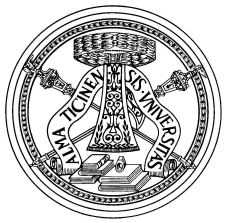
Università degli studi di Pavia
Facoltà di Scienze MM. FF. NN.
Corso di Laurea Magistrale in Matematica



Stabilized reduced basis method for parametrized advection-diffusion PDEs

Relatore:
Prof. Ilaria Perugia

Correlatore:
Dr. Ing. Gianluigi Rozza

Tesi di Laurea Magistrale di
Paolo Pacciarini

Anno Accademico 2011/2012

Ai miei genitori

Contents

Introduction

The aim of this master thesis 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 coefficient 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 [27] or the diffusion of pollutants in the atmosphere [7, 46]. In such applications, we often need very fast evaluations of the approximated solution, depending on some input parameters. This happens, for example, in the case of *real-time* simulation. 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 meets our need for rapidity and it is also able to guarantee the *reliability* of the solution, thanks to sharp *a posteriori* error bounds. We can find in literature many works about the application of the RB method to advection-diffusion problems, e.g. in [15, 45, 51], but they mainly deal with equations in which the Péclet number is low. Some results about the advection dominated case (i.e. high Péclet number) can be found in [7, 8, 44, 46], in which some stabilization techniques are used. The need for stabilization arise from the fact that the finite element (FE) approximated solution - that the RB method aims to recover - shows strong instability problems that have to be fixed.

In this work we want to go further in the study of the stabilization of the RB method for advection dominated problem in both steady and unsteady case. As regards the steady case, we first compare two possible stabilization strategies, by testing them on some test problems, in order to design an efficient *stabilized reduced basis method*. We will then test this method using the piecewise quadratic FE approximation as reference solution, instead of the usual piecewise linear one. Finally, we extend the method designed for the steady case to the time dependent case and we will carry out some numerical tests.

The structure of the work is the following:

Chapter 1 At first we give a brief introduction to parametrized elliptic coercive problems and then we introduce the associated RB approximation method. We will describe in detail this method, with a particular focus on the sampling strategies, the *a posteriori* estimates and the *successive constraint method* for the approximation of the (parametric) coercivity constant.

Chapter 2 After a concise overview of some classical stabilization method for the FE approximation of advection dominated problems, we make a comparison between two possible stabilization strategies for the RB method, by mean of some numerical tests.

Finally we perform some test using the piecewise quadratic FE solution as the reference one.

Chapter 3 We introduce the general RB method for parabolic problems and then we design a suitable stabilization technique, based on a stabilization method for the FE approximation of advection dominated parabolic problems. We finally perform some numerical test to assess the efficiency of the method.

The computations in this work have been performed in MATLAB® software [39] using the MLife (finite elements) library [53] and an enhanced version (co-developed at CMCS, EPFL) of the rbMIT© library [24, 41]. These libraries have been extended while carrying out this work, implementing the stabilization methods we have studied and used.

This thesis has been carried out in the framework of the Erasmus Student Placement project with an internship of 4 months at École Polytechnique Fédérale de Lausanne, Mathematics Institute of Computational Science and Engineering .

Pavia and Lausanne, September 2012

Chapter 1

Reduced basis method for elliptic coercive PDEs

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.

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 and so it turns out to be very useful especially in *real-time* or *many-query* contexts.

There are several options about the type of reduced basis to use. In this work we will focus on Lagrange basis, but it would be possible to choose Taylor basis [42], Hermite basis [28] and *proper orthogonal decomposition* (POD) basis [50]. Moreover, we will use only hierarchical RB spaces [50].

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 solutions (*snapshots*) of the continuous parametrized problem, corresponding to some suitably chosen values of the parameter.

Let us start considering elliptic coercive PPDEs. Denoting with $\boldsymbol{\mu}$ the p-vector parameter, belonging to the parameter space $\mathcal{D} \subset \mathbb{R}^p$, our problem is to find $u(\boldsymbol{\mu})$ in an Hilbert space X such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = F(v; \boldsymbol{\mu}) \quad \forall v \in X \tag{1.1}$$

where $a(\cdot, \cdot; \boldsymbol{\mu})$ and $F(\cdot; \boldsymbol{\mu})$ are a coercive bilinear and a linear form, respectively, depending on the parameter $\boldsymbol{\mu}$. We introduce now the hypothesis that the map $\mathcal{D} \to X$ defined by $\boldsymbol{\mu} \mapsto u(\boldsymbol{\mu})$ is smooth, and so the p-dimensional manifold

$$\mathcal{M} = \{u(\boldsymbol{\mu}) \in X \,|\, \boldsymbol{\mu} \in \mathcal{D}\}. \tag{1.2}$$

turns out to be smooth too. In order to adopt a RB approach, we have to define the underlying high-fidelity *truth* approximation space. To do so, we define $X^{\mathcal{N}}$ as a linear space of finite dimension $\mathcal{N}$, typically very large, in which we will define the *truth* approximated solution $u^{\mathcal{N}}(\boldsymbol{\mu})$. In this work, we will chose as $X^{\mathcal{N}}$ the classical lagrangian FE space and we will use

as *truth* solution the (stabilized) FE one. Other possible choices of *truth* solution can be found in literature, like spectral element [35] and finite volumes [19]. Acting in this way we can consider the "truth" manifold

$$\mathcal{M}^{\mathcal{N}} = \{u^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \,|\, \boldsymbol{\mu} \in \mathcal{D}\} \tag{1.3}$$

where $u^{\mathcal{N}}(\boldsymbol{\mu})$ is the high-fidelity approximation of (1.1). The goal of the RB method will be to provide a low-order approximation of the latter manifold.

The reduced basis method requires the following components [45]

1. Rapidly convergent global approximation by Galerkin projection on a N-dimensional subspace of $X^{\mathcal{N}}$ spanned by solutions of the governing PPDE corresponding to N suitably selected values of the parameter $\boldsymbol{\mu}$. To get a significant reduction of the computational cost, it is crucial that $N \ll \mathcal{N}$.

2. Rigorous and sharp a posteriori error estimators for the error between the RB solution and the "truth" one. This estimation is fundamental for both the certification of the method and the sampling procedure used to build the reduced basis. Moreover we need to require that the computation of these error bounds is inexpensive.

3. Decoupling of the computation in two stages: an expensive Offline stage, to be performed only once, and a very inexpensive Online one, in which is actually performed the *input-output* evaluation.

Intuitively, we can figure the approximation of the *truth* manifold by mean of the Lagrangian RB method as sketched in figure 1.1. The black line is the *truth* manifold in the $\mathcal{N}$-dimensional space X^N. The black dots represent the *snapshot* solutions, which act like Lagrangian interpolation nodes. Finally, the red dashed "interpolant" is our RB approximation, that is built by linear combination of *snapshot* solutions.

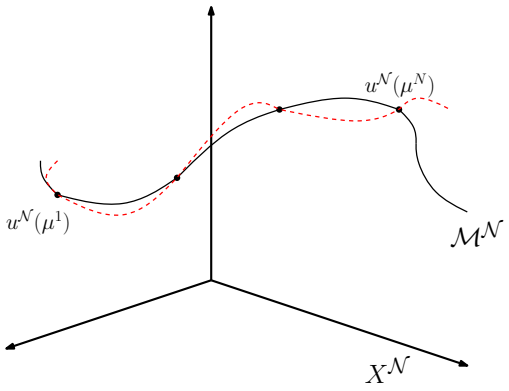


Figure 1.1: Intuitive representation of the *truth* manifold (1.3) (black line) and its RB approximation (red dashed line)

1.1 Elliptic coercive parametrized PDEs

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 consider only second order elliptic PDE, the space X will be such that $H_0^1(\Omega) \subset X \subset H^1(\Omega)$. Formally, our problem can be written as follows:

$$\begin{aligned} &\text{find } u(\mu) \in X \text{ s.t.} \\ &a(u(\boldsymbol{\mu}), v; \mu) = F(v;\boldsymbol{\mu}) \quad \forall v \in X. \end{aligned} \tag{1.4}$$

Let us now define the norms and the inner products we will use. Let a^{sym} be the symmetric part of a. We define:

$$\begin{aligned} ((v,w))_{\boldsymbol{\mu}} &:= a^{sym}(v,w;\boldsymbol{\mu}) \quad \forall v,w \in X \\ |||v|||_{\boldsymbol{\mu}} &:= a^{sym}(v,v;\boldsymbol{\mu})^{\frac{1}{2}} \quad \forall v \in X. \end{aligned} \tag{1.5}$$

The latter forms are of course $\boldsymbol{\mu}$-dependent, but for our purposes we will need norms and inner products that do not depend on the parameter. Thus we choose a particular value of the parameter $\overline{\boldsymbol{\mu}} \in \mathcal{D}$ and we define:

$$\begin{aligned} (v,w)_X &:= ((v,w))_{\overline{\boldsymbol{\mu}}} + \tau\,(v,v)_{L^2(\Omega)} \quad \forall v,w \in X \\ \|v\|_X &:= ((v,v)_X)^{\frac{1}{2}} \quad \forall v \in X \end{aligned} \tag{1.6}$$

with $\tau > 0$. We will further discuss about the choice of τ and $\overline{\boldsymbol{\mu}}$.

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

$$\exists\, \alpha_0 > 0 \quad \text{s.t.} \quad \alpha_0 \le \alpha(\boldsymbol{\mu}) = \inf_{v \in X} \frac{a(v,v;\boldsymbol{\mu})}{\|v\|_X^2} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{1.7}$$

and

$$+\infty > \gamma(\boldsymbol{\mu}) = \sup_{v \in X} \sup_{w \in X} \frac{|a(v,w;\boldsymbol{\mu})|}{\|v\|_X \|w\|_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.8}$$

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:

$$a(v,w;\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(v,w) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.9}$$

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

$$F(v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) F^q(v) \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{1.10}$$

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

Recalling that $X^{\mathcal{N}}$ is a conforming finite element space[1] with $\mathcal{N}$ degrees of freedom, we can now set the *truth* approximation of the problem (1.4):

$$\begin{aligned} &\text{find } u^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\ &a(u^{\mathcal{N}}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = F(v^{\mathcal{N}};\boldsymbol{\mu}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}. \end{aligned} \tag{1.11}$$

[1] *Conforming finite element space* means that $X^{\mathcal{N}} \subset X$.

As we are considering the conforming FE case, conditions similar to (1.7) and (1.8) 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:

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) := \inf_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{a(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu})}{\|v\|_X^2} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{1.12}$$

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

$$\alpha(\boldsymbol{\mu}) \leq \alpha^{\mathcal{N}}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.13}$$

Similarly, for the continuity, we can define

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) := \sup_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \sup_{w^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{|a(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu})|}{\|v^{\mathcal{N}}\|_X \|w^{\mathcal{N}}\|_X} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.14}$$

and it holds that:

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) \leq \gamma(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.15}$$

In this work we will consider as *truth* approximation space $X^{\mathcal{N}}$ a classical finite element space [43].

1.1.1 Geometrical parametrization

An important feature of the the RB method is that it can be used even when the parameter is "geometrical", i.e. the domain of the equation depends on some parameters [26, 34, 36, 37, 38].

As we will see in the next section, to apply the reduced basis method we need a problem like (1.4), in which the forms involved have to be defined on a parameter independent space. To overcome this difficulty, the idea is to assume that the original parametrized domain is the image of a reference parameter-independent domain through a suitable transformation. By doing so, the parametric dependence actually moves from the domain to the coefficients of the equation.

Let us now call *original problem* (subscript o) the one defined on the *original domain* $\Omega_o(\boldsymbol{\mu})$. It reads as follows:

$$\begin{aligned} &\text{find } u_o(\boldsymbol{\mu}) \in X_o(\boldsymbol{\mu}) \text{ s.t.} \\ &a_o(u_o(\boldsymbol{\mu}), v_o; \boldsymbol{\mu}) = F_o(v_o; \boldsymbol{\mu}) \quad \forall v_o \in X_o(\boldsymbol{\mu}) \end{aligned} \tag{1.16}$$

where $X_o(\boldsymbol{\mu})$ is a functional space on $\Omega_o(\boldsymbol{\mu})$, 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 (1.7) and (1.8).

To set the *reference domain* we choose a particular value of the parameter, $\boldsymbol{\mu}_{ref} \in \mathcal{D}$, and define $\Omega = \Omega_o(\boldsymbol{\mu}_{ref})$ as the reference domain. The latter is related to the original domains through a parametric transformation $T(\cdot; \boldsymbol{\mu})$ such that $T(\Omega; \boldsymbol{\mu}) = \Omega_o(\boldsymbol{\mu})$.

