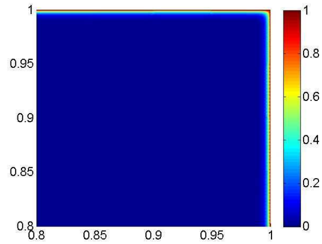


Figure 12: First test case, high Péclet number. RB solution with locally refined mesh for $\mu = 400$ (zoom on $[0.8, 1] \times [0.8, 1]$).

$\mathcal{D}$	$\mathcal{N}$	Off. stab.	$T_{\text{Off}}\,(s)$	N	On. stab.	$T_{\text{On}}\,(s)$
$[100, 1000]$	2605	yes	114	8	no	$1.78 \cdot 10^{-3}$
					yes	$1.95 \cdot 10^{-3}$
$[1, 10]$	2605	no	86	13	no	$1.83 \cdot 10^{-3}$
$[100, 1000]$	2605	no	98	14	no	$1.81 \cdot 10^{-3}$
$[100, 1000]$	21313	no	711	14	no	$1.79 \cdot 10^{-3}$

Table 1: First test case. Numerical tests

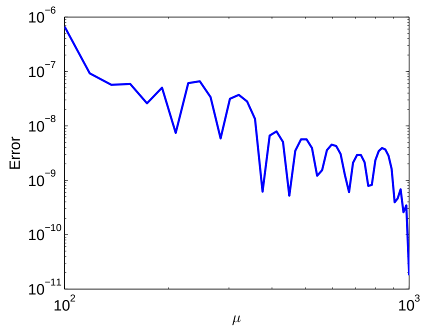


Figure 13: First test case, high Péclet number. RB approximation error for locally refined mesh, expressed as a function of the parameter.

3.2. A second test case: Poiseuille-Graetz flow

We now focus on a different situation, a Graetz problem [11, 20, 36, 42], in which we have two parameters: a physical one (the Péclet number) and a geometrical one (the length of the domain). The Graetz problem deals with steady forced heat convection (advective phenomenon) combined with heat conduction (diffusive phenomenon) in a duct with walls at different temperature. Let us define $\boldsymbol{\mu} = (\mu_1, \mu_2)$ where both μ_1 and μ_2 are positive real numbers. Let $\Omega_o(\boldsymbol{\mu})$ be the rectangle $(0, 1+\mu_2) \times (0,1)$ in $\mathbb{R}^2$. The domain is shown in Figure 14.

The problem is to find a solution $u(\boldsymbol{\mu})$, representing the temperature distribution, such that:

$$\begin{cases} -\dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + 4\,y(1-y)\partial_x u(\boldsymbol{\mu}) = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_{o\,1}(\boldsymbol{\mu}) \cup \Gamma_{o\,2}(\boldsymbol{\mu}) \cup \Gamma_{o\,6}(\boldsymbol{\mu}) \\ u(\boldsymbol{\mu}) = 1 & \text{on } \Gamma_{o\,3}(\boldsymbol{\mu}) \cup \Gamma_{o\,5}(\boldsymbol{\mu}) \\ \dfrac{\partial u}{\partial \nu} = 0 & \text{on } \Gamma_{o\,4}(\boldsymbol{\mu}). \end{cases} \tag{41}$$

In order to exploit a RB approach, we need to set a reference domain Ω that we choose as $\Omega = (0,2)\times(0,1)$, that is the original domain $\Omega_o(\boldsymbol{\mu})$ corresponding to $\mu_2 = 1$. It is useful to subdivide the reference domain into subdomains, so we define $\Omega^1 = (0,1) \times (0,1)$ and $\Omega^2 = (1,2) \times (0,1)$. Now we need the affine transformation that maps the reference domain into the original one [36, 40], so we define:

$$\begin{aligned} &T^1(\boldsymbol{\mu})\colon \Omega^1 \to \mathbb{R}^2, \\ &T^1\left(\begin{pmatrix} x \\ y \end{pmatrix}; \boldsymbol{\mu}\right) = \begin{pmatrix} x \\ y \end{pmatrix}, \end{aligned} \tag{42}$$

that is the identity map, and

$$\begin{aligned} &T^2(\boldsymbol{\mu})\colon \Omega^1 \to \mathbb{R}^2, \\ &T^2\left(\begin{pmatrix} x \\ y \end{pmatrix}; \boldsymbol{\mu}\right) = \mathbf{G}^2 \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \mu_2 \\ 0 \end{pmatrix}, \end{aligned} \tag{43}$$

where

$$\mathbf{G}^2 = \begin{pmatrix} \mu_2 & 0 \\ 0 & 1 \end{pmatrix}.$$

If we glue together these two transformations, for each $\boldsymbol{\mu} \in \mathcal{D}$ we actually define a transformation $T(\boldsymbol{\mu})$ of the whole domain Ω. Note that $T(\boldsymbol{\mu})$ is a continuous one-to-one transformation.

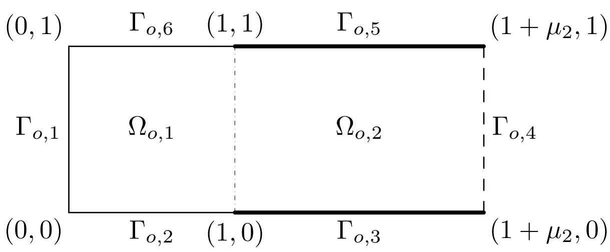


Figure 14: Poiseuille-Graetz test case problem: parametrized domain. Boundary conditions: $u = 1$ on the bold sides, homogeneous Neumann on the dashed side and homogeneous Dirichlet on the remaining boundary sides.

The weak formulation of the Poiseuille-Graetz problem is the following one:

$$\begin{aligned}&\text{find } u_o(\boldsymbol{\mu}) \in V_o := \{v_o \in H^1(\Omega_o) \mid v|_{\Gamma_{o\,3}(\boldsymbol{\mu})\cup\Gamma_{o\,5}(\boldsymbol{\mu})} = 1, v|_{\Gamma_{o\,1}(\boldsymbol{\mu})\cup\Gamma_{o\,2}(\boldsymbol{\mu})\cup\Gamma_{o\,6}(\boldsymbol{\mu})} = 0\} \text{ s.t.}\\ &a(u_o(\mu), v_o; \boldsymbol{\mu}) = 0 \quad \forall v \in H^1_0(\Omega),\end{aligned} \tag{44}$$

where:

$$a(w_o, v_o; \boldsymbol{\mu}) := \int_{\Omega_o(\boldsymbol{\mu})} \frac{1}{\mu_1}\nabla w \cdot \nabla v + 4\,y(1-y)\partial_x u\, v. \tag{45}$$

We set the standard FE problem, exactly as we did in (27), introducing the stabilization term. To do so, let us define a mesh $\mathcal{T}_h$ on the reference domain Ω and let us call $\mathcal{T}_h^1$ and $\mathcal{T}_h^2$ the restrictions $\mathcal{T}_h$ to Ω_1 and Ω_2, respectively. We can also define a mesh on $\Omega_o(\boldsymbol{\mu})$ just by taking the image of $\mathcal{T}_h$ through the transformation $T(\cdot, \boldsymbol{\mu})$, that is:

$$\mathcal{T}_{h,o}(\boldsymbol{\mu}) = \{K_o(\boldsymbol{\mu}) = T(K; \boldsymbol{\mu}) \mid K \in \mathcal{T}_h\}.$$

We can now write the stabilization term, for the $\mathbb{P}^1$-FE case, to be added to the left-hand side:

$$s(w_h, v_h; \boldsymbol{\mu}) := \sum_{K_o(\boldsymbol{\mu})\in\mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \int_{K_o(\boldsymbol{\mu})} (4\,y(1-y)\partial_x w_h)\left(h_{K_o(\boldsymbol{\mu})}\partial_x v_h\right). \tag{46}$$

Now we have to set the problem onto the reference domain, thus our problem turns out to be:

$$\begin{aligned}&\text{find } u(\boldsymbol{\mu}) \in V := \{v_h \in \mathbb{P}^1(\Omega) \mid v_h|_{\Gamma_{o\,3}(\boldsymbol{\mu})\cup\Gamma_{o\,5}(\boldsymbol{\mu})} = 1, v_h|_{\Gamma_{o\,1}(\boldsymbol{\mu})\cup\Gamma_{o\,2}(\boldsymbol{\mu})\cup\Gamma_{o\,6}(\boldsymbol{\mu})} = 0\} \text{ s.t.}\\ &a(u_h(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + s(u_h(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = 0 \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}\end{aligned} \tag{47}$$

where $X^{\mathcal{N}}$ is defined as in the previous section, a is:

$$\begin{aligned}a(w_h, v_h; \boldsymbol{\mu}) := &\int_{\Omega^1} \frac{1}{\mu_1}\nabla w_h \cdot \nabla v_h + 4\,y(1-y)\partial_x w_h\, v_h\\ &+ \int_{\Omega^2} \frac{1}{\mu_1\mu_2}\partial_x w_h \partial_y v_h + \frac{\mu_2}{\mu_1}\partial_x w_h \partial_y v_h\\ &\quad + 4\mu_2\, y(1-y)\partial_x w_h\, v_h\end{aligned} \tag{48}$$

and s is:

$$\begin{aligned}s(w_h, v_h; \boldsymbol{\mu}) := &\sum_{K\in\mathcal{T}_h^1} h_K \int_K (4\,y(1-y)\partial_x w_x)\,\partial_x v_h\\ &+ \sum_{K\in\mathcal{T}_h^2} \frac{h_K}{\sqrt{\mu}_2}\int_K (4\,y(1-y)\partial_x w_x)\,\partial_x v_h.\end{aligned} \tag{49}$$

By introducing a lifting of the Dirichlet boundary condition we can obtain the stabilized FE formulation (34). We point out that for $K \in \mathcal{T}_h^2$ we are choosing $\delta_{K_o(\boldsymbol{\mu})}$ such that $\delta_{K_o(\boldsymbol{\mu})}\, h_{K_o(\boldsymbol{\mu})} = h_K\sqrt{\mu_2}$. The underlying idea is that we would like to choose $\delta_{K_o(\boldsymbol{\mu})} = 1$ but we have to consider how the element diameter transforms, that is $h_{K_o(\boldsymbol{\mu})} \approx h_K\sqrt{J(\boldsymbol{\mu})} = h_K\sqrt{\mu_2}$, where $J(\boldsymbol{\mu})$ is the Jacobian of the transformation $T(\boldsymbol{\mu})$. This rescaling is done mainly for preserving the convergence rate of the SUPG method. We need to make an assumption like this also because it would make no sense, in an RB point of view, to compute Online every exact value of $h_{K_o(\boldsymbol{\mu})}$. Indeed, the Online stage of the RB method actually is independent of the triangulation $\mathcal{T}_h$.

As pointed out in [10], if the advection dominated condition (19) is not fulfilled for all $K \in \mathcal{T}$ we locally lose even the h^r convergence rate of the standard FE method. A possible way to overcome this trouble is to act on the weights δ_K, distinguishing between the elements for which $\mathbb{P}\mathsf{e}_K > 1$ and $\mathbb{P}\mathsf{e}_K \leq 1$. We want to observe that, unfortunately, by using a weighting that depends on both parameter and element size we lose the affinity assumption (4) on the bilinear form, or better, we lose that assumption with a number of affine terms Q_a independent of $\mathcal{N}$. So, if we are facing problems in which the advection dominated condition (19) is not fulfilled for all $K \in \mathcal{T}_h$ and we want to rigorously recover the convergence order of the FE method, in order to resort to a weighting $\delta = \delta(\boldsymbol{x}, \boldsymbol{\mu})$ (as proposed in [10]) we need to exploit some interpolation techniques involving the empirical interpolation [1][4]. In this case it would be also worth to check if it were possible to define a weighting that does not depend on each h_K, but on the mesh size $h = \max_{K \in \mathcal{T}_h} h_k$, under suitable regularity assumptions [23].