We will now focus only on a particular classes of transformations and problems, as it is done in [45, 50]. First of all, for all $\boldsymbol{\mu} \in \mathcal{D}$ we introduce a domain decomposition of $\Omega_o(\boldsymbol{\mu})$ such that:

$$\overline{\Omega_o}(\boldsymbol{\mu}) = \bigcup_{l=1}^{L_{dom}} \overline{\Omega_o^l}(\boldsymbol{\mu}) \tag{1.17}$$

where the subdomains $\Omega_o^l(\boldsymbol{\mu})$, $l = 1, \ldots, L_{dom}$ are mutually non overlapping open subsets of $\Omega_o(\boldsymbol{\mu})$, that is:

$$\Omega_o^l(\boldsymbol{\mu}) \cap \Omega_o^{l'}(\boldsymbol{\mu}) = \emptyset \quad 1 \leq l < l' \leq L_{dom}. \tag{1.18}$$

The need for domain decomposition can arise from modelling reasons, for instance it could happen that the PDE describes a particular application so that different regions of the domain correspond to different materials properties. This can lead to a PDE in which the coefficients show significant discontinuities or the PDE itself can have different form depending on the subdomain. However, a domain decomposition can be set to allow the construction of maps T which guarantees that the forms involved depend "affinely" on the parameter. We will now focus on this second aspect by introducing the piecewise affine transformations.

In order to define the global mapping from the reference domain to the original one, we start by defining the maps between subdomains. For each $\boldsymbol{\mu} \in \mathcal{D}$ we define $T^l(\cdot;\boldsymbol{\mu})\colon \Omega^l \to \Omega_o^l(\boldsymbol{\mu})$, $l = 1, \dots, L_{dom}$, such that:

$$\begin{aligned} &T^l(\overline{\Omega^l};\boldsymbol{\mu}) = \overline{\Omega_o^l}(\boldsymbol{\mu}) \quad 1 \le l \le L_{dom}, \\ &T^l(\boldsymbol{x};\boldsymbol{\mu}) = T^{l'}(\boldsymbol{x};\boldsymbol{\mu}) \quad \forall\, x \in \overline{\Omega^l} \cap \overline{\Omega^{l'}}, \quad 1 \le l < l' \le L_{dom}. \end{aligned} \tag{1.19}$$

We can now define the global mapping $T(\cdot;\boldsymbol{\mu})\colon \Omega \to \Omega_o(\boldsymbol{\mu})$ by gluing together the local maps T^l, that is:

$$T(\boldsymbol{x};\boldsymbol{\mu}) := T^l(\boldsymbol{x};\boldsymbol{\mu}) \quad \forall x \in \overline{\Omega^l} \cap \Omega. \tag{1.20}$$

We assume also that:

i) the maps T^l, $l = 1, \dots, L_{dom}$, are individually bijective and affine;
ii) the map T is continuous.

Each local map T^l can be described by

$$T_i^l(\boldsymbol{x},\boldsymbol{\mu}) = \mathbf{C}_i^l(\boldsymbol{\mu}) + \sum_{j=1}^{d} \mathbf{G}_{ij}^l(\boldsymbol{\mu}) x_j \quad \boldsymbol{x} \in \Omega^l, \quad 1 \le i \le d \tag{1.21}$$

where $\mathbf{C}^l\colon \mathcal{D} \to \mathbb{R}^d$ and $\mathbf{G}^l\colon \mathcal{D} \to \mathbb{R}^{d\times d}$ are smooth maps which associate to each value of the parameter a vector in $\mathbb{R}^d$ and an invertible $d \times d$ matrix, respectively. Roughly speaking, the matrix $\mathbf{G}^l(\boldsymbol{\mu})$ scales and rotates the reference domain, whereas $\mathbf{C}_i^l(\boldsymbol{\mu})$ is a translation vector. For each $\boldsymbol{\mu} \in \mathcal{D}$, we denote with $J^l(\boldsymbol{\mu})$ the determinant of the matrix $\mathbf{G}^l(\boldsymbol{\mu})$. From now on we consider $d = 2$.

1.1.2 Advection-diffusion-reaction operators

After having introduced the geometry transformations, we have now to discuss the choice of the operators. An important class that can be effectively treated within an "affine" framework is the one of advection-diffusion-reaction operators:

$$Lv = \nabla \cdot (\boldsymbol{\nu}(\boldsymbol{\mu})\nabla v) + \boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla v + \gamma(\boldsymbol{\mu}) v \tag{1.22}$$

being $\boldsymbol{\nu}(\boldsymbol{\mu})$ the 2×2 diffusivity tensor, $\boldsymbol{\beta}(\boldsymbol{\mu})$ the advection field in $\mathbb{R}^2$ and $\gamma(\boldsymbol{\mu})$ the reaction coefficient. The bilinear form defined on the original domain associated to (1.22) is, for each $v_o, w_o \in X_o(\boldsymbol{\mu})$

$$a_0(v_o, w_o;\boldsymbol{\mu}) = \sum_{l=1}^{L_{dom}} \int_{\Omega_o^l(\boldsymbol{\mu})} \begin{pmatrix} \frac{\partial v_o}{\partial x_{o\,1}} & \frac{\partial v_o}{\partial x_{o\,2}} & v_o \end{pmatrix} \mathbf{K}_{o,l}(\boldsymbol{\mu}) \begin{pmatrix} \frac{\partial w_o}{\partial x_{o\,1}} \\ \\ \frac{\partial w_o}{\partial x_{o\,2}} \\ \\ w_o \end{pmatrix} \tag{1.23}$$

where $\mathbf{K}_{o,l}\colon \mathcal{D} \to \mathbb{R}^{3\times 3}$, $l = 1, \ldots, L_{dom}$, is a smooth mapping such that, for every $\boldsymbol{\mu} \in \mathcal{D}$, the matrix $\mathbf{K}_{o,l}(\boldsymbol{\mu})$ has the form:

$$\mathbf{K}_{o,l}(\boldsymbol{\mu}) = \left(\begin{array}{c|c} \boldsymbol{\nu}(\boldsymbol{\mu}) & \boldsymbol{\beta}(\boldsymbol{\mu}) \\ \hline \mathbf{0} & \gamma(\boldsymbol{\mu}) \end{array}\right). \tag{1.24}$$

We want to point out that (1.23) satisfies the affinity assumption (1.9), but it is defined on a parameter dependent space. Our goal is now to obtain a formulation of the problem in which all the forms are defined on the reference domain.

Denoting with X the space $X_o(\boldsymbol{\mu}_{ref})$, given a value $\boldsymbol{\mu} \in \mathcal{D}$, for each $v_o \in X_o(\boldsymbol{\mu})$ we can define $v \in X$ as $v = v_o \circ T(\cdot, \boldsymbol{\mu})$ (note that we have actually defined a one-to-one correspondence between X and $X_o(\boldsymbol{\mu})$). We can now track back the integrals in (1.23) obtaining:

$$a(v, w; \boldsymbol{\mu}) := \sum_{l=1}^{L_{dom}} \int_{\Omega^l} \begin{pmatrix} \frac{\partial v}{\partial x_1} & \frac{\partial v}{\partial x_2} & v \end{pmatrix} \mathbf{K}_l(\boldsymbol{\mu}) \begin{pmatrix} \frac{\partial w}{\partial x_1} \\ \\ \frac{\partial w}{\partial x_2} \\ \\ w \end{pmatrix} \tag{1.25}$$

with v and w belonging to X. In (1.25) $\mathbf{K}_l(\boldsymbol{\mu})$, $l = 1, \ldots, L_{dom}$, represents the transformed operator. The latter can be explicitly written in this way:

$$\mathbf{K}_l(\boldsymbol{\mu}) = J^l(\boldsymbol{\mu}) \widetilde{\mathbf{G}}^l(\boldsymbol{\mu}) \mathbf{K}_{o,l}(\boldsymbol{\mu}) (\widetilde{\mathbf{G}}^l(\boldsymbol{\mu}))^{\mathtt{T}} \tag{1.26}$$

where

$$\widetilde{\mathbf{G}}^l(\boldsymbol{\mu}) = \left(\begin{array}{c|c} (\mathbf{G}^l(\boldsymbol{\mu}))^{-1} & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right). \tag{1.27}$$

Similarly we can require that the linear form $f_o(\cdot; \boldsymbol{\mu})\colon X_o(\boldsymbol{\mu}) \to \mathbb{R}$ in (1.16) is, for all $v \in X_o(\boldsymbol{\mu})$:

$$F_o(v_o; \boldsymbol{\mu}) = \sum_{l=1}^{L_{dom}} \int_{\Omega_o^l(\boldsymbol{\mu})} \mathbf{F}_{o,l}(\boldsymbol{\mu})\, v_o. \tag{1.28}$$

Here $\mathbf{F}_{o,l}$, $l = 1, \ldots, L_{dom}$, is a function $\mathcal{D} \to \mathbb{R}$.

Acting exactly as before, we can obtain a linear form defined on the reference space X. This form turns out to be, for $v \in X$:

$$F(v; \boldsymbol{\mu}) = \sum_{l=1}^{L_{dom}} \int_{\Omega^l(\boldsymbol{\mu})} \mathbf{F}_l(\boldsymbol{\mu})\, v \tag{1.29}$$

where, for $l = 1, \ldots, L_{dom}$, the parametric coefficient $\mathbf{F}_l(\boldsymbol{\mu})$ is

$$\mathbf{F}_l(\boldsymbol{\mu}) = J^l(\boldsymbol{\mu}) \mathbf{F}_{o,l}(\boldsymbol{\mu}). \tag{1.30}$$

It is important to note that after this discussion we have actually managed to rewrite the problem (1.16) in the form of (1.4), from which we can obtain the *truth* approximation formulation like in (1.11).

For further details about the construction of the domain decomposition performed, for example, by the rbMIT© software we refer to [50]. For more complex classes of geometries, which involve non-affine mappings, we need to resort to some interpolation technique (e.g. empirical interpolation method) in order to recover the "affinity" assumption [1, 10, 16, 34].

1.2 The Reduced Basis method

As already mentioned before, the RB method aims to approximate the truth solution $u^{\mathcal{N}}(\boldsymbol{\mu})$ of (1.11) by performing a Galerkin projection on a low-dimensional subspace of $X^{\mathcal{N}}$ spanned by solutions of (1.11), that we will call *snapshot* solutions, computed for a well-chosen set of parameter values. In this section we will at first explain the main features of the RB method and then we will illustrate the method used to choose the *snapshots*, highlighting in particular the *a posteriori* error estimates used.

1.2.1 Main features

Let us suppose that we are given a problem in the form (1.4) and its *truth* approximation (1.11). We recall that the dimension of the finite element space $X^{\mathcal{N}}$ is $\mathcal{N}$. We introduce now, given an integer $N_{max} \ll \mathcal{N}$, a sequence of subspaces of $X^{\mathcal{N}}$. For $N = 1, \dots, N_{max}$, let $X_N^{\mathcal{N}}$ be a N-dimensional hierarchical subspace of $X^{\mathcal{N}}$ such that:

$$X_1^{\mathcal{N}} \subset X_2^{\mathcal{N}} \subset \cdots \subset X_N^{\mathcal{N}} \subset \cdots \subset X_{N_{max}}^{\mathcal{N}}. \tag{1.31}$$

We will call these subspaces "RB spaces". Theoretically, the hierarchical choice of the subspaces would not be necessary. Nevertheless, it turns out to be very useful because it allows a better exploitation of the memory during the computation and, as a consequence, this improves the efficiency of the method.

As mentioned at the beginning of this chapter, we focus on Lagrange RB spaces. In order to define them, we need to introduce a set of N_{max} parameter values:

$$S = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^{N_{max}}\} \tag{1.32}$$

and so we can define for $N = 1, \dots, N_{max}$:

$$X_N^{\mathcal{N}} = \text{span}\{u^{\mathcal{N}}(\boldsymbol{\mu}^n) \,|\, 1 \leq n \leq N\}. \tag{1.33}$$

The idea behind this definition is to interpolate the truth manifold (1.3) in correspondence of the parameter values belonging to S.

We observe that, by definition, the spaces defined in (1.33) satisfy the hierarchical property (1.31).

Before going ahead, we want to discuss a little about non-hierarchical RB spaces. A possible choice of such approximation spaces, in the case $P = 1$, can be [50]:

$$X_N^{\mathcal{N}} = \text{span}\{u^{\mathcal{N}}(\boldsymbol{\mu}_N^n) \,|\, 1 \leq n \leq N\} \tag{1.34}$$

where, if we assume that $\mathcal{D} = [\mu^{min}, \mu^{max}]$ and $\mu^{min} > 0$,

$$\boldsymbol{\mu}_N^n = \mu^{min} \exp\left\{\frac{n-1}{N-1} \ln\left(\frac{\mu^{max}}{\mu^{min}}\right)\right\}, \quad 1 \leq n \leq N,\ 2 \leq N \leq N_{max}. \tag{1.35}$$

As we can see from the definition of the $\boldsymbol{\mu}_N^n$, for each value of N we need to recompute all the N *snapshot* solutions $u^{\mathcal{N}}(\boldsymbol{\mu}_N^n)$. Although affected by this practical problem, the spaces (1.34) still have good approximation properties. Indeed, it has been proved (for a particular case) that the ratio between the approximation error (i.e. the energy norm of the difference between the *truth* solution and the RB one) and the energy norm of the truth solution does not depend on $\mathcal{N}$ and it exponentially decays as $N \to \infty$ [41].

Galerkin projection

Given a value $\boldsymbol{\mu} \in \mathcal{D}$ of the parameter and a dimension N, $1 \leq N \leq N_{max}$, of the RB space, we define the RB solution $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ such that:

$$a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = F(v_N; \boldsymbol{\mu}) \quad \forall v_N \in X_N^{\mathcal{N}}. \tag{1.36}$$

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.

If the bilinear form a is symmetric, is straightforward to prove (via Galerkin orthogonality) the following best "fit" approximation result:

$$|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \inf_{w^{\mathcal{N}} \in X_N^{\mathcal{N}}} |||u^{\mathcal{N}}(\boldsymbol{\mu}) - w^{\mathcal{N}}|||_{\boldsymbol{\mu}}. \tag{1.37}$$

Similar estimates can be obtained in the case in which the bilinear form is not symmetric, the main difference is a $\boldsymbol{\mu}$-dependent constant which multiplies the right-hand side [7, 8].

In order to discuss the Offline-Online computational decoupling, we write explicitly the linear system associated to (1.36). First of all we apply the Gram-Schmidt process [47] with, respect to the inner product $(\cdot, \cdot)_X$ defined in (1.6), to the *snapshots* $u(\boldsymbol{\mu}^n)$, $n = 1, \ldots, N_{max}$, spanning the RB spaces. We denote with $\zeta_n^{\mathcal{N}}$, $n = 1, \ldots, N_{max}$, the mutually orthonormal functions obtained. The RB solution can be now expressed by

$$u_N^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{m=1}^{N} u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu}) \zeta_m^{\mathcal{N}} \tag{1.38}$$

then, choosing $\zeta_n^{\mathcal{N}}$ as v in (1.36), we obtain

$$\sum_{m=1}^{N} a(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu}) = F(\zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) \tag{1.39}$$

ant this can be done for $n = 1, \ldots, N$, thus obtaining a $N \times N$ linear system [45].

Offline-Online computational decoupling

Given the system (1.39), we can now resort to the affine assumptions (1.9) and (1.10) to construct an efficient Offline-Online procedure. The system (1.39) could be rewritten

$$\sum_{m=1}^{N} \left(\sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}) \right) u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{q'=1}^{Q_F} \Theta_F^{q'}(\boldsymbol{\mu}) f^{q'}(\zeta_n^{\mathcal{N}}) \tag{1.40}$$

for $n = 1, \ldots, N$. The system we have obtained can be expressed in matrix form

$$\left(\sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{A}_N^q \right) \mathbf{u}_N(\boldsymbol{\mu}) = \sum_{q'=1}^{Q_F} \Theta_F^{q'}(\boldsymbol{\mu}) \mathbf{F}_N^{q'} \tag{1.41}$$

where

$$(\mathbf{u}_N(\boldsymbol{\mu}))_m = u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu}), \quad \left(\mathbf{A}_N^q\right)_{nm} = a^q(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}), \quad \left(\mathbf{F}_N^q\right)_n = f^{q'}(\zeta_n^{\mathcal{N}}) \tag{1.42}$$

for $m, n = 1, \ldots, N$.

In order to compute the matrices $\mathbf{A}_N^q$ and $\mathbf{F}_N^{q'}$ we can recall that $\zeta_n^{\mathcal{N}}$ belongs to $X^{\mathcal{N}}$ for $n = 1, \ldots, N$ and so it holds that:

$$\zeta_n^{\mathcal{N}} = \sum_{i=1}^{\mathcal{N}} \zeta_{ni}^{\mathcal{N}} \varphi_i \quad 1 \leq i \leq N, \tag{1.43}$$

being $\{\varphi\}_{i=1}^{\mathcal{N}}$ the base of the FE space $X^{\mathcal{N}}$. Denoting with $\mathbf{Z}$ the $\mathcal{N} \times N$ matrix whose columns are the coordinates of $\zeta_1^{\mathcal{N}}, \ldots, \zeta_N^{\mathcal{N}}$ with respect to $\{\varphi\}_{i=1}^{\mathcal{N}}$. Then we have that

$$\begin{aligned} \mathbf{A}_N^q &= \mathcal{Z}^{\mathrm{T}} \mathbf{A}_{\mathcal{N}}^q \mathcal{Z} \quad 1 \leq q \leq Q_a \\ \mathbf{F}_N^{q'} &= \mathcal{Z}^{\mathrm{T}} \mathbf{F}_{\mathcal{N}}^{q'} \quad 1 \leq q' \leq Q_F \end{aligned} \tag{1.44}$$

where

$$(\mathbf{A}_{\mathcal{N}}^q)_{ij} = a^q(\varphi_j, \varphi_i), \quad (\mathbf{F}_{\mathcal{N}}^{q'})_i = F(\varphi_i). \tag{1.45}$$

It is crucial to note that in (1.41) the matrices $\mathbf{A}_N^q$ and $\mathbf{F}_N^{q'}$ do not depend on the parameter. So, a good computational strategy is to compute and store them once for all. The computation and storage of the $\boldsymbol{\mu}$-independent structures is called "Offline" stage. More precisely in this stage we compute and store:

- FE stiffness matrices $\mathbf{A}_{\mathcal{N}}^q$, for $q = 1, \ldots, N$, and FE right-hand side terms $\mathbf{F}_{\mathcal{N}}^{q'}$, for $q' = 1, \ldots, N$;
- *snapshot* solutions and the corresponding orthonormal basis $\{\zeta_n^{\mathcal{N}}\}_{n=1}^{N_{max}}$;
- RB stiffness matrices $\mathbf{A}_N^q$, for $q = 1, \ldots, N$, and RB right-hand side terms $\mathbf{F}_N^{q'}$, for $q' = 1, \ldots, N$.

We recall that our aim is to obtain, given a new value $\boldsymbol{\mu} \in \mathcal{D}$, a fast and reliable approximation of $u^{\mathcal{N}}(\boldsymbol{\mu})$. To do this, we need to evaluate the coefficients $\Theta_a^q(\boldsymbol{\mu})$ and $\Theta_F^q(\boldsymbol{\mu})$ in order to assemble the $N \times N$ system in (1.41). Once this system has been solved, the RB solution is obtained trough the relation (1.38). The operations done to perform the evaluation $\boldsymbol{\mu} \mapsto u_N^{\mathcal{N}}(\boldsymbol{\mu})$ constitute the "Online" stage.

Let us now analyse the computational cost of the Online stage. First of all we have to consider a cost of $O(Q_a N^2) + O(Q_F N)$ to get the matrix and the right-hand side of the system (1.41), then we need $O(N^3)$ operations to solve it [45, 47]. At last we have to do $O(N)$ operations to perform the product in (1.38) to obtain the solution. As regards the memory used, during the Online stage the storage cost is $O(Q_a N_{max}^2) + O(Q_F N_{max})$, thanks to the hierarchical space assumption (1.31). The latter assumption allow us to store the RB system matrices related to the RB space $X_{N_{max}}^{\mathcal{N}}$ so, if we want to use RB spaces of dimension $N \leq N_{max}$. we need just to take the principal submatrices (or subvectors) of the already stored ones.

The most important thing to note is that the Online stage cost is completely independent from $\mathcal{N}$.

1.2.2 Sampling strategies

We are now going to discuss about the greedy procedure [45, 50] used to explore the parameter space and to construct the RB space. Let us define the *train* samples set Ξ_{train} as a finite subset of $\mathcal{D}$, with cardinality $|\Xi_{train}| = n_{train}$. We need that n_{train} is large enough to ensure that Ξ_{train} is a good "approximation" of the parameter space $\mathcal{D}$. This means that the results

of the greedy are almost insensitive to further refinement of the parameter sample [50]. The choice of the *train* samples is done by using Monte Carlo methods with respect to a uniform or a log-uniform density.

In order to perform a greedy procedure, we need a sharp and computationally inexpensive *a posteriori* error estimator $\boldsymbol{\mu} \mapsto \Delta_N(\boldsymbol{\mu})$, that is

$$|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \Delta_N(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \quad 1 \leq N \leq N_{max} \tag{1.46}$$

that we will define and discuss in section 1.2.3. The basic idea of the algorithm is, at each step:

1. find the value $\tilde{\boldsymbol{\mu}} \in \Xi_{train}$ for which the estimator Δ_N is maximized;
2. add to the Lagrangian basis the solution $u^{\mathcal{N}}(\tilde{\boldsymbol{\mu}})$, to be computed.

By acting in this way, in the $(N+1)$-th iteration, we are adding to the N already chosen basis the solution that is worst approximated by Galerkin projection onto $X_N^{\mathcal{N}}$. The algorithm stops when the maximum estimated error is less then a prescribed tolerance ε_{tol}^*. We introduce also a secondary stopping criterion by setting N_{max} as the maximum number of basis we are willing to accept. If the tolerance has been obtained with a number of basis $\widetilde{N}$ less than N_{max} we set $N_{max} = \widetilde{N}$. The algorithm can be implemented as follows:

Algorithm 1 Greedy

$S_1 = \{\boldsymbol{\mu}^1\}$; compute $u^{\mathcal{N}}(\boldsymbol{\mu}^1)$;
$X_1 = \text{span}\{u^{\mathcal{N}}(\boldsymbol{\mu}^1)\}$;
for $N = 2 : N_{max}$ **do**
  $\boldsymbol{\mu}^N = \text{argmax}_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_{N-1}(\boldsymbol{\mu})$;
  $\varepsilon_{n-1} = \Delta_{N-1}(\boldsymbol{\mu}^N)$;
  if $\varepsilon_{n-1} \leq \varepsilon_{tol}^*$ **then**
    $N_{max} = N - 1$;
  end if
  compute $u^{\mathcal{N}}(\boldsymbol{\mu}^N)$;
  $S_N = S_{N-1} \cup \{\boldsymbol{\mu}^N\}$;
  $X_N^{\mathcal{N}} = X_{N-1}^{\mathcal{N}} \oplus \text{span}\{u^{\mathcal{N}}(\boldsymbol{\mu}^N)\}$;
end for.

Several options have been recently proposed to improve the sampling strategy [11, 13] For further details about the convergence rates of this algorithm, we refer to [2].

1.2.3 *A posteriori* error estimates

One of the most important features of the reduced basis method is the *a posteriori* error estimation. As we have seen in section 1.2.2, the estimators Δ_N, $N = 1 \ldots, N_{max}$, play a crucial role in the construction of the RB space. For our purposes, a good *a posteriori* error estimator have to fulfil the following characteristics:

- It has to be *rigorous*, in the sense that the inequality

  $$|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq \Delta_N(\boldsymbol{\mu})$$

  must hold for all $\boldsymbol{\mu} \in \mathcal{D}$. This is a fundamental requirement to ensure reliability to the RB method.

- It has to be *sharp*. An overly conservative error bound can cause inefficient approximation spaces, that is with a dimension N unnecessarily high.
- It has to be computationally *efficient*. The computation of the error bound must be very inexpensive both to speed up the Offline stage (i.e. greedy algorithm) and to allow its use in the Online stage. The computational cost should be independent of $\mathcal{N}$.

Before defining the error estimator we will use, we need some preliminaries [45]. First of all we observe that the error $e(\boldsymbol{\mu}) := u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})$, that belongs to $X^{\mathcal{N}}$, satisfies

$$a(e(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = r(v^{\mathcal{N}}; \boldsymbol{\mu}), \quad \forall v \in X^{\mathcal{N}}, \tag{1.47}$$

where $r(\cdot; \boldsymbol{\mu}) \in (X^{\mathcal{N}})'$ is the RB residual

$$r(v^{\mathcal{N}}; \boldsymbol{\mu}) = F(v^{\mathcal{N}}; \boldsymbol{\mu}) - a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}), \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}. \tag{1.48}$$

As $r(\cdot; \boldsymbol{\mu})$ is a continuous linear functional defined on $X^{\mathcal{N}}$, we can apply the Riesz representation theorem and get $\hat{e}(\boldsymbol{\mu}) \in X^{\mathcal{N}}$ such that:

$$r(v^{\mathcal{N}}; \boldsymbol{\mu}) = (\hat{e}(\boldsymbol{\mu}), v^{\mathcal{N}})_X \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \tag{1.49}$$

and

$$\|r(v^{\mathcal{N}}; \boldsymbol{\mu})\|_{(X^{\mathcal{N}})'} = \sup_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{|r(v^{\mathcal{N}}; \boldsymbol{\mu})|}{\|v^{\mathcal{N}}\|_X} = \|\hat{e}(\boldsymbol{\mu})\|_X. \tag{1.50}$$

Now we introduce a lower bound $\alpha_{LB}^{\mathcal{N}} \colon \mathcal{D} \to \mathbb{R}$ for the coercivity constant (1.12) such that:

$$\begin{aligned} &0 < \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}) \leq \alpha^{\mathcal{N}}\boldsymbol{\mu}, \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \\ &\text{the computational cost to evaluate } \boldsymbol{\mu} \mapsto \alpha_{LB}(\boldsymbol{\mu}) \text{ is independent of } \mathcal{N}. \end{aligned} \tag{1.51}$$

To get this lower bound we can resort to the Successive Constraints Method (SCM) [25, 50] that we will discuss in section 1.2.4.