We would like also to recall that the convergence performances of the stabilization method depend on the regularity properties of the mesh. So, as the meshes $\mathcal{T}_{h,o}(\boldsymbol{\mu})$ we are actually using to stabilize on the original domains are the image through T of the triangulation defined on the reference domain, we should guarantee that the transformation T does not worsen the properties of the reference triangulation. In our numerical tests the reference domain will be the one corresponding to $\mu_2 = 1$ and we will let the parameter range from 0.5 to 4, so we will not have an excessive deformation. We will also use a quite coarse mesh (mesh size $h = 0.06$) and high values for μ_1 (from 10000 to 20000) in order to have significant instability problems. The point is that the boundary layer arises in an area in which the norm of the advection field (an thus the value of the local Péclet number) is relatively small. In Figure 15 we show the local Péclet number computed on the reference domain ($\mu_2 = 1$) in correspondence of the quadrature point used to compute the FE matrices, and thus the RB ones. We would like to point out that even if the advection field vanishes as we get close to the boundary, the Péclet number that is actually considered is just the one computed in the quadrature points. The lowest value assumed by the local Péclet number is then 1.79, while the highest is 307.

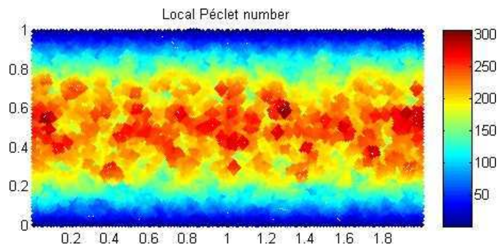


Figure 15: Poiseuille-Graetz test case. Local Péclet number computed in the quadrature points of the reference domain ($\mu_2 = 1$.)

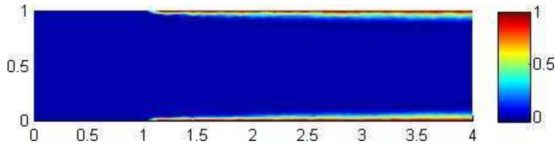
In Figures 16 and 17 we show some solutions obtained respectively by *Offline-Online stabilized* method and *Offline-only stabilized* method.

Finally, in Figure 18 we show the error curves of the two methods. As in the previous test-case, we can see that only the *Offline-Online stabilized* produces satisfactory results, even if now the *Offline-only stabilized* method has slightly better performances than in the previous test case. Here we used a tolerance[5] on the greedy algorithm $\varepsilon^*_{tol} = 10^{-3}$.

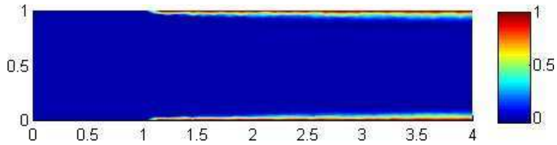
In Table 2, we report some informations about the numerical tests performed using the parameter space $\mathcal{D} = [10000, 20000] \times [0.5, 4]$. The tolerance for the greedy algorithm is $\varepsilon^*_{tol} = 10^{-3}$.

[4] We have to remark that the weighting proposed in [10] is discontinuous in both $\boldsymbol{x}$ and $\boldsymbol{\mu}$ even if the coefficients ε and $\boldsymbol{\beta}$ are smooth.

[5] The tolerance is on the stabilized energy norm, that is greater than the non-stabilized one (see (50) for the definitions).

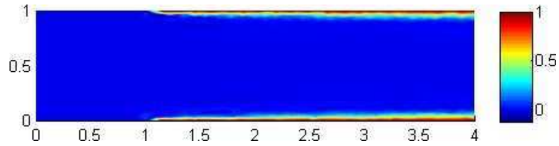


(a) $\mu_1 = 12500$

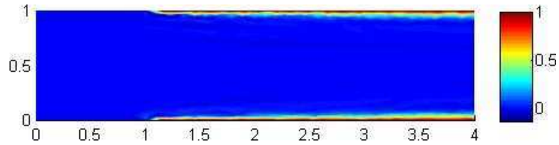


(b) $\mu_1 = 17500$

Figure 16: Poiseuille-Graetz test case. Representative *Offline-Online stabilized* RB solutions for $\mu_2 = 3$ and several values of μ_1.

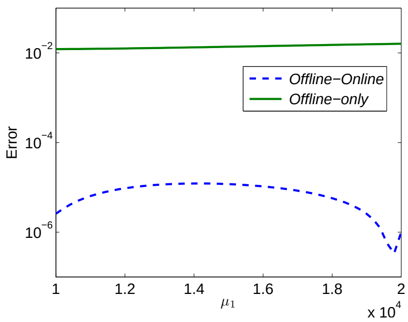


(a) $\mu_1 = 12500$

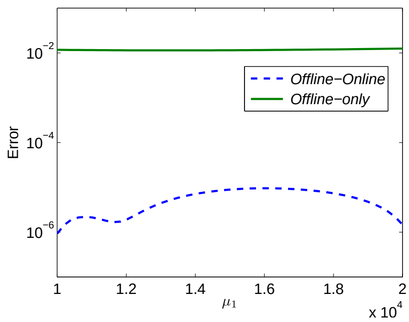


(b) $\mu_1 = 17500$

Figure 17: Poiseuille-Graetz test case. Representative *Offline-only stabilized* RB solutions for $\mu_2 = 3$ and several values of μ_1.



(a) $\mu_2 = 2.5$



(b) $\mu_2 = 3.5$

Figure 18: Poiseuille-Graetz test case. Comparison of the RB approximation error obtained for the two different stabilization strategies; here the error is expressed as a function of the parameter μ_1, given a value of μ_2.

$\mathcal{N}$	Off. stab.	$T_{\text{Off}}\,(s)$	N	On. stab.	$T_{\text{On}}\,(s)$
1309	yes	168	15	no	$0.97 \cdot 10^{-3}$
				yes	$1.04 \cdot 10^{-3}$
1309	no	341	22	no	$1.81 \cdot 10^{-3}$

Table 2: Poiseuille-Graetz test case. Numerical tests

3.3. Discussions on the results

Observing the results obtained up to now, it seems that the best way to perform stabilization is the *Offline-Online stabilized* one. But let us try to understand why the *Offline-only stabilized* option has a bad behaviour.

Let us introduce some notation. Let us call *energy norm* the norm on $H^1_0(\Omega(\boldsymbol{\mu}))$ induced by the symmetric part of the advection diffusion operator a and *stabilized energy norm* the one induced by the symmetric part of a_{stab}. In symbols:

$$
\begin{aligned}
|||\cdot|||_{\boldsymbol{\mu}} &= \sqrt{a^{sym}(\cdot,\cdot;\boldsymbol{\mu})} \\
|||\cdot|||_{\boldsymbol{\mu},stab} &= \sqrt{a^{sym}_{stab}(\cdot,\cdot;\boldsymbol{\mu})}.
\end{aligned} \tag{50}
$$

First of all we have to note that by performing the Offline stage using the stabilized operator and the standard *a posteriori* error estimators, we are actually assuring that the "reliable" RB approximation is the *Offline-Online stabilized* one. This is because the Greedy algorithm [36, 40] (performed using the stabilized form a_{stab}) guarantees that

$$
|||u^s_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu},stab} \leq \varepsilon^*_{tol} \quad \forall \boldsymbol{\mu} \in \mathcal{D}^6, \tag{51}
$$

so the Offline procedure actually allows us to control only the error committed by the *Offline-Online stabilized* method.

Thus we have to find some estimates for the difference in norm between $u_N(\boldsymbol{\mu})$ and $u^{s\,\mathcal{N}}(\boldsymbol{\mu})$. We can try by splitting the difference in this way:

$$
|||u_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq |||u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} + |||u^s_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}. \tag{52}
$$

Of course, it holds that

$$
|||u^s_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq |||u^s_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu},stab} \leq \varepsilon^*_{tol}, \tag{53}
$$

therefore we have to provide an estimate of the distance with respect to the energy norm between $u_N(\boldsymbol{\mu})$ and $u^s_N(\boldsymbol{\mu})$. To do so we can simply start from the definition:

$$
\begin{aligned}
|||u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu})|||^2_{\boldsymbol{\mu}} =& a(u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu}), u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \\
=& - F^s(u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu}); \boldsymbol{\mu}) + s(u^s_N(\boldsymbol{\mu}), u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \\
=& s(u^s_N(\boldsymbol{\mu}) + g_h, u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu}); \boldsymbol{\mu}),
\end{aligned} \tag{54}
$$

where g_h is the lifting of the Dirichlet boundary data. For the SUPG stabilization with $\mathbb{P}^1$ elements, the following bound holds:

$$
\begin{aligned}
|s(u^s_N(\boldsymbol{\mu}) + g_h, u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu}))| \; &\leq h_{max}(\boldsymbol{\mu}) \, \|\boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla(u^s_N(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\
&\quad \cdot \left\| \frac{\boldsymbol{\beta}(\boldsymbol{\mu})}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} \cdot \nabla(u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu})) \right\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\
&\leq h_{max}(\boldsymbol{\mu}) \, \|\boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla(u^s_N(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\
&\quad \cdot |u_N(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu})|_{H^1_0(\Omega_o(\boldsymbol{\mu}))}.
\end{aligned}
$$

where $h_{max}(\boldsymbol{\mu}) = \max_{K \in \mathcal{T}_h} h_K \sqrt{J(T(\cdot,\boldsymbol{\mu}))}$. As the energy norm is equivalent to $|\cdot|_{H^1_0}$, we have that:

$$
|v|_{H^1_0(\Omega_o(\boldsymbol{\mu}))} \;\leq\; C(\boldsymbol{\mu}) \, |||v|||_{\boldsymbol{\mu}} \quad \forall v \in H^1_0(\Omega_o(\boldsymbol{\mu})). \tag{55}
$$

We can also bound the L^2-norm of the streamline derivative of the *Offline-Online* stabilized RB solution with that of the FE stabilized solution:

$$
\begin{aligned}
\|\boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla(u^s_N(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \; &\leq \|\boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla(u^s_h(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\
&\quad + \|\boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla(u^s_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\
&\leq \|\boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla(u^s_h(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} + C(\boldsymbol{\mu}) \|\boldsymbol{\beta}(\boldsymbol{\mu})\|_{L^\infty(\Omega_o(\boldsymbol{\mu}))} \varepsilon^*_{tol}.
\end{aligned} \tag{56}
$$

[6] More precisely, we can guarantee that the inequality holds for all $\boldsymbol{\mu}$ in Ξ_{train}, that is the subset of $\mathcal{D}$ in which the Greedy algorithm chooses the parameter values corresponding to the reduced basis [36, 40].

We recall that $u_h^s(\mu) = u^{s\,\mathcal{N}}(\mu) + g_h$.

We note that this argument allows us to prove the inequalities shown in the following proposition, that are upper bound for the distance with respect to the energy norm between $u_N(\boldsymbol{\mu})$ and $u^{s\,\mathcal{N}}(\boldsymbol{\mu})$. The first one is a proper *a priori* upper bound, while the second one is a sharper bound, but not properly *a priori*.