Now we are ready to give the definition of our *a posteriori* error estimator. Let us then define the estimator for the energy norm of the error:

$$\Delta_N(\boldsymbol{\mu}) := \frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{(\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}))^{\frac{1}{2}}}. \tag{1.52}$$

We next introduce the effectivity associated to the estimator:

$$\eta_N(\boldsymbol{\mu}) := \frac{\Delta_N(\boldsymbol{\mu})}{|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}} \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.53}$$

We can prove the following result [50]:

Proposition 1.2.1. *For any $N = 1, \ldots, N_{max}$ and for any $\boldsymbol{\mu} \in \mathcal{D}$ the effectivity satisfies*

$$1 \leq \eta_N(\boldsymbol{\mu}) \leq \sqrt{\frac{\gamma(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}} \tag{1.54}$$

The left inequality means that the error bound is *rigorous*, while the right one is related to the sharpness of the estimate. Note that the upper bound on the effectivity in (1.54) is independent of N and hence stable with respect to RB "refinement". As the method we are using to construct the coercivity lower bound $\alpha_{LB}^{\mathcal{N}}$ is designed in such a way that

$$\frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})} \leq C \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{1.55}$$

where C is a constant, recalling (1.13) we can observe that:

$$\eta_N(\boldsymbol{\mu}) \leq \sqrt{\frac{\gamma(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}} \leq \sqrt{\frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}\frac{\gamma(\boldsymbol{\mu})}{\alpha(\boldsymbol{\mu})}} \leq \sqrt{C\frac{\gamma(\boldsymbol{\mu})}{\alpha(\boldsymbol{\mu})}} \tag{1.56}$$

which means that the upper bound for the sensitivity does not depend on the finite element approximation.

Computation of the dual norm of the residual

In order to compute the error bound, we want to show how the dual norm of the residual, that is $\|\hat{e}(\boldsymbol{\mu})\|_X$, can be computed. The goal will be to build a procedure with a computational cost independent of $\mathcal{N}$, by exploiting the affine assumptions (1.9) and (1.10).

First of all let us expand the residual. Given $v^{\mathcal{N}} \in X^{\mathcal{N}}$, we have:

$$\begin{aligned} r(v^{\mathcal{N}};\boldsymbol{\mu}) &= F(v^{\mathcal{N}};\boldsymbol{\mu}) - a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v^{\mathcal{N}};\boldsymbol{\mu}) \\ &= \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) F^q(v^{\mathcal{N}}) - \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q\left(\sum_{m=1}^{N} u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu})\zeta_m^{\mathcal{N}}, v^{\mathcal{N}}\right) \\ &= \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) F^q(v^{\mathcal{N}}) - \sum_{m=1}^{N} u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu}) \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(\zeta_m^{\mathcal{N}}, v^{\mathcal{N}}). \end{aligned} \tag{1.57}$$

Recalling (1.49), we have:

$$\hat{e}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu})\mathcal{F}^q + \sum_{m=1}^{N} u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu}) \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu})\mathcal{L}_m^q \tag{1.58}$$

where $\mathcal{F}^q$ and $\mathcal{L}_m^q$ are such that:

$$\begin{aligned} &(\mathcal{F}^q, v^{\mathcal{N}})_X = F^q(v^{\mathcal{N}}) && \forall\, v^{\mathcal{N}} \in X^{\mathcal{N}}, \quad 1 \leq q \leq Q_F, \\ &(\mathcal{L}_m^q, v^{\mathcal{N}})_X = -a^q(\zeta_m^{\mathcal{N}}, v^{\mathcal{N}}) && \forall\, v^{\mathcal{N}} \in X^{\mathcal{N}}, \quad 1 \leq q \leq Q_a, 1 \leq m \leq N. \end{aligned} \tag{1.59}$$

From this expression of the error, we easily obtain that:

$$\begin{aligned} \|\hat{e}(\boldsymbol{\mu})\|_X^2 = &\sum_{q=1}^{Q_F}\sum_{q'=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu})\Theta_F^{q'}(\boldsymbol{\mu})(\mathcal{F}^q, \mathcal{F}^{q'})_X + \sum_{q=1}^{Q_a}\sum_{m=1}^{N} \Theta_a^q(\boldsymbol{\mu}) u_{N\,m}^{\mathcal{N}}(\boldsymbol{\mu})\Big\{ \\ &2\sum_{q'=1}^{Q_F} \Theta_F^{q'}(\boldsymbol{\mu})(\mathcal{F}^{q'}, \mathcal{L}^q)_X + \sum_{q'=1}^{Q_a}\sum_{m'=1}^{N} \Theta_a^q(\boldsymbol{\mu}) u_{N\,m'}^{\mathcal{N}}(\boldsymbol{\mu})(\mathcal{L}_m^q, \mathcal{L}_{m'}^{q'})_X\Big\}. \end{aligned} \tag{1.60}$$

It is now possible to see that we can compute and store the parameter independent quantities once for all. These quantities are:

- the FE "pseudo-solutions" $\mathcal{F}^q$, and $\mathcal{L}^q_m$;
- the scalar products $(\mathcal{F}^q, \mathcal{F}^{q'})_X$, $(\mathcal{F}^{q'}, \mathcal{L}^q)_X$ and $(\mathcal{L}^q_m, \mathcal{L}^{q'}_{m'})_X$.

The cost of their computation depends on Q_a, Q_F, $\mathcal{N}$ and N_{max}.

Once we have the latter quantities, we can evaluate $\boldsymbol{\mu} \mapsto \|\hat{e}(\boldsymbol{\mu})\|^2_X$ at a very low computational cost. Given a new value $\boldsymbol{\mu} \in \mathcal{D}$ we need just to evaluate the Θ functions and then perform the weighted sum of the already stored quantities. The operation count is $O(N^2Q_a^2 + NQ_aQ_F + Q_F^2)$ and it does not depend on $\mathcal{N}$.

1.2.4 Lower bound for the coercivity constant

As we mentioned in section 1.2.3 when we defined the *a posteriori* error estimator, we have now to introduce the Successive Constraint Method (SCM) [25] for the evaluation of the coercivity lower bound $\alpha^{\mathcal{N}}_{LB}$.

First of all, we note that the computation of the discrete coercivity constant (1.12) is actually a generalized minimum eigenvalue problem. Denoting with $\{\varphi_i\}_{i=1}^{\mathcal{N}}$ a lagrangian FE base for $X^{\mathcal{N}}$, we can define the mass matrix $\overline{\mathbf{M}}$ corresponding to the scalar product (1.6) such that:

$$\overline{\mathbf{M}}_{ij} = (\varphi_i, \varphi_j)_X \quad 1 \le i \le j \le \mathcal{N}. \tag{1.61}$$

We can also define, for $q = 1, \ldots, \mathcal{N}$, the $\boldsymbol{\mu}$-dependent symmetric matrix $\mathbf{B}(\boldsymbol{\mu})$ associated to the symmetric part of the bilinear form a:

$$(\mathbf{B}(\boldsymbol{\mu}))_{ij} = a^{sym}(\varphi_i, \varphi_j) \quad 1 \le i \le j \le \mathcal{N}. \tag{1.62}$$

Denoting with $\mathbf{v}$ the coordinate vector of $v^{\mathcal{N}} \in X^{\mathcal{N}}$ with respect to the given base, we have that:

$$\frac{a(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu})}{\|v^{\mathcal{N}}\|^2_X} = \frac{a^{sym}(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu})}{\|v^{\mathcal{N}}\|^2_X} = \frac{\mathbf{v}^T\mathbf{B}(\boldsymbol{\mu})\mathbf{v}}{\mathbf{v}^T\mathbf{M}\mathbf{v}} \tag{1.63}$$

and then the problem

$$\begin{aligned} &\text{given } \boldsymbol{\mu} \in X^{\mathcal{N}}, \text{ compute} \\ &\alpha^{\mathcal{N}}(\boldsymbol{\mu}) = \inf_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{a(v^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu})}{\|v^{\mathcal{N}}\|^2_X} \end{aligned} \tag{1.64}$$

is equivalent to find the minimum generalized eigenvalue of the following generalized eigenvalue problem:

$$\mathbf{B}(\boldsymbol{\mu})\mathbf{v} = \lambda\overline{\mathbf{M}}\mathbf{v}. \tag{1.65}$$

that can be treated by the Lanczos method [9]. To improve the efficiency, we define the norm (1.6) by setting

$$\tau = \inf_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{a(v^{\mathcal{N}}, v^{\mathcal{N}}; \overline{\mu})}{\|v^{\mathcal{N}}\|^2_X} \tag{1.66}$$

as done in [50].

Before introducing the SCM, let us call $a^{sym,q}$ the symmetric part of the form a^q defined in (1.9).

We define now an objective functional $\mathcal{J}^{obj} : \mathcal{D} \times \mathbb{R}^{Q_a} \to \mathbb{R}$ as

$$J^{obj}(\boldsymbol{\mu}; y) = \sum_{i=1}^{Q_a} \Theta^q_a(\boldsymbol{\mu})\, y^q, \tag{1.67}$$

where $y = (y_1, \dots, y_{Q_a})$. An equivalent formulation of the problem (1.64) can be:

$$\begin{aligned} &\text{given } \boldsymbol{\mu} \in X^{\mathcal{N}}, \text{ compute} \\ &\alpha^{\mathcal{N}}(\boldsymbol{\mu}) = \inf_{y \in \mathcal{Y}} J^{obj}(\boldsymbol{\mu}; y) \end{aligned} \tag{1.68}$$

where

$$\mathcal{Y} = \left\{ y \in \mathbb{R}^q | \exists\, w_y^{\mathcal{N}} \in X^{\mathcal{N}} \quad \text{s.t.} \quad y_q = \frac{a^{sym,q}(w_y^{\mathcal{N}}, w_y^{\mathcal{N}})}{\|w_y^{\mathcal{N}}\|_X}, 1 \le q \le Q_a \right\}. \tag{1.69}$$

We then define the "bounding box"

$$\mathcal{B} = \prod_{q=1}^{Q_a} \left[\inf_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{a^{sym,q}(v^{\mathcal{N}}, v^{\mathcal{N}})}{\|v^{\mathcal{N}}\|_X^2}, \sup_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{a^{sym,q}(v^{\mathcal{N}}, v^{\mathcal{N}})}{\|v^{\mathcal{N}}\|_X^2} \right]; \tag{1.70}$$

from the continuity hypothesis, $\mathcal{B}$ is bounded. At last we define the *coercivity constraint sample*

$$C_J = \{\boldsymbol{\mu}_{SCM}^1, \dots, \boldsymbol{\mu}_{SCM}^J\}, \tag{1.71}$$

where $\boldsymbol{\mu}_{SCM}^i \in \mathcal{D}$ for $i = 1, \dots, J$, and we denote with $C_J^{M,\boldsymbol{\mu}}$ the set of $M \ge 1$ points in C_J closest to $\boldsymbol{\mu} \in \mathcal{D}$, with respect to the usual Euclidean norm in $\mathbb{R}^p$. If $M > J$, we set $C_J^{M,\boldsymbol{\mu}} = C_J$.

Now, given $\boldsymbol{\mu} \in \mathcal{D}$ we define the "lower bound" set $\mathcal{Y}_{LB}(\boldsymbol{\mu}; C_J, M) \subset \mathbb{R}^q$ as

$$\mathcal{Y}_{LB}(\boldsymbol{\mu}; C_J, M) = \left\{ y \in \mathbb{R}^q | y \in \mathcal{B}; \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}') y^q \ge \alpha^{\mathcal{N}}(\boldsymbol{\mu}') \quad \forall \boldsymbol{\mu}' \in C_J^{M,\boldsymbol{\mu}} \right\}. \tag{1.72}$$

We have the following lemma [50]

Lemma 1.2.2. *Given $C_J^{M,\boldsymbol{\mu}} \subset \mathcal{D}$ and $M \in \boldsymbol{N}$*

$$\mathcal{Y} \subset \mathcal{Y}_{LB}(\boldsymbol{\mu}; C_J, M) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.73}$$

Once we have this lemma, we can define our lower bound

$$\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M) = \min_{y \in \mathcal{Y}_{LB}(\boldsymbol{\mu}; C_J, M)} \mathcal{J}^{obj}(\boldsymbol{\mu}; y) \tag{1.74}$$

and we easily obtain [50] that

Proposition 1.2.3. *Given $C_J^{M,\boldsymbol{\mu}} \subset \mathcal{D}$ and $M \in \boldsymbol{N}$*

$$\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}) \le \alpha^{\mathcal{N}}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.75}$$

It is important to note that the (1.74) is a Linear Program (LP) problem, with Q_a design variables and $2Q_a + M$ (one-sided) inequality constraints. It is fundamental to observe that the computational cost for evaluate $\boldsymbol{\mu} \mapsto \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})$, given $\mathcal{B}$ and the set $\{\alpha^{\mathcal{N}}(\boldsymbol{\mu}') | \boldsymbol{\mu}' \in C_J\}$ is independent of $\mathcal{N}$. This observation, together with Proposition 1.2.3, shows that our requirements (1.51) are fulfilled.

In order to perform a "greedy" construction of the *coercivity constraint samples*, we introduce an upper bound for the coercivity constant. We define the set

$$\mathcal{Y}_{UB}(\boldsymbol{\mu}; C_J, M) = \left\{ y^*(\boldsymbol{\mu}') \,|\, \boldsymbol{\mu}' \in \mathcal{C}_J^{M,\boldsymbol{\mu}} \right\} \tag{1.76}$$

where

$$y^*(\boldsymbol{\mu}') = \underset{y \in \mathcal{Y}}{\operatorname{argmin}} \, \mathcal{J}^{obj}(\boldsymbol{\mu}; y). \tag{1.77}$$

In case of non-uniqueness, any minimizer suffices.

We define our upper bound in the following way:

$$\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M) = \min_{y \in \mathcal{Y}_{UB}(\boldsymbol{\mu}; C_J, M)} \mathcal{J}^{obj}(\boldsymbol{\mu}; y). \tag{1.78}$$

As by construction $\mathcal{Y}_{UB}(\boldsymbol{\mu}; C_J, M) \subset \mathcal{Y}$, for given $C_J \subset \mathcal{D}$ and $M \in \mathbb{N}$, it holds that:

$$\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M) \geq \alpha^{\mathcal{N}}(\boldsymbol{\mu}) \tag{1.79}$$

for any $\boldsymbol{\mu} \in \mathcal{D}$.

Greedy selection of the *coercivity constraint samples*

We illustrate now a greedy algorithm [25, 50] for the construction of the set C_J. Similarly to how we did in section 1.2.2, we define a finite "train" sample $\Xi_{train}^{SCM} \subset \mathcal{D}$, such that $|\Xi_{train}^{SCM}| = n_{train}^{SCM}$. We also set a tolerance $\varepsilon_{SCM} \in (0, 1)$.

The algorithm is the following:

Algorithm 2 SCM

$J = 1$;
$C_1 = \{\boldsymbol{\mu}_{SCM}^1\}$;
while $\max_{\boldsymbol{\mu} \in \Xi_{train}^{SCM}} \left[\frac{\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M) - \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)}{\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)} \right] > \varepsilon_{SCM}$ **do**
  $\boldsymbol{\mu}_{SCM}^{J+1} = \operatorname{argmax}_{\boldsymbol{\mu} \in \Xi_{train}^{SCM}} \left[\frac{\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M) - \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)}{\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)} \right]$;
  $C_{J+1} = C_J \cup \boldsymbol{\mu}_{SCM}^{J+1}$;
  $J = J + 1$;
end while
$J_{max} = J$.

Let us observe that our construction of the lower bound guarantees the condition (1.55). We can indeed see that:

$$\begin{aligned} \frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)} &= \frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M) - (\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M) - \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)} \\ &\leq \frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)} \frac{1}{1 - \varepsilon_{SCM}} \\ &\leq \frac{1}{1 - \varepsilon_{SCM}} \quad \forall \boldsymbol{\mu} \in \Xi_{train,SCM} \end{aligned} \tag{1.80}$$

Note that, by requiring $\varepsilon_{SCM} \in (0, 1)$, we guarantee that $\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}; C_J, M)$ is positive. In our computations we chose a tolerance $\varepsilon_{SCM} = 0.85$.

Once we have set $M \in \mathbb{N}$ and we have performed the greedy selection of the set $C_{J_{max}}$, for all $\boldsymbol{\mu} \in \mathcal{D}$ we can denote the lower bound with $\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})$, as we actually did in section 1.2.3.

Computational cost

During the Offline stage the following computations are performed:

i) $2Q_a$ eigenproblems over $X^{\mathcal{N}}$ to build the "continuity constraint" box $\mathcal{B}$. Cost: $O(2Q_a\mathcal{N})$.
ii) J_{max} eigenproblems over $X^{\mathcal{N}}$ to form the set $\{\alpha^{\mathcal{N}}(\boldsymbol{\mu})|\boldsymbol{\mu}' \in C_{J_{max}}\}$. Cost: $O(J_{max}\mathcal{N})$.
iii) $J_{max}Q_a$ inner products over $X^{\mathcal{N}}$ to compute $\left\{y^*(\boldsymbol{\mu}') \,|\, \boldsymbol{\mu}' \in \mathcal{C}_J^{M,\boldsymbol{\mu}}\right\}$. Cost: $O(J_{max}Q_a\mathcal{N})$.
iv) $n_{train}^{SCM} J_{max}$ lower bound LP's of size $2Q_a + M$ and the associated enumerations to compute the upper bounds. Cost: $O(n_{train}^{SCM} J_{max} Q_a M)$.

Note that the global Offline computational cost does not depend on the product $n_{train}^{SCM}\mathcal{N}$, so we can choose large train sets and *truth* approximation spaces with high dimension $\mathcal{N}$ without worsening too much the computational efficiency.

In the online stage, for each evaluation $\boldsymbol{\mu} \mapsto \alpha_{LB}(\boldsymbol{\mu})$ we have to:

i) perform a sort of the J_{max} point of C_J in order to build $C_J^{M,\boldsymbol{\mu}}$;
ii) evaluate the Θ functions;
iii) solve the resulting LP to obtain the lower bound.

As usual in our Online procedures, the computational cost is independent of $\mathcal{N}$.

Finally, we want to say that the SCM can be efficiently applied also to compute the inf-sup lower bound for non-coercive problems [45, 52]. Moreover, several improvements for the SCM have been recently proposed [6, 23, 33].

Chapter 2

Stabilized reduced basis method for advection dominated PDEs

In this chapter we will study the performance of the RB method in the approximation of advection diffusion equations, especially when the advection effects are much stronger than the diffusive ones (advection dominated problems). Since such problems give rise to numerical instability, we will need to resort to some classical stabilization methods [48].

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 problem such as Graetz problem or "thermal fin" problem [15, 45, 49, 51]. However, the latter works consider only the case in which the Péclet number - that is, roughly speaking, the ratio between the advection coefficient and the diffusion one - is low, in a sense that we will specify in section 2.1.

The stabilization methods have been used in the RB framework in some works about the approximation of steady advection reaction equations and steady control problems [7, 8, 44, 46]. In these papers we can also find some applications to environmental science problems concerning, in particular, air pollution.

In our work we want to go further in the study of the *stabilized RB method* for advection diffusion problems. After a brief introduction of some stabilization method (section 2.1), we want to 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 see that the latter strategy is not satisfactory, but we will also see that if we "stabilize" both stages we can obtain very good RB approximations. In the last section of this chapter (section 2.3) we will apply the stabilized RB method - with both stages stabilized - using a piecewise quadratic polynomial *truth* approximation space.

2.1 Stabilization methods

In this chapter we will illustrate some stabilization methods for advection-diffusion equations. We will focus in particular on equations of the form

$$-\varepsilon\Delta u + \boldsymbol{\beta}\cdot\nabla u = f \quad \text{in } \Omega \subset \mathbb{R}^2 \tag{2.1}$$

This is a particular case of the general problem $Lu = f$, when L is the diffusion-advection-reaction operator defined in (1.22). When the advective $\boldsymbol{\beta}\cdot\nabla u$ term dominates the diffusive term $-\varepsilon\Delta u$, that is when $|\boldsymbol{\beta}| \gg \varepsilon$, the classical FE approach can be very unsatisfactory,

because the approximated solution can show strong instability phenomena, especially along the direction of the advection field. In the next sections we will illustrate and analyse some stabilization method, able to fix this lack of stability.

2.1.1 Advection dominated problems

Let us make precise assumption on our setting. As mentioned before, we will focus on equations like (2.1) where:

- the diffusion coefficient $\varepsilon\colon \Omega \to \mathbb{R}$ belongs to $L^\infty(\Omega)$ and it exists $\varepsilon_0 > 0$ such that

$$\varepsilon(\boldsymbol{x}) \geq \varepsilon_0 \quad \forall \boldsymbol{x} \in \Omega; \tag{2.2}$$

- the advection field $\boldsymbol{\beta}\colon \Omega \to \mathbb{R}^2$ belongs to $(L^\infty(\Omega))^2$;
- $f\colon \Omega \to \mathbb{R}^2$ is an $L^2(\Omega)$ function.

In order to guarantee the well-posedness of our problem, we suppose also that the following inequality holds:

$$0 \geq \operatorname{div} \boldsymbol{\beta}(\boldsymbol{x}) \geq -d_1 \quad \forall \boldsymbol{x} \in \Omega. \tag{2.3}$$

where d_1 is a positive real constant.

We suppose now that we are given a regular triangulation $\mathcal{T}_h$ (for the definition see [43]), where h is the maximum element diameter (mesh size). For any element $K \in \mathcal{T}_h$, we can then define the local Péclet number [43]:

$$\mathbb{P}\mathrm{e}_K(\boldsymbol{x}) := \frac{|\boldsymbol{\beta}(\boldsymbol{x})| h_K}{2\varepsilon(\boldsymbol{x})} \quad \forall \boldsymbol{x} \in K, \tag{2.4}$$

where h_K is the diameter of K.

We say that we are dealing with an *advection dominated* problem if it holds that:

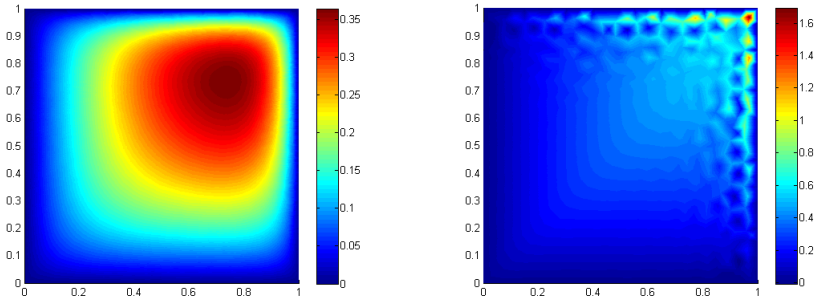
$$\mathbb{P}\mathrm{e}_K(\boldsymbol{x}) > 1 \quad \forall \boldsymbol{x} \in K, \quad \forall K \in \mathcal{T}_h. \tag{2.5}$$

To give just an example of what can happen in an *advetion dominated* situation, we consider the following problem:

$$\begin{cases} -\varepsilon \Delta u + (1,1) \cdot \nabla u = 1 & \text{in } \Omega := (0,1) \times (0,1) \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{2.6}$$

If the coefficient ε is "large", e.g. $\varepsilon = \frac{1}{10}$, the FE method yields a good approximation of the solution, as it can be seen in figure 2.1a. On the contrary, if we choose a "smaller" value of the same coefficient, e.g. $\varepsilon = \frac{1}{300}$, the FE solution is highly affected by spurious oscillations, as we can see in figure 2.1b. The mesh used in these computations has a size $h \approx 0.06$.

Theoretically, to avoid instability problems, it would be sufficient to reduce the mesh size h in order to lower the local Péclet number. Unfortunately, a "small" mesh size yields a significant increase of the computational cost, that can become unaffordable. Just to give an example, we can try to approximate the problem (2.6), with $\varepsilon = \frac{1}{300}$, by using a refined mesh (figure 2.3b) with size $h \approx 0.004$. In this case we have that the local Péclet number (2.4) is always smaller than 1. The solution that we obtain is reported in figure 2.2. In figure 2.3 there is a visual comparison between the meshes used to compute the solutions shown in figures 2.1 and 2.2.



(a) $\varepsilon = \frac{1}{10}$ (b) $\varepsilon = \frac{1}{300}$

Figure 2.1: FE solutions of (2.6) for different values of the diffusion coefficient ε. $h \approx 0.06$

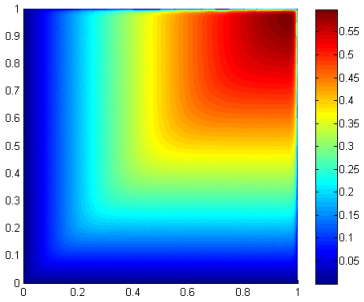


Figure 2.2: FE solutions with refined mesh ($h \approx 0.004$) and$\varepsilon = \frac{1}{300}$.

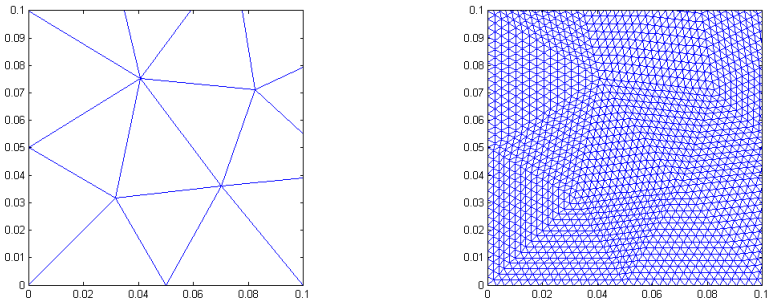
Several stabilization methods have been developed to fix the approximated solution without resorting to mesh refinement [48]. The basic idea behind these methods is to add some sort of artificial diffusion, in order to smooth the "jumps" (boundary or internal layers) that the exact solution can show. Let us note also that ε_0 is the coercivity constant of the bilinear form associated to the equation (2.1) while, when we are in an advection dominated situation, $\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}$ is actually the continuity constant. So, enforcing somehow the coercivity constant we can obtain a better Céa error estimate [48].

A first proposal can be to add to the left-hand side of the equation one of the following additional diffusion terms:

$$L^{ad}\, u = -h\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}\Delta u, \tag{2.7}$$

$$L^{sd}\, u = -\frac{h}{\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}}\,\mathrm{div}[(\boldsymbol{\beta}\cdot\nabla u)\boldsymbol{\beta}]. \tag{2.8}$$

The term (2.7) corresponds to the so called *artificial diffusion method* [48]. By adding this term, we are increasing the diffusion in all directions, producing then an unnecessary "crosswind" smoothing of the solution. To reduce the amount of diffusion introduced, we could use the term (2.8), which add diffusion only along the wind direction, thus avoiding the crosswind smoothing. The latter method is called *streamline upwind diffusion method* [48]. Both these methods are only *weakly consistent*, with a consistency error of $O(h)$. As a consequence, these methods can be useful only with $\mathbb{P}^1$ FE approximation. For further details about these methods we refer to [48].