Proposition 3.1 (Error bounds for the *Offline-only stabilized* method)**.** *The following inequalities hold:*

$$\begin{aligned} |||u_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq & h_{max}(\boldsymbol{\mu})C(\boldsymbol{\mu})\|\boldsymbol{\beta}(\boldsymbol{\mu})\cdot\nabla(u_h^s(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ & + \left(1 + h_{max}(\boldsymbol{\mu})C(\boldsymbol{\mu})^2\|\boldsymbol{\beta}(\boldsymbol{\mu})\|_{L^\infty(\Omega_o(\boldsymbol{\mu}))}\right)\varepsilon_{tol}^*. \end{aligned} \tag{57}$$

$$\begin{aligned} |||u_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq & \, h_{max}(\boldsymbol{\mu})\,C(\boldsymbol{\mu})\|\boldsymbol{\beta}(\boldsymbol{\mu})\cdot\nabla(u_N^s(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ & + |||u_N^s(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}. \end{aligned} \tag{58}$$

Remark 3.1. We point out that the bound in (57) depends on the L^2 norm of the streamline derivative of the stabilized solution (with the non homogeneous Dirichlet boundary conditions). *This means that the Offline-only stabilized method has better performances when applied to problems in which the strongest variations occur along a direction orthogonal to the advection field. This could happen in the cases in which the boundary layers are parallel to the advection field, e.g. the Poiseuille-Graetz problem.* The "improvement" of the approximation is confirmed by comparing the numerical results shown in Figures 11 and 18.

Remark 3.2. A very similar computation shows that the error $|||u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$ is actually the same as $|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$ (just drop N in the previous computaions).

We performed some numerical tests for the bound in (58). The results are shown in Figures 19 and 20. Concerning the first test case we set:

$$C(\mu) = \sqrt{\mu}$$

and for the Poiseuille-Graetz problem:

$$C(\boldsymbol{\mu}) = \sqrt{\mu_1}.$$

With these choices, (55) is actually an equality.

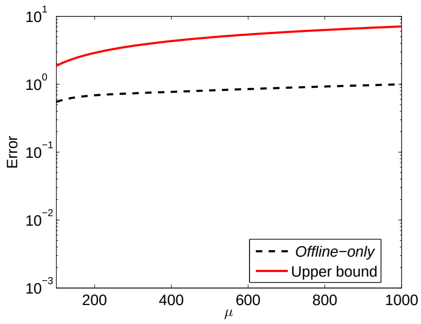
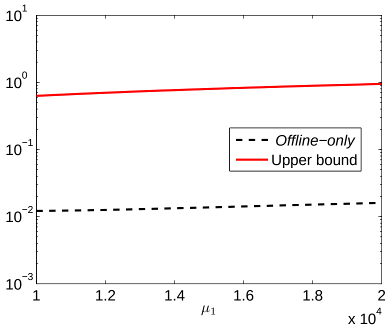
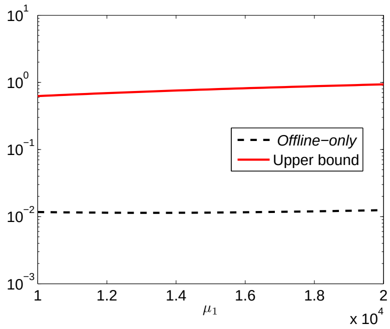


Figure 19: First test case. Upper bound (58) compared to the true error

We can see that the bound is sharp in the first test case, while in the Graetz problem the bound tends to overestimate the real error by two orders of magnitude.



(a) $\mu_2 = 2.5$



(b) $\mu_2 = 3.5$

Figure 20: Poiseuille-Graetz test case. Upper bound (58) compared to the true error.

The reasonable sharpness of the error bound obtained for at least one test case leads us to state that the *Offline-only stabilized* approach is not a good approximation method.

Even if the *Offline-only stabilized* is strongly consistent with respect to the continuous problem, we have to note that we actually have a "consistency" problem caused by the use of different bilinear forms in the two stages. One problem is that the Offline stage, as described in [36, 40], is tailored to minimize the error between the *Offline-Online stabilized* solution and the stabilized FE one. This implies that, during the Offline stage, we are not controlling the approximation error of the *Offline-only stabilized* solution, causing the "inconsistency", quantified by the streamline derivative term in (57) and (58).

3.3.1. Effectivity of the a posteriori error estimator for the Offline-Online stabilized method

In order to further assess the performances of the *Offline-Online stabilized* method, we want to study the effectivity of the error estimators for the Online stage.

We recall that the *Offline-Online stabilized* method is a particular case of the RB method for elliptic problems, in which we consider as bilinear form the stabilized one. So we can exploit the already existing theory for the *a posteriori* estimation of the error. We can therefore define the error estimator $\boldsymbol{\mu} \mapsto \Delta_{N,stab}(\boldsymbol{\mu})$ such that

$$|||u_N^s(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu},stab} \leq \Delta_{N,stab}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{59}$$

Following [36, 40], we define the *effectivity* of the error estimator $\Delta_{N,stab}$ as:

$$\eta_{N,stab}(\boldsymbol{\mu}) = \frac{\Delta_{N,stab}(\boldsymbol{\mu})}{|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu},stab}} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{60}$$

It can be proven [40] that:

$$1 \leq \eta_{N,stab}(\boldsymbol{\mu}) \leq \sqrt{\frac{\gamma_{stab}(\boldsymbol{\mu})}{\alpha_{LB,stab}^{\mathcal{N}}(\boldsymbol{\mu})}} \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{61}$$

where γ_{stab} and $\alpha_{LB,stab}^{\mathcal{N}}$ are the continuity constant and the lower bound of the coercivity constant, respectively, of the form a_{stab}.

In our numerical tests, we considered a set $\Xi_{test} \subset \mathcal{D}$ of about 100 elements and then we computed the average efficiency

$$\eta_{N,stab}^{av} = \frac{1}{|\Xi_{test}|} \sum_{\boldsymbol{\mu} \in \Xi_{test}} \eta_{N,stab}(\boldsymbol{\mu}). \tag{62}$$

In Table 3 we show the results obtained for both the first test case and the Poiseuille-Graetz problem. We can observe that the effectivity is small and this means that the error estimator does not provide unnecessary overly conservative estimates.

Test case	$\vert\Xi_{test}\vert$	$\eta^{av}_{N,stab}$
First test case	101	8.37
Poiseuille-Graetz	126	6.59

Table 3: First test case and Poiseuille-Graetz test case. Average effectivities in numerical tests

4. Stabilized reduced basis: higher order polynomial approximation

In the previous Section our aim was to study a good stabilization strategy for the RB method. It turned out that the *Offline-Online* method seems to be a good choice.

In this Section we want to test our stabilization method also for higher order polynomial approximation spaces, i.e. piecewise quadratic polynomials. To do so, we introduce a different test problem, also used in [3]. Let Ω be the unit square in $\mathbb{R}^2$, as sketched in Figure 21, and let us define $\boldsymbol{\mu} = (\mu_1, \mu_2)$, where $\mu_1, \mu_2 \in \mathbb{R}$. The problem is the following one:

$$\begin{cases} -\dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + (\cos\mu_2, \sin\mu_2)\cdot\nabla u(\boldsymbol{\mu}) = 0 & \text{in } \Omega \\ u(\boldsymbol{\mu}) = 1 & \text{on } \Gamma_1\cup\Gamma_2 \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_3\cup\Gamma_4\cup\Gamma_5. \end{cases} \tag{63}$$

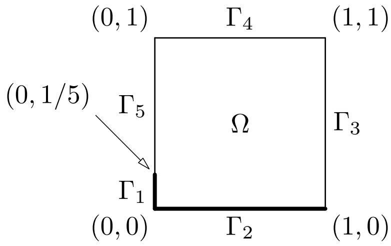


Figure 21: Higher order polynomial approximation. Domain of the problem (63). On the bold sides we impose $u = 1$, while on the other ones $u = 0$.

Let us note that μ_1 represents the Péclet number of the advection-diffusion problem, while μ_2 is the angle between the x axis and the direction of the constant advection field. The bilinear form associated to the problem is:

$$a(w, v; \boldsymbol{\mu}) = \int_\Omega \frac{1}{\mu_1}\nabla w\cdot\nabla v + (\cos\mu_2\,\partial_x w + \sin\mu_2\,\partial_y w)v. \tag{64}$$

We introduce again a triangulation $\mathcal{T}_h$ on the domain Ω and we consider $\mathbb{P}^r(\mathcal{T}_h)$, that is the piecewise polynomial interpolation space of order r ($r = 1, 2$). Now we can define, for $r = 1, 2$, our stabilization term:

$$
\begin{aligned}
s^r(w_h, v_h; \boldsymbol{\mu}) = & -\left(\sum_{K\in\mathcal{T}_h} \delta_K^r \int_K \frac{1}{\mu_1}\Delta w_h \; (\cos\mu_2, \sin\mu_2)\cdot\nabla v_h\right)^{r-1} \\
& + \sum_{K\in\mathcal{T}_h} \delta_K^r \int_K (\cos\mu_2, \sin\mu_2)\cdot\nabla w_h \; (\cos\mu_2, \sin\mu_2)\cdot\nabla v_h,
\end{aligned}
\tag{65}
$$

in which the value of the weights δ_K^r is to be assigned.

Acting as we did in Section 3, we define $g_h^r \in \mathbb{P}^r(\mathcal{T}_h)$ a lifting of the boundary conditions and then we can obtain our final FE approximation problem:

$$
\begin{aligned}
& \text{find } u^{s\,\mathcal{N}_r}(\boldsymbol{\mu}) \in X^{\mathcal{N}_r} \text{ s.t.} \\
& a_{stab}^r(u^{s\,\mathcal{N}_r}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = F_{stab}^r(v^{\mathcal{N}_r}; \boldsymbol{\mu}) \quad \forall v \in X^{\mathcal{N}_r},
\end{aligned}
\tag{66}
$$

where $X^{\mathcal{N}_r}$, a_{stab}^r and F_{stab}^r are defined as in (34) and (36) (the only difference is that now there is the dependency on the polynomial degree r).

As regards the weights δ_K^r, we made different choice for the two different polynomial order. As we saw in Section 3, if $r = 1$ we do not have any restriction on the weights, so we choose

$$
\delta_K^1 = 1 \quad \forall K \in \mathcal{T}_h. \tag{67}
$$

On the contrary, if $r = 2$, we recall that the weights δ_K^2 have to be sufficiently small to guarantee the stability with respect to the SUPG norm (37), as shown in [10, 38]:

$$
a(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) + s(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) \geq \frac{1}{2}\|v^{\mathcal{N}}\|_{SUPG,\boldsymbol{\mu}}, \tag{68}
$$

that is

$$
|||v^{\mathcal{N}}|||_{\boldsymbol{\mu},stab} \geq \frac{1}{2}\|v^{\mathcal{N}}\|_{SUPG,\boldsymbol{\mu}} \tag{69}
$$

for all $v^{\mathcal{N}} \in X^{\mathcal{N}}$.