(a) $h \approx 0.06$ (b) $h \approx 0.004$

Figure 2.3: Visual comparison between the different meshes we used (zoom on $[0, 0.1] \times [0, 0.1]$).

2.1.2 Strongly consistent stabilization methods

As we mentioned in the previous section, the main problem of the *artificial diffusion* and the *streamline upwind diffusion* methods is the fact that they are not consistent, which deteriorates the accuracy of the polynomial space used in the FE approximation. A way to avoid this kind of problem is to use a *strongly consistent* stabilization method. Several methods have been proposed and many of them can be considered as particular cases of the general class that we are going to introduce.

Let us consider now the advection-diffusion operator defined on $H_0^1(\Omega)$:

$$Lv = -\varepsilon \Delta v + \boldsymbol{\beta} \cdot \nabla v \quad \forall v \in H_0^1(\Omega). \tag{2.9}$$

We can split the operator L into its symmetric and skew-symmetric parts:

$$\text{symmetric part:} \qquad L_S = -\varepsilon \Delta v - \frac{1}{2}(\operatorname{div} \boldsymbol{\beta}) v \tag{2.10}$$

$$\text{skew-symmetric part:} \qquad L_{SS} = \boldsymbol{\beta} \cdot \nabla v + (\frac{1}{2} \operatorname{div} \boldsymbol{\beta}) v \tag{2.11}$$

and the following relation holds:

$$L = L_S + L_{SS}. \tag{2.12}$$

Symmetric and skew-symmetric parts can be recovered using the formulae:

$$\begin{aligned} L_S &= \frac{L + L^*}{2} \\ L_{SS} &= \frac{L - L^*}{2} \end{aligned} \tag{2.13}$$

where L^* is the adjoint operator.

We consider now the weak form of the problem (2.1), that is:

$$\begin{aligned} &\text{find } u \in H_0^1(\Omega) \text{ s.t.} \\ &a(u, v) = F(v) \quad \forall v \in H_0^1(\Omega) \end{aligned} \tag{2.14}$$

where: a is the bilinear form associated to the advection diffusion operator

$$a(v, w) = \int_\Omega \varepsilon \nabla v \cdot \nabla w + \boldsymbol{\beta} \cdot \nabla v \, w, \quad v, w \in H_0^1(\Omega), \tag{2.15}$$

while F is the linear form defined by:

$$F(v) = \int_\Omega fv, \quad v \in H_0^1(\Omega). \tag{2.16}$$

As in the previous section, let us suppose that we are given a regular triangulation $\mathcal{T}_h$. We consider the following piecewise polynomial approximation space:

$$\mathbb{P}^r(\mathcal{T}_h) = \{v \in H^1(\Omega) \,|\, v|_K \in \mathbb{P}^r(K), K \in \mathcal{T}_h\} \tag{2.17}$$

where $\mathbb{P}^r(K)$ is the space of polynomials of degree r on the element K. We will denote with $X^{\mathcal{N}}$ the space $\mathbb{P}^r(\mathcal{T}_h) \cap H_0^1(\Omega)$, where $\mathcal{N}$ is its dimension, i.e. the number of degrees of freedom. We define the stabilization terms:

$$\begin{aligned} s^{(\rho)}(v^{\mathcal{N}}, w^{\mathcal{N}}) &= \sum_{K \in \mathcal{T}_h} \delta_K \left(Lv^{\mathcal{N}}, \frac{h_K}{|\boldsymbol{\beta}|}(L_{SS} + \rho L_S) w^{\mathcal{N}} \right)_K \\ \phi^{(\rho)}(v^{\mathcal{N}}) &= \sum_{K \in \mathcal{T}_h} \delta_K \left(f, \frac{h_K}{|\boldsymbol{\beta}|}(L_{SS} + \rho L_S) w^{\mathcal{N}} \right)_K \end{aligned} \tag{2.18}$$

where $(\cdot,\cdot)_K$ is the scalar product in $L^2(K)$. The weights $\delta_K > 0$, $K \in \mathcal{T}_h$, have to be chosen as well as the parameter $\rho \in \mathbb{R}$, which identifies the method.

We can consider now the stabilized problem:

$$\begin{aligned} &\text{find } u^{\mathcal{N}} \in X^{\mathcal{N}} \text{ s.t.} \\ &a(u^{\mathcal{N}}, v^{\mathcal{N}}; \boldsymbol{\mu}) + s^{(\rho)}(u^{\mathcal{N}}, v^{\mathcal{N}}) = F(v^{\mathcal{N}}) + \phi^{(\rho)}(v^{\mathcal{N}}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}. \end{aligned} \tag{2.19}$$

Note that this formulation is *strongly consistent*, i.e. the continuous solution of (2.14) satisfies the variational equality (2.19).

We have actually defined a family of strongly consistent methods that can be identified through the parameter ρ, as we have said before. Several possible choice of ρ have been studied in literature. A first choice can be to set the parameter ρ equal to zero, thus defining the so called *Streamline Upwind/Petrov Galerkin* (SUPG) method [4, 21, 30, 31]. Another possibility is to choose $\rho = 1$ and the corresponding method is called *Galerkin/Least-Squares* (GALS) method [22]. The choice $\rho = -1$ leads to the *Douglas-Wang/Galerkin* (DWG) method [14].

Remark 2.1.1. If the polynomial approximation space chosen is $\mathbb{P}^1(\mathcal{T}_h)$ and the advection field is divergence free, any choice of the parameter ρ yields the same method.

Before going on with the analysis of the SUPG method, in figure 2.4 we show how this method can approximate the solution of (2.6) with $\varepsilon = \frac{1}{300}$. The mesh used is the one shown in figure 2.3a, with $h \approx 0.06$. We recall that the approximated solution given by the standard FE method was very unsatisfactory (see figure 2.1b). Comparing figure 2.4 with 2.2 it is evident the smoothing of the boundary layer performed by the stabilization method.

Analysis of the SUPG method

As in this work we will focus mainly on the SUPG method, we will now analyse its properties.

Denoting with $|||\cdot|||$ the energy norm associated to the bilinear form a, which turns out to be

$$|||v|||^2 = \varepsilon \|\nabla v\|^2_{L^2(\Omega)} + \frac{1}{2}\|(-\operatorname{div}\boldsymbol{\beta})^{\frac{1}{2}} v\|^2_{L^2(\Omega)} \quad \forall v \in H_0^1(\Omega), \tag{2.20}$$

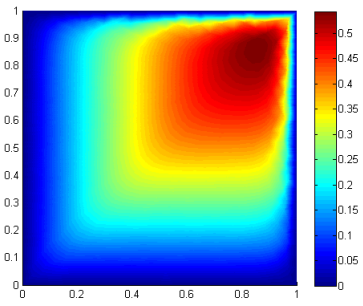


Figure 2.4: FE solutions for different values of the diffusion coefficient ε.

we define the SUPG norm on $H_0^1(\Omega)$ as

$$\|v\|_{SUPG}^2 = |||v|||^2 + \sum_{K\in\mathcal{T}_h} \delta_K \left(L_{SS}v, \frac{h_K}{|\boldsymbol{\beta}|} L_{SS}v \right)_K \quad \forall v \in H_0^1(\Omega). \tag{2.21}$$

It holds that the SUPG bilinear form is coercive with respect to the SUPG norm. This is straightforward, even without any requirement on the parameters δ_K, if the polynomial approximation space is $\mathbb{P}^1(\mathcal{T}_h)$ and the advection field is divergence free, as the energy norm of the SUPG bilinear form is actually the SUPG norm. In general, we have the following theorem, as it is shown in [48]:

Theorem 2.1.1 (Stability)**.** *We assume that we are dealing with the advection dominated case* (2.5) *and that for any element $K \in \mathcal{T}_K$ parameter δ_K satisfies the conditions*

$$0 < \delta_K < C_r^{-1} \tag{2.22}$$

where C_r is the constant of the inverse inequality

$$\sum_{K\in\mathcal{T}_h} h_K^2 \int_K |\Delta v^{\mathcal{N}}|^2 \leq C_r \|\nabla v^{\mathcal{N}}\|_{L^2(\Omega)}^2 \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \tag{2.23}$$

in which r stands for the degree of the piecewise polynomial approximation space. Moreover, if the advection field is not divergence free, we suppose that we have a positive constant $d_0 > 0$ such that

$$0 > -d_0 \geq \operatorname{div}(\beta.) \tag{2.24}$$

and we require that

$$\delta_K h_K \leq \frac{\boldsymbol{\beta}(\boldsymbol{x})}{-\operatorname{div}(\boldsymbol{\beta}(\boldsymbol{x}))} \quad \forall \boldsymbol{x} \in K. \tag{2.25}$$

Then the bilinear associated to the SUPG method is coercive with respect to the advection-diffusion energy norm, that is:

$$a(v^{\mathcal{N}}, v^{\mathcal{N}}) + s^{(0)}(v^{\mathcal{N}}, v^{\mathcal{N}}) \geq \frac{1}{2}\|v^{\mathcal{N}}\|_{SUPG}. \tag{2.26}$$

The theorem 2.1.1 easily yields the following result [48]:

Proposition 2.1.2. *It exists $C > 0$, independent of h, such that*

$$\|u^{\mathcal{N}}\|_{SUPG} \leq C\|f\|_{L^2(\Omega)}, \tag{2.27}$$

where $u^{\mathcal{N}}$ is the solution of (2.19) with $\rho = 0$.

Acting like in [14, 48], it is also possible to prove the following convergence theorem:

Theorem 2.1.3 (Convergence)**.** *Assume that the advection dominated condition (2.5) holds and assume also that the space $X^{\mathcal{N}}$ satisfies the following approximation property: for each $v \in H_0^1(\Omega) \cap H^{k+1}(\Omega)$ there exists $\hat{v}^{\mathcal{N}} \in X^{\mathcal{N}}$ such that*

$$\begin{aligned}\|v - \hat{v}^{\mathcal{N}}\|_{L^2(K)} + h_K\|\nabla(v - \hat{v}^{\mathcal{N}})\|_{L^2(K)} + h_K^2\|D^2(v - \hat{v}^{\mathcal{N}})\|_{L^2(K)} \\ \leq Ch_K^{r+1}|v|_{H^{r+1}(K)}\end{aligned} \tag{2.28}$$

for each $K \in \mathcal{T}_h$. Then the SUPG method has the following order of convergence:

$$\|u - u^{\mathcal{N}}\|_{SUPG} \leq Ch^{r+\frac{1}{2}}|u|_{H^{r+1}(K)} \tag{2.29}$$

provided that $u \in H^{r+1}(K)$.

Proof. First of all we point out that if we choose the approximation space (2.17), for any $v \in H_0^1(\Omega) \cap H^{k+1}(\Omega)$ there exist an element $\hat{v}^{\mathcal{N}}$ fulfilling the condition (2.28). Indeed, it is possible to choose

$$\hat{v}^{\mathcal{N}} = \Pi_h^r v \tag{2.30}$$

where Π_h^r is the piecewise polynomial interpolation operator (see [43] for the definition).

We can now define

$$\sigma^{\mathcal{N}} := u^{\mathcal{N}} - \hat{u}^{\mathcal{N}}, \quad \eta := u - \hat{u}^{\mathcal{N}}. \tag{2.31}$$

and we observe that

$$u - u^{\mathcal{N}} = \eta - \sigma^{\mathcal{N}}. \tag{2.32}$$

We want now to estimate the quantity $\|\sigma^{\mathcal{N}}\|_{SUPG}$. We observe that, from Theorem 2.1.1 and Galerkin orthogonality, we have

$$\frac{1}{2}\|\sigma^{\mathcal{N}}\|_{SUPG}^2 \leq a(\sigma^{\mathcal{N}}, \sigma^{\mathcal{N}}) + s^{(0)}(\sigma^{\mathcal{N}}, \sigma^{\mathcal{N}}) = a(\eta, \sigma^{\mathcal{N}}) + s^{(0)}(\eta, \sigma^{\mathcal{N}}). \tag{2.33}$$

In order to make effective estimates, we write explicitly the right-hand side of (2.33):

$$\begin{aligned}a(\eta, \sigma^{\mathcal{N}}) + s^{(0)}(\eta, \sigma^{\mathcal{N}}) = &\varepsilon \int_\Omega \nabla\eta \cdot \nabla\sigma^{\mathcal{N}} + \int_\Omega \boldsymbol{\beta} \cdot \nabla\eta\, \sigma^{\mathcal{N}} \\ &+ \sum_{K \in \mathcal{T}_h} \delta_K \left(-\varepsilon\Delta\eta + \boldsymbol{\beta} \cdot \nabla\eta, \frac{h_K}{|\boldsymbol{\beta}|}(\boldsymbol{\beta} \cdot \sigma^{\mathcal{N}} + \frac{1}{2}\operatorname{div}\boldsymbol{\beta}\sigma^{\mathcal{N}}) \right)\end{aligned} \tag{2.34}$$

Let us start by estimating the first term of the sum:

$$\varepsilon \int_\Omega \nabla\eta \cdot \nabla\sigma^{\mathcal{N}} \leq \frac{\varepsilon}{8}\|\nabla\sigma^{\mathcal{N}}\|_{L^2(\Omega)}^2 + 2\varepsilon\|\nabla\eta\|_{L^2(\Omega)}^2. \tag{2.35}$$