In particular, to set properly the weights, we follow the choice proposed in [10]. First of all we need to slightly redefine the "element size" h_K, as suggested in [15]:

$$
h_K^2 = \frac{4A_K}{\sqrt{3\sum_{i=1}^3 |\boldsymbol{x}_{i,K} - \boldsymbol{x}_{c,K}|^2}} \quad \forall K \in \mathcal{T}_h, \tag{70}
$$

where, for each element $K \in \mathcal{T}_h$, A_K is the area, $\boldsymbol{x}_{c,K}$ is the barycentre and $\boldsymbol{x}_{i,K}$, for $i = 1, 2, 3$, is the i-th vertex. We also redefine, for any element $K \in \mathcal{T}_h$, the local Péclet number for the $\mathbb{P}^2$-FE approximation as:

$$
\mathbb{P}\mathsf{e}_K(\boldsymbol{\mu}) = \frac{|\boldsymbol{\beta}(\boldsymbol{\mu})| h_K}{C_2\,\varepsilon(\boldsymbol{\mu})} \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{71}
$$

where C_2 is the constant of the inverse inequality

$$
\sum_{K\in\mathcal{T}_h} h_K^2 \int_K |\Delta v^{\mathcal{N}}|^2 \leq C_2 \|\nabla v^{\mathcal{N}}\|_{L^2(\Omega)}^2 \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \subset \mathbb{P}^2(\mathcal{T}_h). \tag{72}
$$

It can be proven that, by defining the element size as in (70), the best value for the constant C_2 is 48 [15]. Finally, we set:

$$
\delta_K^2 = \frac{1}{2} \quad \forall K \in \mathcal{T}_h. \tag{73}
$$

r	$\mu_1 \in$	$\mu_2 \in$	$T_{\text{off}}\,(s)$	SCM iter	N	$\eta^{av}_{N,stab}$	$T_{\text{On}}(s)$
1	$\{10^4\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	460	5	20	18.24	$1.74 \cdot 10^{-3}$
1	$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	623	7	21	53.61	$1.75 \cdot 10^{-3}$
1	$[10^4, 10^5]$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	688	8	36	75.48	$1.85 \cdot 10^{-3}$
1	$[10^4, 10^5]$	$\{\frac{\pi}{4}\}$	138	2	3	31.30	$1.49 \cdot 10^{-3}$
2	$\{10^4\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	856	7	28	74.94	$1.94 \cdot 10^{-3}$
2	$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	3856	19	29	172.44	$2.01 \cdot 10^{-3}$
2	$[10^4, 10^5]$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	3067	28	66	154.63	$9.03 \cdot 10^{-3}$
2	$[10^4, 10^5]$	$\{\frac{\pi}{4}\}$	277	2	4	58.07	$1.74 \cdot 10^{-3}$

Table 4: Higher order polynomial approximation. Numerical tests

We did different choices also for the tolerance ε^*_{tol} of the greedy algorithm. We recall, at first, that for stabilized $\mathbb{P}^1$-FE approximation the error scales as $h^{\frac{3}{2}}$, whereas for stabilized $\mathbb{P}^2$-FE it scales as $h^{\frac{5}{2}}$ [10, 38]. Here h stands for the mesh size

$$h = \max_{K \in \mathcal{T}_h} h_K \tag{74}$$

using, as h_K, either the element diameter or the quantity defined in (70) depending on the polynomial order of the approximation we are using ($\mathbb{P}^1$-FE and $\mathbb{P}^2$-FE, respectively). We note then that the total error between the exact (continuous) solution and the stabilized RB one is

$$\begin{aligned} \|u(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu})\|_{SUPG,\boldsymbol{\mu}} &\leq \|u(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})\|_{SUPG,\boldsymbol{\mu}} + 2\,|||u^{s\,\mathcal{N}}(\boldsymbol{\mu}) - u^s_N(\boldsymbol{\mu})|||_{\boldsymbol{\mu},stab} \\ &\leq C(u(\boldsymbol{\mu}),\boldsymbol{\mu})h^{k+\frac{1}{2}} + 2\,\varepsilon^*_{tol}. \end{aligned} \tag{75}$$

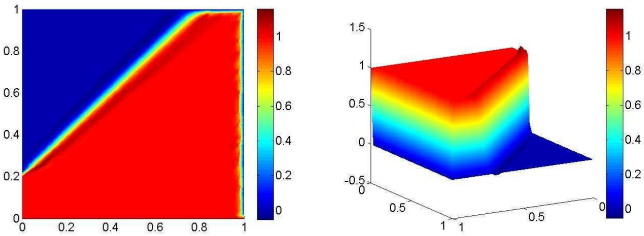
Therefore we choose a value for the tolerance ε^*_{tol} of the same order of magnitude as the FE approximation error. As in our numerical experiments we chose a mesh size $h = 0.03$ for the $\mathbb{P}^1$ approximation and $h = 0.06$ for the $\mathbb{P}^2$ one, ε^*_{tol} was set equal to 10^{-3} and 10^{-4} respectively. In both cases, the *truth* space dimension is $\mathcal{N} = 2605$.

In Table 4 we report some data about the tests we have performed, such as the Offline computational time, T_{off}, and the average effectivity of the *a posteriori* error estimator, $\eta^{av}_{N,stab}$, computed on a set Ξ_{test} with 100 elements.

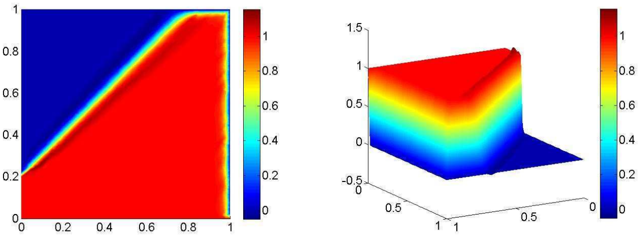
Remark 4.1. We observe that, when building the reduced basis, the variations of the advection field direction have more influence on the dimension N of the reduced basis than the variations of the Péclet number. Indeed, if we keep the advection field constant and we let vary only the Péclet number, the solution shows only variations of the "thickness" of the layers, whose position inside the domain does not change. On the contrary, if we let the advection field change, the position of the internal layer changes too. The reduced basis has then to be able to capture a parameter dependent layer and so, recalling that the reduced solution is a linear combination of the reduced basis functions, it is reasonable that the dimension N increases.

We note also that the Offline time is often much higher for the $\mathbb{P}^2$ approximation. This is because of the SCM, which needs more iterations in the $\mathbb{P}^2$ case than in the $\mathbb{P}^1$ one to provide stability factors approximations [18].

In Figure 22 and 23 we show a visual comparison among the tested methods. The RB approximations shown in this figures are obtained using a parameter space $\mathcal{D} = \{10^5\} \times [\frac{\pi}{6}, \frac{\pi}{3}]$. We note that the internal layer, whose position in the domain is parameter dependent, is well described in the RB solution.

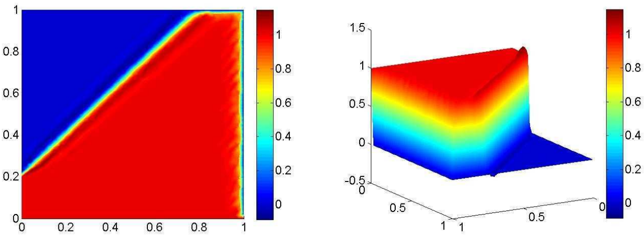


(a) $\mathbb{P}^1$-RB solution.

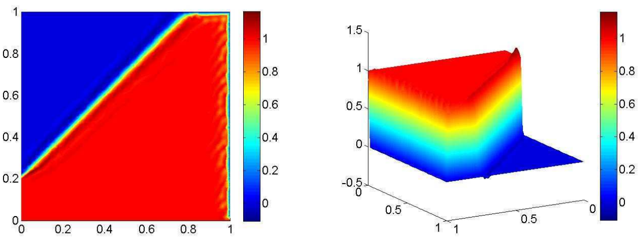


(b) $\mathbb{P}^1$-FE solution

Figure 22: $\mathbb{P}^1$-RB and $\mathbb{P}^1$-FE approximated solution of (63), with $\boldsymbol{\mu} = (10^5, \frac{\pi}{4})$ and mesh size $h = 0.03$ ($\mathcal{N} = 2605$).



(a) $\mathbb{P}^2$-RB solution



(b) $\mathbb{P}^2$-FE solution

Figure 23: $\mathbb{P}^2$-RB and $\mathbb{P}^2$-FE approximated solution of (63), with $\boldsymbol{\mu} = (10^5, \frac{\pi}{4})$ and mesh size $h = 0.06$ ($\mathcal{N} = 2605$).

5. Stabilized reduced basis method for time-dependent problems

In this section we want to apply to time dependent advection-diffusion problems the stabilized RB method introduced for steady problems in the previous sections. The RB method for time dependent problem has been already studied in several works, e.g. [11, 36, 39, 42], but, as regards the advection diffusion equations, we can only find applications with low Péclet number. In this work we are going to test a method that can be effectively applied to advection-diffusion problems with high Péclet number.

In Section 5.1 we introduce the general RB setting for parabolic problems, while in Section 5.2 we briefly recall the SUPG stabilization method for parabolic problems [2, 3, 23]. Finally, in Section 5.3, we show and discuss some numerical tests.

5.1. Reduced basis method for linear parabolic equations

As in Section 2.1, we define the *parameter domain* $\mathcal{D}$ as a closed subset of $\mathbb{R}^P$ and we call $\boldsymbol{\mu}$ any general P-tuple belonging to $\mathcal{D}$. Again, let Ω be a bounded open subset of $\mathbb{R}^d$ $(d = 1, 2, 3)$ with regular boundary $\partial\Omega$ and let X be a functional space such that $H^1_0(\Omega) \subset X \subset H^1(\Omega)$. For each admissible value of the parameter, i.e. for each $\boldsymbol{\mu} \in \mathcal{D}$, we define the continuous bilinear forms

$$\begin{aligned} &a(\cdot,\cdot;\boldsymbol{\mu})\colon X \times X \to \mathbb{R}, \\ &m(\cdot,\cdot;\boldsymbol{\mu})\colon L^2(\Omega) \times L^2(\Omega) \to \mathbb{R}. \end{aligned} \tag{76}$$

We suppose that the form a satisfies the coercivity and *affinity* assumptions (2) and (4), respectively. We assume also that the *mass* form m satisfies an *affinity* assumption like the following one:

$$m(v, w;\boldsymbol{\mu}) = \sum_{q=1}^{Q_m} \Theta^m_q(\boldsymbol{\mu}) m^q(v, w) \tag{77}$$

where, like in (4), $\Theta^m_q\colon \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_m$, are smooth functions whereas $m^q\colon L^2(\Omega) \times L^2(\Omega) \to \mathbb{R}$, $q = 1, \dots, Q_m$, are continuous $\boldsymbol{\mu}$-independent bilinear forms. Finally, for each $\boldsymbol{\mu} \in \mathcal{D}$, we define the right-hand side continuous linear form $F(\cdot;\boldsymbol{\mu})\colon X \to \mathbb{R}$ which satisfies the *affine* assumption (5). Let us finally denote our time domain with $I = [0, T]$, where T is the final time.