As regards the second term, we have to rewrite it a little before making effective estimates:

$$\begin{aligned}\int_\Omega \boldsymbol{\beta}\cdot\nabla\eta\,\sigma^{\mathcal{N}} = &-\int_\Omega \frac{1}{2}\eta(\operatorname{div}\boldsymbol{\beta})\sigma^{\mathcal{N}}\\ &-\sum_{K\in\mathcal{T}_h}\int_K\sqrt{\frac{h_K\delta_K}{|\boldsymbol{\beta}|}}\left(\boldsymbol{\beta}\cdot\nabla\sigma^{\mathcal{N}}+\frac{1}{2}(\operatorname{div}\boldsymbol{\beta})\sigma^{\mathcal{N}}\right)\sqrt{\frac{|\boldsymbol{\beta}|}{h_K\delta_K}}\eta,\end{aligned} \tag{2.36}$$

then, by using the Young inequality, we obtain:

$$\int_\Omega \boldsymbol{\beta}\cdot\nabla\eta\,\sigma^{\mathcal{N}} \le \frac{1}{4}\|\sigma^{\mathcal{N}}\|^2_{SUPG}+2\|\eta\|^2_{L^2(\Omega)}+2\sum_{K\in\mathcal{T}_h}\left\|\sqrt{\frac{|\boldsymbol{\beta}|}{h_K\delta_K}}\,\eta\right\|^2_{L^2(K)}. \tag{2.37}$$

Finally, we have to care about the stabilization term in (2.34). Also in this case we can resort to the Young inequality and obtain

$$\begin{aligned}&\sum_{K\in\mathcal{T}_h}\delta_K\left(-\varepsilon\Delta\eta+\boldsymbol{\beta}\cdot\nabla\eta,\frac{h_K}{|\boldsymbol{\beta}|}(\boldsymbol{\beta}\cdot\sigma^{\mathcal{N}}+\frac{1}{2}(\operatorname{div}\boldsymbol{\beta})\sigma^{\mathcal{N}})\right)\\ &\qquad\le\frac{1}{8}\|\sigma^{\mathcal{N}}\|^2_{SUPG}+2\sum_{K\in\mathcal{T}_h}\delta_K\int_K\frac{h_K}{|\boldsymbol{\beta}|}(-\varepsilon\Delta\eta+\boldsymbol{\beta}\cdot\nabla\eta)^2\\ &\le\frac{1}{8}\|\sigma^{\mathcal{N}}\|^2_{SUPG}+4\sum_{K\in\mathcal{T}_h}\delta_K h_K\left\|\frac{\varepsilon}{\sqrt{|\boldsymbol{\beta}|}}\Delta\eta\right\|^2_{L^2(K)}+4\sum_{K\in\mathcal{T}_h}\delta_K h_K\|\nabla\eta\|^2_{L^2(K)}\\ &\le\frac{1}{8}\|\sigma^{\mathcal{N}}\|^2_{SUPG}+\sum_{K\in\mathcal{T}_h}\delta_K\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}h_K^3\,\|\Delta\eta\|^2_{L^2(K)}+4\sum_{K\in\mathcal{T}_h}\delta_K h_K\|\nabla\eta\|_{L^2(K)}\end{aligned} \tag{2.38}$$

and in the last inequality we used the advection dominated condition (2.5). From inequalities (2.33), (2.35), (2.37), (2.38) we obtain that

$$\begin{aligned}\frac{1}{8}\|\sigma^{\mathcal{N}}\|^2_{SUPG}\le 2\varepsilon\|\nabla\eta\|^2_{L^2(\Omega)}+2\|\eta\|^2_{L^2(\Omega)}+2\sum_{K\in\mathcal{T}_h}\left\|\sqrt{\frac{|\boldsymbol{\beta}|}{h_K\delta_K}}\,\eta\right\|^2_{L^2(K)}\\ +\sum_{K\in\mathcal{T}_h}\delta_K\|\boldsymbol{\beta}\|_{L^\infty(\Omega)}h_K^3\,\|\Delta\eta\|^2_{L^2(K)}+4\sum_{K\in\mathcal{T}_h}\delta_K h_K\|\nabla\eta\|^2_{L^2(K)}.\end{aligned} \tag{2.39}$$

Exploiting the condition (2.28) and the inverse inequality (2.23), we obtain that

$$\|\sigma^{\mathcal{N}}\|_{SUPG}\le Ch^{r+\frac{1}{2}}|u|_{H^{r+1}(\Omega)}. \tag{2.40}$$

Using again the condition (2.28), we easily have that

$$\|\eta\|_{SUPG}\le Ch^{r+\frac{1}{2}}|u|_{H^{r+1}(\Omega)}. \tag{2.41}$$

Thus, by triangular inequality we can conclude that

$$\|u-u^{\mathcal{N}}\|_{SUPG}\le\|\eta\|_{SUPG}+\|\sigma^{\mathcal{N}}\|_{SUPG}\le Ch^{r+\frac{1}{2}}|u|_{H^{r+1}(\Omega)}. \tag{2.42}$$

□

Remark 2.1.2. As pointed out in [14], if the advection dominated condition (2.5) is not fulfilled for all $K \in \mathcal{T}$ we locally lose even the h^r convergence rate of the standard FE method. To recover at least the standard convergence rate, we need that the coefficient C_K in

$$\sum_{K \in \mathcal{T}_h} C_K \left\| \Delta \eta \right\|^2_{L^2(K)} \tag{2.43}$$

in (2.38) must scale, at least, as h^2. A possible way to overcome this trouble is to act on the weights δ_K, distinguishing between the elements for which $\mathbb{P}\mathtt{e}_K > 1$ and $\mathbb{P}\mathtt{e}_K \leq 1$. Unfortunately, in a RB context, this strategy does not allow an immediate affine representation (1.9) of the bilinear form. We will return on this point in section 2.2.2.

2.2 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* stabilization method and an *offline-only* stabilized one when used to approximate the solution of an advection-diffusion problem:

$$-\varepsilon(\boldsymbol{\mu})\Delta u(\boldsymbol{\mu}) + \boldsymbol{\beta}(\boldsymbol{\mu}) \cdot \nabla u(\boldsymbol{\mu}) = 0 \quad \text{on } \Omega_o(\boldsymbol{\mu}) \subset \mathbb{R}^2.$$

Offline-Online means that the Galerkin projections are performed, in both Offline and Online stage, with respect to the SUPG stabilized bilinear form that is

$$\begin{aligned} a_{stab}(w^{\mathcal{N}}, v^{\mathcal{N}}) = \int_\Omega \varepsilon \nabla w^{\mathcal{N}} \cdot \nabla v^{\mathcal{N}} + (\boldsymbol{\beta} \cdot \nabla w^{\mathcal{N}}) v^{\mathcal{N}} \\ + \sum_{K \in \mathcal{T}_h} \delta_K \int_K (-\varepsilon \Delta w^{\mathcal{N}} + \boldsymbol{\beta} \cdot \nabla w^{\mathcal{N}})(h_K \frac{\boldsymbol{\beta}}{|b\beta|} \cdot \nabla v^{\mathcal{N}}) \end{aligned} \tag{2.44}$$

with $w^{\mathcal{N}}, v^{\mathcal{N}} \in X^{\mathcal{N}} \subseteq \mathbb{P}^r(\mathcal{T}_h)$, where $\mathcal{T}_h$ is a triangulation of Ω. This is a bilinear coercive form, 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 (2.44) 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, to avoid the Online stabilization.

In both these approaches we have to provide an *affine* expansion like (1.9) of the involved bilinear forms and right-hand side operators. If this is not possible in an exact way, we could resort to some interpolation techniques (e.g. empirical interpolation [1, 10, 16, 34]). As we always need an affine expansion, the advantage of using the *Offline-only* method could be a certain reduction of the computational cost, that could be significant if the number of affine 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 2.2.1, does not represent a particular application, it is just a problem that shows strong instability effect that can be effectively fixed by the chosen stabilization method. The second test case, shown in section 2.2.2, is a Poiseuille-Graetz problem [27, 45].

Let us make a remark about the notation we will use from now on. First of all, we will write explicitly the FE space dimension $\mathcal{N}$ only when it will be necessary. Moreover, as we will use only the SUPG stabilization method, we will omit the value of ρ in the stabilization terms.

2.2.1 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 2.5. The problem is the following one:

$$\begin{cases} -\dfrac{1}{\mu}\Delta u(\mu) + (1,1)\cdot\nabla u(\mu) = 0 & \text{in } \Omega \\ u(\mu) = 0 & \text{on } \Gamma_1 \cup \Gamma_2 \\ u(\mu) = 1 & \text{on } \Gamma_3 \cup \Gamma_4 \end{cases} \tag{2.45}$$

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.

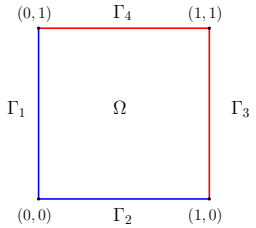


Figure 2.5: First testcase: domain. On the blue sides we impose $u = 0$, while on the red ones $u = 1$.

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

$$\begin{aligned} &\text{find } u(\mu) \in V := \{v \in H^1(\Omega) \mid v|_{\Gamma_1} = 1, v|_{\Gamma_0} = 0\} \text{ s.t.} \\ &a(u(\mu), v; \mu) = 0 \quad \forall v \in H^1_0(\Omega) \end{aligned} \tag{2.46}$$

where:

$$a(w, v; \mu) := \int_\Omega \frac{1}{\mu}\nabla w \cdot \nabla v + (\partial_x u + \partial_y u)\, v. \tag{2.47}$$

We know from the general theory of PDEs that the problem (2.46) admits a unique solution.

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

$$\begin{aligned} &\text{find } u_h(\mu) \in V_h := \{v_h \in \mathbb{P}^r(\mathcal{T}_h) \mid v_h|_{\Gamma_1} = 1, v_h|_{\Gamma_0} = 0\} \text{ s.t.} \\ &a(u_h(\mu), v^{\mathcal{N}}; \mu) = 0 \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \end{aligned} \tag{2.48}$$

with $X^{\mathcal{N}}$ defined as the subspace of $\mathbb{P}^r(\mathcal{T}_h)$ (see (2.17) for its definition) made up by the functions that vanish on the boundary of Ω. 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:

$$\begin{aligned} &\text{find } u^{\mathcal{N}}(\mu) \in X^{\mathcal{N}} \text{ s.t.} \\ &a(u^{\mathcal{N}}(\mu), v^{\mathcal{N}}; \mu) = F(v^{\mathcal{N}}) \quad \forall v \in X^{\mathcal{N}}. \end{aligned} \tag{2.49}$$

where

$$F(v^{\mathcal{N}}; \mu) := -a(g_h, v^{\mathcal{N}}; \mu). \tag{2.50}$$

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

$$\mathbb{P}\mathrm{e}_K := \frac{\mu\, h_K}{\sqrt{2}} < 1 \quad \forall K \in \mathcal{T}_h \tag{2.51}$$

that is when the advection dominated condition (2.5) is not fulfilled. In figure 2.6 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 (2.45) for a parameter range from 1 to 10. In figure 2.8 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, in 2.8 we show the linear interpolation of the RB approximation error computed for of 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 [45]. It is evident that the RB approximation error tends to vanish in correspondence of the parameter values selected by the greedy algorithm. This phenomenon is clearly expected because, since we are using Lagrange basis, our RB solution "interpolates" exactly the *truth* manifold (1.3) in the "interpolation nodes" represented by the *snapshot* solutions. In figure 2.7 we show some representative RB solution computed in correspondence of some value of the parameter μ. The dimension of the RB space is $N = 8$.

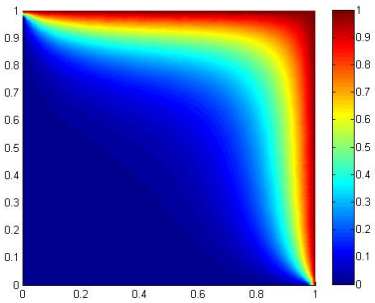
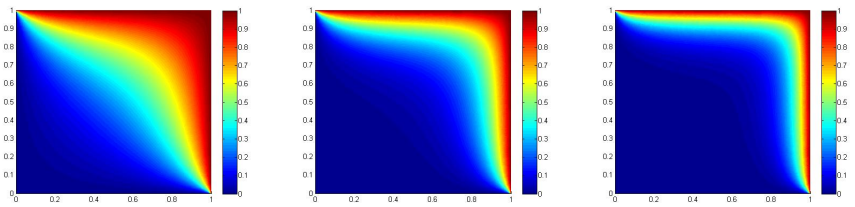


Figure 2.6: First test case, low Péclet number. FE solution for $\mu = 6$.



(a) $\mu = 2$ (b) $\mu = 6$ (c) $\mu = 10$

Figure 2.7: First test case, low Péclet number. representative RB solutions for different values of the parameter.