We can now define our continuous problem:

$$\begin{aligned} &\text{find } u(\cdot;\boldsymbol{\mu}) \in C^0(I; L^2(\Omega)) \cap L^2(I; X) \text{ s.t.} \\ &m(\partial_t u(t;\boldsymbol{\mu}), v) + a(u(t;\boldsymbol{\mu}), v;\boldsymbol{\mu}) = g(t)F(v;\boldsymbol{\mu}) \quad \forall v \in X, \quad \forall t \in I \\ &\text{given the initial value } u(0;\boldsymbol{\mu}) = u_0 \in L^2(\Omega). \end{aligned} \tag{78}$$

where $g\colon I \to R$ is a *control function* such that $g \in L^2(I)$. We want now to explain in which situations we need such a *control function.* Let us suppose that the problem we are trying to approximate is of the form:

$$\begin{cases} \partial_t u(\boldsymbol{\mu}) + Lu(\boldsymbol{\mu}) = h & \text{in } \Omega \\ u(\cdot, t;\boldsymbol{\mu}) = 0 & \text{on } \partial\Omega,\ \forall t \in I \\ & + \quad \text{initial conditions} \end{cases} \tag{79}$$

where L is a differential operator and $h \in L^2(\Omega \times I)$. If we suppose that $h(\boldsymbol{x}, t) = g(t)f(\boldsymbol{x})$ for each $(\boldsymbol{x}, t) \in \Omega \times I$, with $g \in L^2(I)$ and $f \in L^2(\Omega)$, we obtain a weak formulation like (78). Another situation can be the following one:

$$\begin{cases} \partial_t u(\boldsymbol{\mu}) + Lu(\boldsymbol{\mu}) = 0 & \text{in } \Omega \\ u(\cdot, t;\boldsymbol{\mu}) = h(\cdot, t) & \text{on } \partial\Omega,\ \forall t \in I \\ & + \quad \text{initial conditions} \end{cases} \tag{80}$$

in which L is again a differential operator, while h is a sufficiently regular function defined on the boundary $\partial\Omega$. We can assume for example that $h(\boldsymbol{x}, t) = g(t)f(\boldsymbol{x})$ for each $(\boldsymbol{x}, t) \in \Omega \times I$, with $g \in L^2(I)$ and $f \in H^{\frac{1}{2}}(\partial\Omega)$. Denoting with $\tilde{f}$ a $H^1(\Omega)$ lifting of the boundary datum f, we obtain a weak formulation like (78) in which the functional F is given by:

$$F(v; \boldsymbol{\mu}) = -a(\tilde{f}, v; \boldsymbol{\mu}) \quad \forall v \in X, \tag{81}$$

where a is the bilinear form associated with the differential operator L.

5.1.1. Discretization and RB formulation

To discretize the time-dependent problem (78) we follow the approach used in [13, 32, 36] that is to use finite differences in time and FE in space discretization [38].

We start by discretizing the spatial part of the problem. We thus define the FE *truth* approximation space $X^{\mathcal{N}}$ and we denote its basis with $\{\varphi_i\}_{i=1}^{\mathcal{N}}$. The *semi-discretized* problem reads as

$$\begin{aligned}
&\text{for each } t \in I, \text{ find } u^{\mathcal{N}}(t; \boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.}\\
&m(\partial_t u^{\mathcal{N}}(t; \boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a(u^{\mathcal{N}}(t; \boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g(t)F(v^{\mathcal{N}}; \boldsymbol{\mu}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}},\\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.}\\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.
\end{aligned} \tag{82}$$

To obtain a fully discretized problem, we subdivide the time interval I into J subintervals of length $\Delta t = T/J$ and we define $t^j = j\Delta t$, $j = 1, \ldots, J$. We then replace the time derivative in (82) with a backward finite difference approximation. The fully discretized problem we are considering is:

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.}\\
&\frac{1}{\Delta t} m(u^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a(u^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g(t^j)F(v^{\mathcal{N}}; \boldsymbol{\mu}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}},\\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.}\\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.
\end{aligned} \tag{83}$$

We denote with $\boldsymbol{u}(\boldsymbol{\mu})$ the solution array, that is:

$$\boldsymbol{u}^{\mathcal{N}}(\boldsymbol{\mu}) = (u^{\mathcal{N}\,1}(\boldsymbol{\mu}), \ldots, u^{\mathcal{N}\,J}(\boldsymbol{\mu})) \in \left(X^{\mathcal{N}}\right)^J. \tag{84}$$

The latter problem is the *Backward Euler-Galerkin* discretization of (78). Of course, this is not the only way to discretize the time-dependent problem (78), for example we can resort to other theta-methods (e.g. Crank-Nicholson) or to higher order methods [38].

The RB formulation of the problem is based on a RB space whose basis functions are built by properly combining *snapshots* in time and space. More precisely, we construct the reduced basis in the time-dependent case following the so called *POD-greedy* approach [14, 31, 32, 36]. It consists in using a greedy technique to explore the parameter space $\mathcal{D}$ and the Proper Orthogonal Decomposition (POD) method to deal with the time evolution. For the *a posteriori* error estimates, we follow the choice presented in [13], but other possibilities have recently been proposed [44, 45].

The RB problem is then:

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}} \text{ s.t.}\\
&\frac{1}{\Delta t} m(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u_N^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) + a(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = g(t^j)F(v_N) \quad \forall v_N \in X_N^{\mathcal{N}},\\
&\text{given the initial condition } u_N^{\mathcal{N}\,0} \text{ s.t.}\\
&(u_N^{\mathcal{N}\,0}, v_N)_{L^2(\Omega)} = (u^{\mathcal{N}\,0}, v_N)_{L^2(\Omega)} \quad \forall v_N \in X_N^{\mathcal{N}}.
\end{aligned} \tag{85}$$

Again, as in (84), we define

$$\boldsymbol{u}_N^{\mathcal{N}}(\boldsymbol{\mu}) = (u_N^{\mathcal{N}\,1}(\boldsymbol{\mu}), \dots, u_N^{\mathcal{N}\,J}(\boldsymbol{\mu})) \in \left(X^{\mathcal{N}}\right)^J. \tag{86}$$

that is the RB solution of the time dependent problem.

5.2. SUPG stabilization method for time dependent problems

In this section we briefly introduce the SUPG method for time-dependent problems [2, 3, 23]. The idea is the same of the steady case: we add a stabilization term to the bilinear form in order to improve the stability. The stabilization term is almost the same than in the steady case, but now we have to consider also the time dependency to guarantee the strong consistency. We thus set

$$s(v^{\mathcal{N}}(t), w^{\mathcal{N}}; \boldsymbol{\mu}) = \sum_{K\in\mathcal{T}_h} \delta_K \left(\partial_t v^{\mathcal{N}}(t) + L^{\boldsymbol{\mu}} v^{\mathcal{N}}(t), \frac{h_K}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L^{\boldsymbol{\mu}}_{SS} w^{\mathcal{N}}\right)_K \tag{87}$$

where $v^{\mathcal{N}}(t) \in X^{\mathcal{N}}$ for each $t \in I$ and $w^{\mathcal{N}} \in X^{\mathcal{N}}$. Here $L^{\boldsymbol{\mu}}$ is the steady advection-diffusion operator and $L^{\boldsymbol{\mu}}_{SS}$ its skew-symmetric part (22).

We note that if either the coefficients of the equation or its domain are $\boldsymbol{\mu}$-dependent, then the stabilization terms will depend on $\boldsymbol{\mu}$ too, as we have actually shown in Section 3.

Assuming the parametric dependence, we can write the *Backward Euler-SUPG* formulation as follows:

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.}\\
&\frac{1}{\Delta t} m_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g(t^j) F_{stab}(v^{\mathcal{N}})\\
&\quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}},\\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.}\\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.
\end{aligned} \tag{88}$$

in which m_{stab}, a_{stab} and F_{stab} are

$$\begin{aligned}
m_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= m(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) + \sum_{K_o(\boldsymbol{\mu})\in\mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \left(v^{\mathcal{N}}, \frac{h_{K_o(\boldsymbol{\mu})}}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L^{\boldsymbol{\mu}}_{SS}\, w^{\mathcal{N}}\right)_{K_o(\boldsymbol{\mu})}\\
a_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= a(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) + \sum_{K_o(\boldsymbol{\mu})\in\mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \left(L^{\boldsymbol{\mu}} v^{\mathcal{N}}, \frac{h_{K_o(\boldsymbol{\mu})}}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L^{\boldsymbol{\mu}}_{SS}\, w^{\mathcal{N}}\right)_{K_o(\boldsymbol{\mu})}\\
F_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= F(v^{\mathcal{N}}; \boldsymbol{\mu}) + \sum_{K_o(\boldsymbol{\mu})\in\mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \left(f, \frac{h_{K_o(\boldsymbol{\mu})}}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L^{\boldsymbol{\mu}}_{SS}\, w^{\mathcal{N}}\right)_{K_o(\boldsymbol{\mu})}
\end{aligned} \tag{89}$$

where $K_o(\boldsymbol{\mu})$ are the elements which form the mesh $\mathcal{T}_{h,o}$ defined on the original domain Ω_o and f can be a source term of the advection-diffusion equation or a lifting of the Dirichlet boundary data.

For the analysis of stability and convergence of this method, we refer to [2, 4, 21].

5.3. Numerical results

We are showing now some numerical tests of the stabilized RB method for parabolic PDEs. The first one, discussed in Section 5.3.1 is the time dependent version of the problem studied in Section 4, while the second test case, Section 5.3.2, is a time-dependent Poiseuille-Graetz problem.

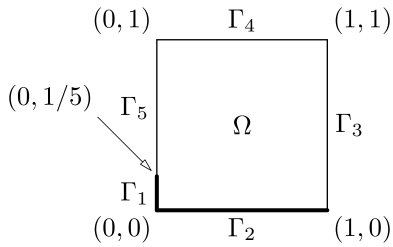


Figure 24: First time dependent test case. Domain of the problem (90). On the bold sides we impose $u = g$, while on the other ones $u = 0$.

5.3.1. A first time dependent test case

Let us denote with Ω the unit square in $\mathbb{R}^2$, and let us subdivide its boundary into five parts Γ_i, $i = 1, \ldots, 5$, as sketched in Figure 24. Moreover, let us denote with I the time interval $[0, T]$.

Finally, let us define $\boldsymbol{\mu} = (\mu_1, \mu_2)$, with $\mu_1, \mu_2 \in \mathbb{R}$. The problem we are dealing with is the following:

$$\begin{cases} \partial_t u - \dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + (\cos\mu_2, \sin\mu_2) \cdot \nabla u(\boldsymbol{\mu}) = 0 & \text{in } \Omega \times I \\ u(\cdot, t; \boldsymbol{\mu}) = g(t) & \text{on } \Gamma_1 \cup \Gamma_2,\ \forall t \in I \\ u(\cdot, t; \boldsymbol{\mu}) = 0 & \text{on } \Gamma_3 \cup \Gamma_4 \cup \Gamma_5,\ \forall t \in I, \\ u(\cdot, 0; \boldsymbol{\mu}) = 0 & \text{on } \Omega, \end{cases} \tag{90}$$

where g is a control function.

To build our approximation procedure, we first define a triangulation $\mathcal{T}_h$, with which we can define the polynomial approximation space $\mathbb{P}^1(\mathcal{T}_h)$ (see (28)). More precisely, we define $X^{\mathcal{N}} = \mathbb{P}^1(\mathcal{T}_h) \cap H_0^1(\Omega)$. We can thus obtain the stabilized FE formulation (88) in which, for all $v^{\mathcal{N}}, w^{\mathcal{N}} \in X^{\mathcal{N}}$, we have:

$$\begin{aligned} m_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= \int_\Omega v^{\mathcal{N}} w^{\mathcal{N}} + \sum_{K \in \mathcal{T}_h} h_K \left(v^{\mathcal{N}}, (\cos\mu_2, \sin\mu_2) \cdot \nabla w^{\mathcal{N}}\right)_K \\ a_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= \int_\Omega \frac{1}{\mu_1} \nabla v^{\mathcal{N}} \cdot \nabla w^{\mathcal{N}} + (\cos\mu_2, \sin\mu_2) \cdot \nabla v^{\mathcal{N}}\, w^{\mathcal{N}} \\ &\quad + \sum_{K \in \mathcal{T}_h} h_K \left((\cos\mu_2, \sin\mu_2) \cdot \nabla v^{\mathcal{N}}, (\cos\mu_2, \sin\mu_2) \cdot \nabla w^{\mathcal{N}}\right)_K \\ F_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= \sum_{K \in \mathcal{T}_h} h_K \left(f_h, (\cos\mu_2, \sin\mu_2) \cdot \nabla\, w^{\mathcal{N}}\right)_K \end{aligned} \tag{91}$$

where f_h is a lifting function corresponding to the boundary condition $u = 1$ on $\partial\Omega$. We recall that, as we are using piecewise linear polynomials, we are allowed to omit the term containing the laplacian into the stabilization term. It is evident from the previous definitions that we have used a constant weighting $\delta_K = 1 \quad \forall K \in \mathcal{T}_h$.

The computations were performed using $T = 2.5$ and subdividing the time interval into $J = 50$ time-steps. As regards the spatial discretization, we used a mesh with size $h \approx 0.03$. The dimension of the polynomial approximation space is $\mathcal{N} = 2605$. The tolerance on the POD-greedy algorithm is $\varepsilon^*_{tol} = 10^{-2}$ (see [36] for the definition). In table 5 we report informations about the computational time and the average efficiency (62) of the error estimator for the stabilized parabolic problem [36] (computed on a set Ξ_{test} with 100 elements). We note that the variations of the parameter μ_2, that is the direction of the advection field, has stronger effect on the number of reduced basis N than the variations of the Péclet number μ_1, as we observed also in the steady case in Section 4. In Figures 25 and 26 we report some pictures of the RB solutions obtained for $\boldsymbol{\mu} = (10^5, \frac{\pi}{6})$, using the parameter space $\mathcal{D} = [10^4, 10^5] \times [\frac{\pi}{6}, \frac{\pi}{3}]$. More precisely, in Figure 25, we show the RB solution (computed at some time-steps) of (90) obtained using a constant control function $g \equiv 1$. In Figure 26 we show the solution corresponding to the control function $g(t) = \sin^2(\frac{4}{5}\pi t)$, for all $t \in [0, T]$.

$\mu_1 \in$	$\mu_2 \in$	$T_{\text{off}}\,(s)$	N	$T_{\text{on}}\,(s)$	$\eta^{av}_{N,stab}$
$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2346	28	$8.75 \cdot 10^{-2}$	2.35
$[10^4, 10^5]$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2857	69	$8.19 \cdot 10^{-2}$	3.43
$[10^4, 10^5]$	$\{\frac{\pi}{4}\}$	339	15	$8.44 \cdot 10^{-2}$	1.93

Table 5: First time dependent test case. Numerical tests

5.3.2. Time dependent Poiseuille-Graetz problem

In this section we want to test the stabilized RB method for a time dependent Poiseuille-Graetz problem [11, 20, 36, 42]. We have already dealt with the steady case of this problem in Section 3.2.

Let $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathbb{R}^2$ such that $\mu_1, \mu_2 > 0$. For each value of the parameter $\boldsymbol{\mu}$, let $\Omega_o(\boldsymbol{\mu})$ be the rectangle in $\mathbb{R}^2$ sketched in Figure 27. We first subdivide $\Omega_o(\boldsymbol{\mu})$ into two subdomains, $\Omega_{o\,1}(\boldsymbol{\mu})$ and $\Omega_{o\,2}(\boldsymbol{\mu})$, and then we subdivide the boundary $\partial\Omega$ into 6 parts $\Gamma_{o\,i}$, $i = 1, \ldots, 6$. We then define I the time interval $[0, T]$.

The problem is to find the temperature distribution $u(\boldsymbol{\mu})$ such that:

$$\begin{cases} \partial_t u(\boldsymbol{\mu}) - \dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + 4\,y(1-y)\partial_x u(\boldsymbol{\mu}) = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \\ u(\cdot, t; \boldsymbol{\mu}) = g_1(t) & \text{on } \Gamma_{o\,1}(\boldsymbol{\mu}) \cup \Gamma_{o\,2}(\boldsymbol{\mu}) \cup \Gamma_{o\,6}(\boldsymbol{\mu}),\ \forall\, t \in I, \\ u(\cdot, t; \boldsymbol{\mu}) = g_2(t) & \text{on } \Gamma_{o\,3}(\boldsymbol{\mu}) \cup \Gamma_{o\,5}(\boldsymbol{\mu}),\ \forall\, t \in I, \\ \dfrac{\partial u}{\partial \nu}(\cdot, t; \boldsymbol{\mu}) = 0 & \text{on } \Gamma_{o\,4}(\boldsymbol{\mu}),\ \forall\, t \in I, \\ u(\cdot, 0; \boldsymbol{\mu}) = 1 & \text{on } \Omega_o(\boldsymbol{\mu}). \end{cases} \tag{92}$$

where g_1 and g_2 are *control* functions.

Before introducing the FE formulation, we have to set some notation. First of all, we chose a particular $\bar{\boldsymbol{\mu}} \in \mathcal{D}$ and we define the reference domain $\Omega = \Omega_o(\bar{\boldsymbol{\mu}})$. We coherently define the reference subdomains $\Omega_i = \Omega_{o\,i}$, $i = 1, 2$, and the boundary regions $\Gamma_i = \Gamma_{o\,i}(\boldsymbol{\mu})$, $i = 1, \ldots, 6$. The reference domain can be mapped onto the original domain $\Omega_o(\boldsymbol{\mu})$, for each $\boldsymbol{\mu} \in \mathcal{D}$, using the transformation $T(\boldsymbol{\mu})$, introduced in Section 3.2 by defining its restrictions on the subdomains Ω_i, $i = 1, 2$ (see (42) and (43)). Now, we build a triangulation $\mathcal{T}_h^1$ on Ω^1 and a triangulation $\mathcal{T}_h^2$ on Ω^2 such that their union $\mathcal{T}_h$ is a proper triangulation on Ω. We can then define the approximation space $X^{\mathcal{N}} = \mathbb{P}^1(\mathcal{T}_h) \cap H_0^1(\Omega)$.

We define now the lifting of the boundary data, f_h^1 and f_h^2, as functions in $\mathbb{P}^1(\mathcal{T}_h)$ such that:

$$f_h^1|_{\Gamma_3 \cap \Gamma_5} \equiv 1 \qquad f_h^2|_{\Gamma_1 \cap \Gamma_2 \cap \Gamma_6} \equiv 1 \tag{93}$$

Like in Section 3.2, we can write the weak formulation of the problem (92) and then track it back on the reference domain. We can thus obtain the following Backward-Euler/stabilized FE

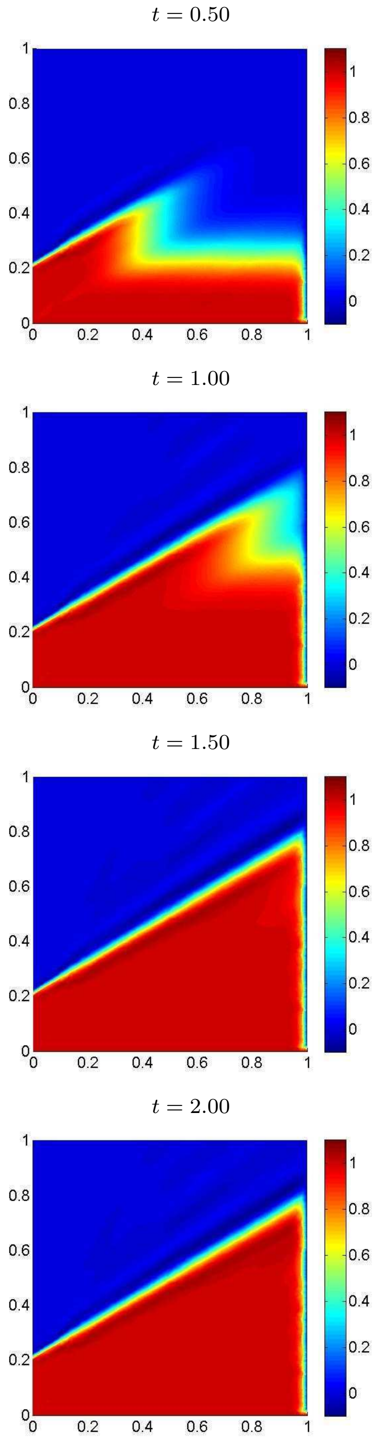


Figure 25: First time dependent test case. RB solution of (90), with $g(t) = 1$ for all $t \in [0, T]$, for a parameter value $\boldsymbol{\mu} = (10^5, \frac{\pi}{6})$.

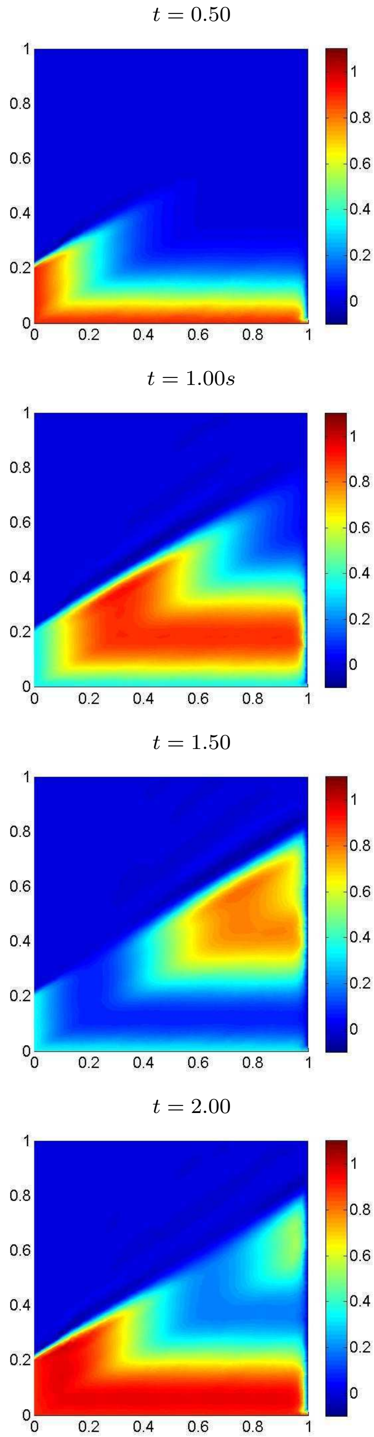


Figure 26: First time dependent test case. RB solution of (90), with $g(t) = \sin^2(\frac{4}{5}\pi t)$ for all $t \in [0, T]$, for a parameter value $\boldsymbol{\mu} = (10^5, \frac{\pi}{6})$.

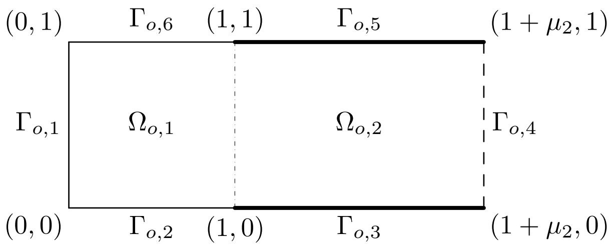


Figure 27: Time dependent Poiseuille-Graetz problem. Domain of the problem (92). On the bold sides we impose $u = g_2$, on the dashed side we impose homogeneous Neumann conditions, on the remaining sides we impose $u = g_1$.

problem:

$$
\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.}\\
&\frac{1}{\Delta t} m_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu})\\
&\quad = g_1(t^j) F^1_{stab}(v^{\mathcal{N}}) + g_2(t^j) F^2_{stab}(v^{\mathcal{N}})\\
&\qquad \forall v^{\mathcal{N}} \in X^{\mathcal{N}},\\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.}\\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}
\end{aligned}
\tag{94}
$$

where, in the left-hand side:

$$
\begin{aligned}
m_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) = &\int_{\Omega^1} v^{\mathcal{N}} w^{\mathcal{N}} + \sum_{K \in \mathcal{T}_h^1} h_K \int_K v^{\mathcal{N}} \partial_x w^{\mathcal{N}}\\
&+ \int_{\Omega^2} \frac{\mu_2}{\mu_1} v^{\mathcal{N}} w^{\mathcal{N}} + \sum_{K \in \mathcal{T}_h^1} \frac{h_K}{\sqrt{\mu_2}} \int_K v^{\mathcal{N}} \partial_x w^{\mathcal{N}}\\
a_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) = &\int_{\Omega^1} \frac{1}{\mu_1} \nabla v^{\mathcal{N}} \cdot \nabla w^{\mathcal{N}} + 4\, y(1-y) \partial_x v^{\mathcal{N}}\, w^{\mathcal{N}}\\
&+ \sum_{K \in \mathcal{T}_h^1} h_K \int_K \left(4\, y(1-y) \partial_x v^{\mathcal{N}}\right) \partial_x w^{\mathcal{N}}\\
&+ \int_{\Omega^2} \frac{1}{\mu_1 \mu_2} \partial_x v^{\mathcal{N}} \partial_y w^{\mathcal{N}} + \frac{\mu_2}{\mu_1} \partial_x v^{\mathcal{N}} \partial_y w^{\mathcal{N}}\\
&\quad + 4\mu_2\, y(1-y) \partial_x v^{\mathcal{N}}\, w^{\mathcal{N}}\\
&+ \sum_{K \in \mathcal{T}_h^2} \frac{h_K}{\sqrt{\mu}_2} \int_K \left(4\, y(1-y) \partial_x w^{\mathcal{N}}\right) \partial_x v^{\mathcal{N}}.
\end{aligned}
\tag{95}
$$

and, concerning the right-hand side, we have:

$$
\begin{aligned}
F^1_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= -\, a_{stab}(f^1_h, v^{\mathcal{N}}; \boldsymbol{\mu})\\
F^2_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= -\, a_{stab}(f^2_h, v^{\mathcal{N}}; \boldsymbol{\mu}),
\end{aligned}
\tag{96}
$$

for all $v^{\mathcal{N}}, w^{\mathcal{N}} \in X^{\mathcal{N}}$. The weighting has been chosen as in Section 3.2.

In order to apply the RB method exposed in Section 5.1, we exploit the linearity of the problem (92) and we consider the two problems:

$$
\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } \varphi^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.}\\
&\frac{1}{\Delta t} m_{stab}(\varphi^{\mathcal{N}\,j}(\boldsymbol{\mu}) - \varphi^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(\varphi^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g_1(t^j) F^1_{stab}(v^{\mathcal{N}})\\
&\qquad \forall v^{\mathcal{N}} \in X^{\mathcal{N}},\\
&\text{given the initial condition } \varphi^{\mathcal{N}\,0} = \theta u^{\mathcal{N}\,0}
\end{aligned}
\tag{97}
$$

and

$$
\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } \psi^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.}\\
&\frac{1}{\Delta t} m_{stab}(\psi^{\mathcal{N}\,j}(\boldsymbol{\mu}) - \psi^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(\psi^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g_2(t^j) F^1_{stab}(v^{\mathcal{N}})\\
&\qquad \forall v^{\mathcal{N}} \in X^{\mathcal{N}},\\
&\text{given the initial condition } \psi^{\mathcal{N}\,0} = (1-\theta) u^{\mathcal{N}\,0},
\end{aligned}
\tag{98}
$$

with $\theta \in [0,1]$ to be set.

Obviously, if $\boldsymbol{\varphi}^{\mathcal{N}}(\boldsymbol{\mu})$ and $\boldsymbol{\psi}^{\mathcal{N}}(\boldsymbol{\mu})$ are solution of (97) and (98), respectively, then $\boldsymbol{\varphi}^{\mathcal{N}}(\boldsymbol{\mu})+\boldsymbol{\psi}^{\mathcal{N}}(\boldsymbol{\mu})$ is a solution of (94).

Once we have this "separation" of the problem, we can apply the RB method to (97) and (98) separately. We then define the RB solution of (94) $u_N^{\mathcal{N}}(\boldsymbol{\mu}) := \varphi_N^{\mathcal{N}}(\boldsymbol{\mu}) + \psi_N^{\mathcal{N}}(\boldsymbol{\mu})$. Concerning the RB approximation error, the triangular inequality implies that:

$$
|||\boldsymbol{u}^{\mathcal{N}}(\boldsymbol{\mu}) - \boldsymbol{u}_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \le |||\boldsymbol{\varphi}^{\mathcal{N}}(\boldsymbol{\mu}) - \boldsymbol{\varphi}_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} + |||\boldsymbol{\psi}^{\mathcal{N}}(\boldsymbol{\mu}) - \boldsymbol{\psi}_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep}, \tag{99}
$$

where:

$$
|||\boldsymbol{v}^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} = \left(m(v^{\mathcal{N}\,J}(\boldsymbol{\mu}), v^{\mathcal{N}\,J}(\boldsymbol{\mu}); \boldsymbol{\mu}) + \sum_{j=1}^{J} a(v^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}\,j}(\boldsymbol{\mu}); \boldsymbol{\mu}) \Delta t \right)^{\frac{1}{2}} \tag{100}
$$

for all sequences $\boldsymbol{v}^{\mathcal{N}}(\boldsymbol{\mu}) = (v^{\mathcal{N}\,1}(\boldsymbol{\mu}), \ldots, v^{\mathcal{N}\,J}(\boldsymbol{\mu})) \in \left(X^{\mathcal{N}}\right)^J$. In our numerical tests we have used $\mathcal{D} = [10000, 20000] \times [0.5, 4]$, $T = 5$, $J = 100$ and $\theta = 1$. The dimension of the FE space is $\mathcal{N} = 1309$ ($h \approx 0.06$). The RB method yields $N^1 = 98$ basis for the problem (97) (Offline computational time: $5773\,s$) and $N^2 = 50$ basis for the problem (98) (Offline computational time: $1658\,s$). The tolerance on the greedy algorithm is $\varepsilon^*_{tol} = 10^{-2}$ (see [36] for the definition).

In Figure 28, we show the RB solution of (94) for $\boldsymbol{\mu} = (15000, 2)$, computed at some time steps. Here we used the following control functions:

$$
\begin{aligned}
g_1(t) &= 1 \quad \forall t \in I,\\
g_2(t) &= e^{-t} \quad \forall t \in I.
\end{aligned}
\tag{101}
$$

The *a posteriori* error estimator give the following result:

$$
|||\boldsymbol{\varphi}^{\mathcal{N}}(\boldsymbol{\mu}) - \boldsymbol{\varphi}_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \le 0.058, \qquad |||\boldsymbol{\psi}^{\mathcal{N}}(\boldsymbol{\mu}) - \boldsymbol{\psi}_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \le 0.047, \tag{102}
$$

then for the total RB approximation error holds

$$
|||\boldsymbol{u}^{\mathcal{N}}(\boldsymbol{\mu}) - \boldsymbol{u}_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \le 0.105. \tag{103}
$$

This error has the same order of magnitude as the time dependent SUPG approximation error, which is bounded by $C(h^3 + \Delta t^2)^{\frac{1}{2}}$ [21]. The computational time of the Online stage is $0.255\,s$. The average effectivity (62), computed on a set Ξ_{test} with 100 elements, is 1.85.

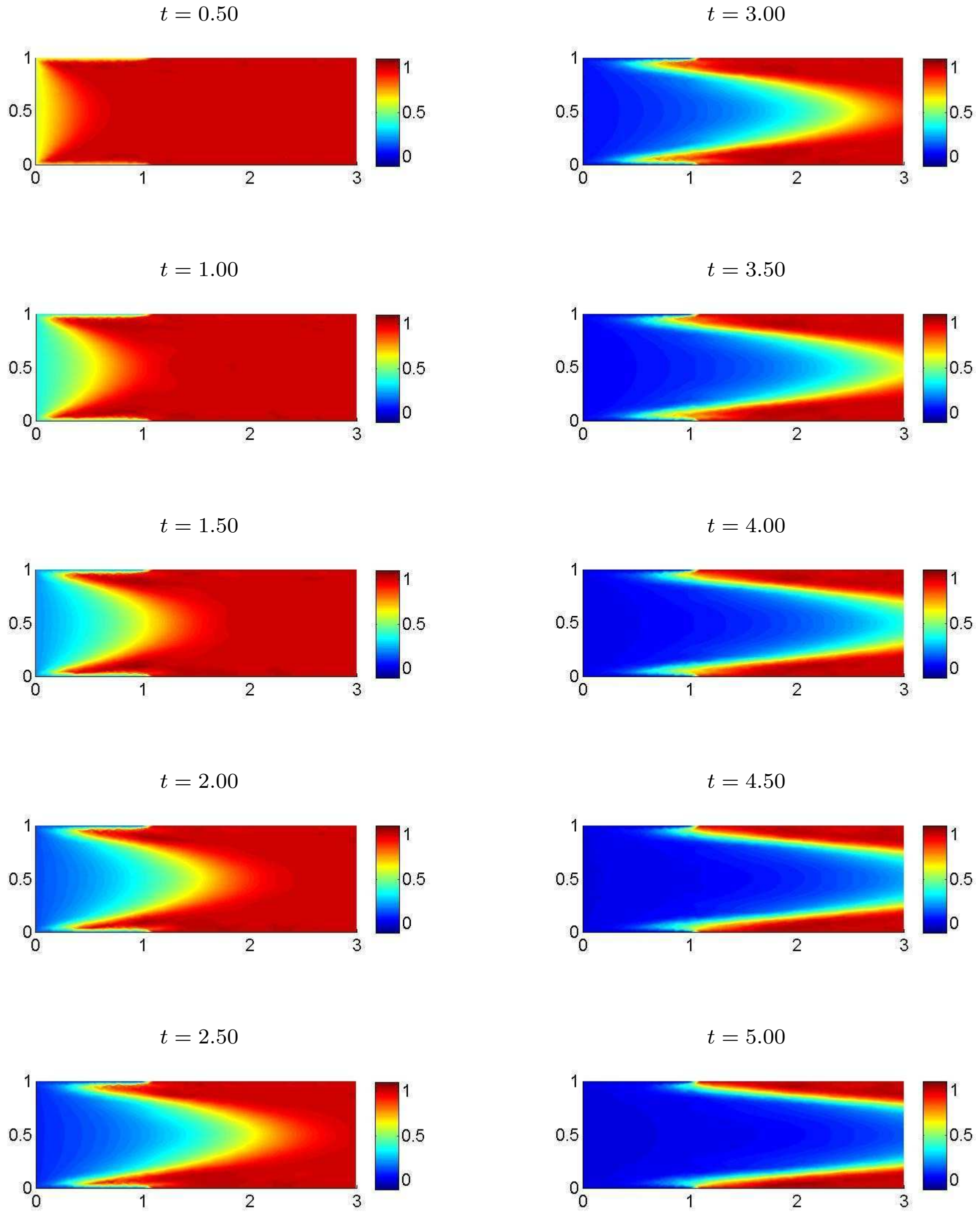


Figure 28: Time dependent Poiseuille-Graetz problem. RB solution computed at some time steps.

6. Conclusions

In this work we have dealt with stabilization techniques for the approximation of the solution of advection dominated problems using the reduced basis approach, in both steady and unsteady case, for high Péclet numbers.

Concerning the steady case, we have carried out a comparison between two possible stabilization techniques, an *Offline-Online* stabilization strategy and an *Offline-only* option. The *Offline-Online* strategy has turned out to be the best choice because it produces stable RB solutions and also the *a posteriori* error estimators discussed in [36, 40] are still effective. As regards the *Offline-only* method, we have observed strong instability phenomena in the RB solution and we have shown that this is because of "inconsistency" problems arising from the use of different bilinear forms in the two stages of the RB method. We have provided also some explanations of the different behaviour of the two stabilization techniques based on physical considerations involving the advection field and the boundary layer.

Having determined which stabilization strategy gives the best results, we have tested it also using the piecewise quadratic FE space as *truth* approximation space, instead of the usual piecewise linear one, obtaining satisfactory results. We performed in particular some numerical test on a problem with steep boundary layers and an internal layer that strongly depend on the direction of the parametric advection field.

In the last part of our work, we have developed a stabilization strategy for the RB approximation of time dependent advection dominated problems. The FE stabilization method - on which our strategy has been based upon - is a time-dependent SUPG method [2, 3]. Considering what we have shown in the steady case, we have proposed to use the same stabilized form in both Offline and Online stage. The method has been successfully tested on some test problems, in particular on an unsteady Poiseuille-Graetz problem with time dependent boundary conditions.

A first natural continuation of this work could be the application of these stabilization strategies to problems with more complex affine geometries, in order to understand if bigger geometrical variations in the shape of the domain can affect negatively the stabilized RB solution. Then, the next step could be to use non-affinely parametrized geometries, which requires an empirical interpolation pre-processing [1, 24], in order to obtain a suitable RB formulation.

Stabilization techniques are also needed in problems dealing with reduced order modelling for advection-diffusion stochastic equations [5].

Possible further developments will be related to stabilization techniques for nonlinear problems, e.g. Navier-Stokes equations to increase Reynolds number.

7. Acknowledgements

P. Pacciarini acknowledges the support provided by the Erasmus Student Placement Project of the University of Pavia, Italy, to allow a research period at École Polytechnique Fédérale de Lausanne, Switzerland, Mathematics Institute of Computational Science and Engineering.

G. Rozza acknowledges the program NOFYSAS, New Opportunities for Young Scientists at SISSA, International School for Advanced Studies, Trieste, Italy.

Both authors wish to thank Prof. Ilaria Perugia, University of Pavia, for her helpful contributions to this work.

The computations in this work have been performed in MATLAB® software [29] using the MLife (finite elements) library [43] and an enhanced version (co-developed at CMCS, EPFL) of the rbMIT© library [17, 33]. These libraries have been extended while carrying out this work, implementing the stabilization methods studied and used.

[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] P.B. Bochev, M.D. Gunzburger, and J.N. Shadid. Stability of the SUPG finite element method for transient advection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 193(23-26):2301–2323, 2004.

[3] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, 1982. FENOMECH '81, Part I (Stuttgart, 1981).

[4] E. Burman and G. Smith. Analysis of the space semi-discretized SUPG method for transient convection-diffusion equations. *Math. Models Methods Appl. Sci.*, 21(10):2049–2068, 2011.

[5] P. Chen, A. Quarteroni, and G. Rozza. Stochastic optimal robin boundary control problems of advection-dominated elliptic equations. To appear in SIAM Journal on Numerical Analysis.

[6] W. Dahmen, C. Plesken, and G. Welper. Double greedy algorithms: reduced basis methods for transport dominated problems. 2013. Submitted.

[7] L. Dedè. *Adaptive and reduced basis approach for optimal control problems in environmental applications.* PhD thesis, Politecnico di Milano, 2008.

[8] L. Dedè. Reduced basis method for parametrized elliptic advection-reaction problems. *J. Comput. Math.*, 28(1):122–148, 2010.

[9] J.L. Eftang, M.A. Grepl, and A.T. Patera. A posteriori error bounds for the empirical interpolation method. *C. R. Math. Acad. Sci. Paris*, 348(9-10):575–579, 2010.

[10] L.P. Franca, S.L. Frey, and T.J.R. Hughes. Stabilized finite element methods. I. Application to the advective-diffusive model. *Comput. Methods Appl. Mech. Engrg.*, 95(2):253–276, 1992.

[11] F. Gelsomino and G. Rozza. Comparison and combination of reduced-order modelling techniques in 3D parametrized heat transfer problems. *Math. Comput. Model. Dyn. Syst.*, 17(4):371–394, 2011.

[12] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[13] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN Math. Model. Numer. Anal.*, 39(1):157–181, 2005.

[14] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *M2AN Math. Model. Numer. Anal.*, 42(2):277–302, 2008.

[15] I. Harari and T.J.R. Hughes. What are C and h?: inequalities for the analysis and design of finite element methods. *Comput. Methods Appl. Mech. Engrg.*, 97(2):157–192, 1992.

[16] T.J.R. Hughes and A. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In *Finite element methods for convection dominated flows*, volume 34 of *AMD*, pages 19–35. Amer. Soc. Mech. Engrs. (ASME), New York, 1979.

[17] D.B.P. Huynh, N.C. Nguyen, A.T. Patera, and G. Rozza. Rapid reliable solution of the parametrized partial differential equations of continuum mechanics and transport. Available at `http://augustine.mit.edu`. Copyright MIT 2008-2011.

[18] 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. Math. Acad. Sci. Paris*, 345(8):473–478, 2007.

[19] L. Iapichino, A. Quarteroni, and G. Rozza. A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. *Comput. Methods Appl. Mech. Engrg.*, 221–222:63–82, 2012.

[20] F.P. Incropera and D.P. DeWitt. *Fundamentals of Heat and Mass Transfer.* John Wiley & Sons, 1990.

[21] V. John and J. Novo. Error analysis of the SUPG finite element discretization of evolutionary convection-diffusion-reaction equations. *SIAM J. Numer. Anal.*, 49(3):1149–1176, 2011.

[22] C. Johnson and U. Nävert. An analysis of some finite element methods for advection-diffusion problems. In *Analytical and numerical approaches to asymptotic problems in analysis (Proc. Conf., Univ. Nijmegen, Nijmegen, 1980)*, volume 47 of *North-Holland Math. Stud.*, pages 99–116. North-Holland, Amsterdam, 1981.

[23] C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. *Comput. Methods Appl. Mech. Engrg.*, 45(1-3):285–312, 1984.

[24] T. Lassila and G. Rozza. Parametric free-form shape design with PDE models and reduced basis method. *Comput. Methods Appl. Mech. Engrg.*, 199(23-24):1583–1592, 2010.

[25] A.E. Løvgren, Y. Maday, and E.M. Rønquist. The reduced basis element method for fluid flows. In *Analysis and simulation of fluid dynamics*, Adv. Math. Fluid Mech., pages 129–154. Birkhäuser, Basel, 2007.

[26] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *Int. J. Numer. Meth. Biomed. Engng.*, 28:604–625, 2012.

[27] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *Int. J. Numer. Meth. Fluids*, 70:646–670, 2012.

[28] A. Manzoni and G. Rozza. Model order reduction by geometrical parametrization for shape optimization in computational fluid dynamics. In J.C.F. Pereira and A. Sequeira, editors, *Proceedings of the V European Conf. Computat. Fluid Dynamics, ECCOMAS CFD 2010, June 14-17, Lisbon, Portugal*, 2010.

[29] MATLAB®. The MathWorks. `http://www.mathworks.com`.

[30] F. Negri, G. Rozza, A. Manzoni, and A. Quarteroni. Reduced basis method for parametrized elliptic optimal control problems. To appear in SIAM Journal on Scientific Computing.

[31] N. C. Nguyen, G. Rozza, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo*, 46:157–185, 2009.

[32] N.C. Nguyen, G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for parametrized parabolic PDEs: application to real-time Bayesian parameter estimation. In *Large-scale inverse problems and quantification of uncertainty*, Wiley Ser. Comput. Stat., pages 151–177. Wiley, Chichester, 2010.

[33] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. Version 1.0, Copyright MIT 2006-2007, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering. Available at `http://augustine.mit.edu`.

[34] A. Quarteroni. *Numerical models for differential problems*, volume 2 of *MS&A. Modeling, Simulation and Applications*. Springer-Verlag Italia, Milan, 2009.

[35] A. Quarteroni, G. Rozza, L. Dedè, and A. Quaini. Numerical approximation of a control problem for advection-diffusion processes. In *System modeling and optimization*, volume 199 of *IFIP Int. Fed. Inf. Process.*, pages 261–273. Springer, New York, 2006.

[36] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *J. Math. Ind.*, 1:1(3), 2011.

[37] A. Quarteroni, G. Rozza, and A. Quaini. Reduced basis methods for optimal control of advection-diffusion problem. In W. Fitzgibbon, R. Hoppe, J. Periaux, O. Pironneau, and Y. Vassilevski, editors, *Advances in Numerical Mathematics*, pages 193–216, Moscow, Russia and Houston, USA, 2007.

[38] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1994.

[39] G. Rozza, D.B.P. Huynh, N.C. Nguyen, and A.T. Patera. Real-time reliable simulation of heat transfer phenomena. In *ASME -American Society of Mechanical Engineers - Heat Transfer Summer Conference Proceedings, HT2009 3 , pages 851-860*, S. Francisco, CA, USA, 2009.

[40] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Methods Eng.*, 15(3):229–275, 2008.

[41] G. Rozza, T. Lassila, and A. Manzoni. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. In J.S. Hesthaven and E. M. RRønquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, volume 76 of *Lecture Notes in Computational Science and Engineering*, pages 307–315. Springer Berlin Heidelberg, 2011.

[42] G. Rozza, N.C. Nguyen, A.T. Patera, and S. Deparis. Reduced basis methods and a posteriori error estimators for heat transfer problems. In *ASME -American Society of Mechanical Engineers - Heat Transfer Summer Conference Proceedings, HT2009 2 , pages 753-762*, S. Francisco, CA, USA, 2009.

[43] F. Saleri, P. Gervasio, G. Rozza, and A. Manzoni. MLife, a MATLAB® library for Finite Elements, tutorial (in progress). MOX, Politecnico di Milano and CMCS, EPFL, 2000-2011. Copyrigth Politecnico di Milano.

[44] K. Urban and A.T. Patera. An improved error bound for reduced basis approximation of linear parabolic problems. To appear in Mathematics of Computation.

[45] K. Urban and A.T. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *C. R. Math. Acad. Sci. Paris*, 350(3-4):203–207, 2012.

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

39/2013 IEVA, F.; MARRA, G.; PAGANONI, A.M.; RADICE, R.
A semiparametric bivariate probit model for joint modeling of outcomes in STEMI patients

40/2013 PACCIARINI P.; ROZZA G.
Stabilized reduced basis method for parametrized advection-diffusion PDEs

38/2013 BONIZZONI, F.; NOBILE, F.
Perturbation analysis for the Darcy problem with log-normal permeability

33/2013 MENAFOGLIO, A; GUADAGNINI, A; SECCHI, P
A Kriging Approach based on Aitchison Geometry for the Characterization of Particle-Size Curves in Heterogeneous Aquifers

34/2013 TAVAKOLI, A.; ANTONIETTI, P.F.; VERANI, M.
Automatic computation of the impermeability of woven fabrics through image processing

35/2013 CATTANEO, L.; FORMAGGIA, L.; IORI G. F.; SCOTTI, A,; ZUNINO, P.
Stabilized extended finite elements for the approximation of saddle point problems with unfitted interfaces

36/2013 FERRAN GARCIA, LUCA BONAVENTURA, MARTA NET, JUAN SANCHEZ
Exponential versus IMEX high-order time integrators for thermal convection in rotating spherical shells

37/2013 LEVER, V.; PORTA, G.; TAMELLINI, L.; RIVA, M.
Characterization of basin-scale systems under mechanical and geochemical compaction

32/2013 TADDEI, T.; PEROTTO, S.; QUARTERONI, A.
Reduced basis techniques for nonlinear conservation laws

31/2013 DASSI, F.; ETTINGER, B.; PEROTTO, S.; SANGALLI, L.M.
A mesh simplification strategy for a spatial regression analysis over the cortical surface of the brain