More interesting is when the Péclet number assumes higher values, for which the condition (2.51) is not fulfilled. In figure 2.9 the solution obtained by using a FE approximation with $\mu = 600$ 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 2.10. For this simple case, if we let the parameter range from 100 to 1000 the greedy

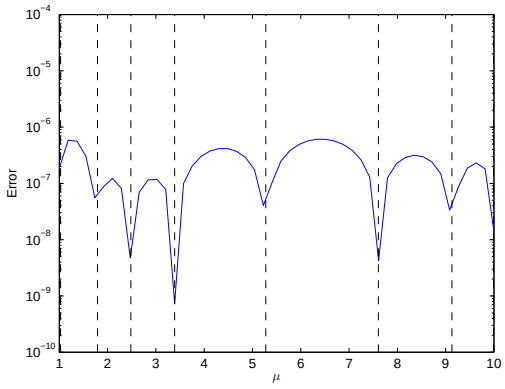


Figure 2.8: First test case, low Péclet number. RB approximation error as a function of the parameter.

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 2.11. 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.

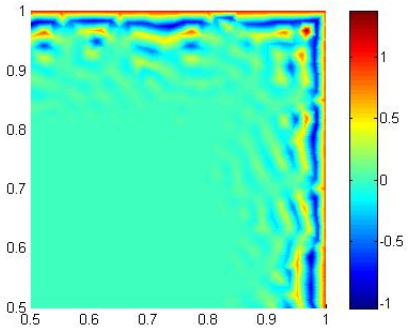


Figure 2.9: First test case, high Péclet number. FE solution for $\mu = 600$ (zoom on $[0.5, 1] \times [0.5, 1]$).

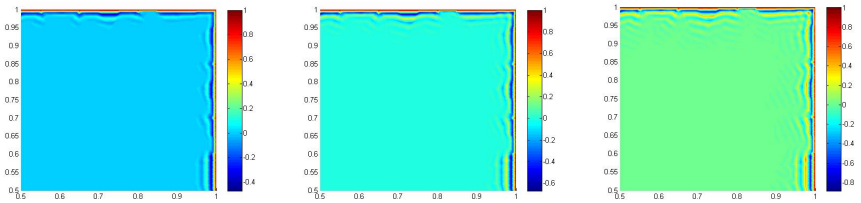
A possible way to fix this instability problems could be to use some stabilization methods. We chose to use the SUPG stabilization method. First of all, we have to impose the stabilization correction to the weak formulation (2.48). We thus define the following bilinear form:

$$s(w_h, v_h; \mu) := \sum_{K \in \mathcal{T}_h} \delta_K \int_K \left(-\frac{1}{\mu} \Delta w_h + (1,1) \cdot \nabla w_h \right) \left(\frac{h_K}{\sqrt{2}} (1,1) \cdot \nabla v_h \right) \tag{2.52}$$

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:

$$s(w_h, v_h; \mu) = \sum_{K \in \mathcal{T}_h} \frac{\delta_K h_K}{\sqrt{2}} \int_K (\partial_x w_h + \partial_y w_h)(\partial_x v_h + \partial_y v_h) \tag{2.53}$$

again with $w_h, v_h \in \mathbb{P}^r(\mathcal{T}_h)$. As we discussed in the section 2.1, for our choice of polynomial approximation space, we do not need to fulfil any requirement on the weights delta. We then



(a) $\mu = 400$ (b) $\mu = 600$ (c) $\mu = 800$

Figure 2.10: First test case, high Péclet number. representative RB solutions for different values of the parameter (zoom on $[0.5, 1] \times [0.5, 1]$).

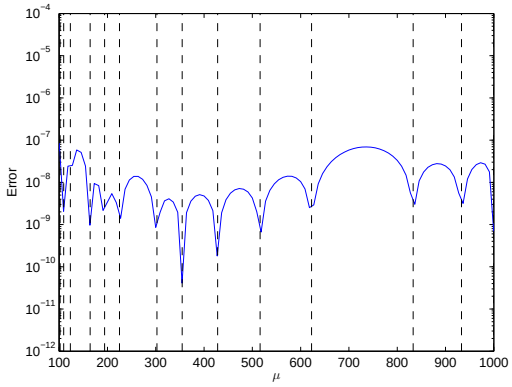


Figure 2.11: First test case, high Péclet number. RB approximation error expressed as a function of the parameter

set $\delta_K = 1$ for each element $K \in \mathcal{T}$ If we define $u^{s\mathcal{N}}(\mu) = u_h^s(\mu) - g_h$ we can obtain the final formulation of the stabilized FE problem:

$$\begin{aligned} &\text{find } u^{s\mathcal{N}}(\mu) \in X^{\mathcal{N}} \text{ s.t.} \\ &a(u^{s\mathcal{N}}(\mu), v^{\mathcal{N}}) + s(u^{s\mathcal{N}}(\mu), v^{\mathcal{N}}) = F(v^{\mathcal{N}}) + F^s(v^{\mathcal{N}}) \quad \forall v \in X^{\mathcal{N}}. \end{aligned} \tag{2.54}$$

where F is the same as in (2.50) and f^s is

$$F^s(v^{\mathcal{N}}) := -s(g_h, v^{\mathcal{N}}). \tag{2.55}$$

Let us call a_{stab} the bilinear form and f_{stab} the right-hand side, that is

$$\begin{aligned} a_{stab} &= a + s \\ F_{stab} &= F + F^s \end{aligned} \tag{2.56}$$

In figure 2.12 is shown a SUPG stabilized FE solution for $\mu = 600$.

Now we can try the two different approaches described before: the *Offline-Online* and the *Offline-only* stabilization. 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* 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 the space spanned by the reduced basis, the *Offline-Online* solution $u_N^s(\mu) \in X^N$ satisfies

$$a_{stab}(u_N^s(\mu), v_N; \mu) = F_{stab}(v_N) \quad \forall\, v_N \in X^N \tag{2.57}$$

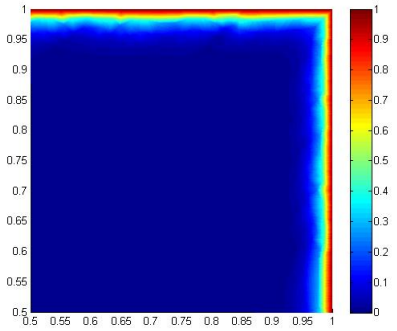


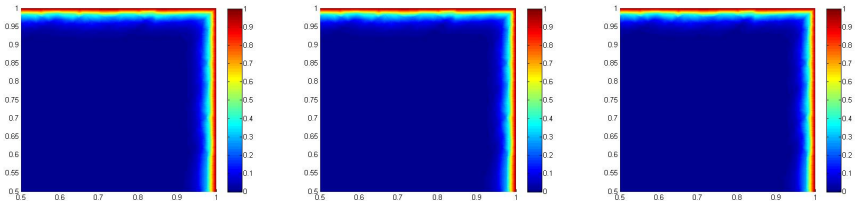
Figure 2.12: First test case, high Péclet number. SUPG solution for $\mu = 600$ (zoom on $[0.5, 1] \times [0.5, 1]$).

while the *Offline-only* solution $u_N(\mu) \in X^N$ is such that

$$a(u_N(\mu), v_N; \mu) = F(v_N) \quad \forall\, v \in X^N. \tag{2.58}$$

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.

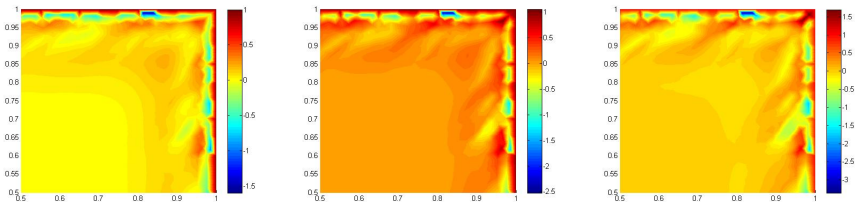
The *Offline-Online* method, as expected, produces stable RB solutions, as shown in figure 2.13, and the actual error, with respect to the stabilized FE solution, is smaller than the tolerance guaranteed by the greedy algorithm ($\varepsilon^*_{tol} = 10^{-5}$), as we can see in figure 2.15. On the contrary, the behaviour of the *Offline-only* approach is very unsatisfactory. As we can see in figure 2.14, even though the reduced basis are stable, the *Offline-only* RB solutions show large oscillations.



(a) $\mu = 400$ (b) $\mu = 600$ (c) $\mu = 800$

Figure 2.13: First test case, high Péclet number. representative *Offline-Online* RB solutions for different values of the parameter (zoom on $[0.5, 1] \times [0.5, 1]$).

In order to exclude the possibility that this instability is only due to the advection-diffusion operator (with high Péclet number) while we project on the reduced basis space, we set up an RB approximation by using a locally refined mesh. 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 (2.51) is now satisfied, at least where we actually have instability problems. Obviously, by increasing the number of degrees of freedom, we quite increase the computational cost. The Offline algorithm produces 14 basis and it takes 711 seconds while the stabilized Offline algorithm takes only 114 seconds and builds 8 basis. The RB



(a) $\mu = 400$ (b) $\mu = 600$ (c) $\mu = 800$

Figure 2.14: First test case, high Péclet number. representative *Offline-only* RB solutions for different values of the parameter (zoom on $[0.5, 1] \times [0.5, 1]$).

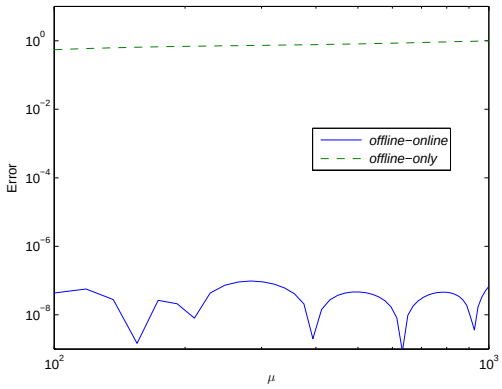


Figure 2.15: First test case, high Péclet number. comparison of the RB approximation error obtained for the two different stabilization strategies; here the error is expressed as a function of the parameter.

solutions obtained for the same parameter range as before (figure 2.16) do not show instability phenomena, so an explanation of the behaviour of the *Offline-only* method tested before has to be found analysing the "inconsistency" between the different methods used in the two stages, as we will do further. The distance in energy norm[1] between the FE solution and the RB one is showed in figure 2.17 (we recall that $\varepsilon^*_{tol} = 10^{-5}$). Comparing figure 2.16 and figure 2.13 we can also see how the stabilization method tends to "smooth" the boundary layer.

Before going on, in table 2.1, we report informations about the computations performed in this section. In all the numerical tests we used a tolerance $\varepsilon^*_{tol} = 10^{-3}$ on the greedy algorithm.

2.2.2 Graetz-Poiseuille test case

We now focus on a different situation, a Graetz problem [15, 27, 45, 51], in which we have two parameters: one physical (the Péclet number) and one geometrical (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 $\boldsymbol{\mu} = (\mu_1, \mu_2)$ with 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 2.18.

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

[1] i.e. the norm induced by the symmetric part of the original advection-diffusion bilinear form.

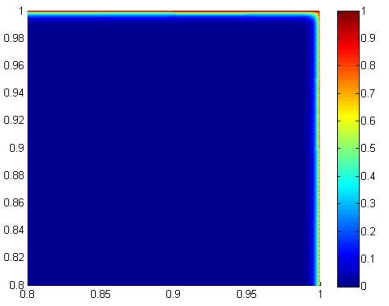


Figure 2.16: First test case, high Péclet number. RB solution with locally refined mesh for $\mu = 400$ (zoom on $[0.8, 1] \times [0.8, 1]$).

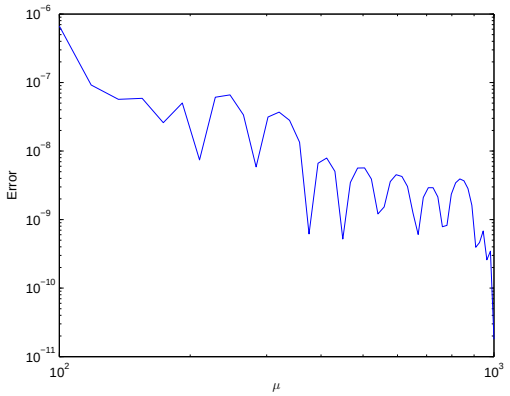


Figure 2.17: First test case, high Péclet number. RB approximation error for locally refined mesh, expressed as a function of the parameter.

$\mathcal{D}$	$\mathcal{N}$	Offline stab.	Offline time (s)	N	Online stab.	Online time (s)
$[100, 1000]$	2605	yes	114	8	no	$1.78 \cdot 10^{-3}$
					yes	$1.95 \cdot 10^{-3}$
$[1, 10]$	2605	no	86	14	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 2.1: First time dependent test case. Numerical tests

that:

$$\begin{cases} -\dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + 4\,y(1-y)\partial_x u(\boldsymbol{\mu}) = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_{o\,1}(\boldsymbol{\mu}) \cup \Gamma_{o\,2}(\boldsymbol{\mu}) \cup \Gamma_{o\,6}(\boldsymbol{\mu}) \\ u(\boldsymbol{\mu}) = 1 & \text{on } \Gamma_{o\,3}(\boldsymbol{\mu}) \cup \Gamma_{o\,5}(\boldsymbol{\mu}) \\ \dfrac{\partial u}{\partial \nu} = 0 & \text{on } \Gamma_{o\,4}(\boldsymbol{\mu}). \end{cases} \tag{2.59}$$

In order to use an 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

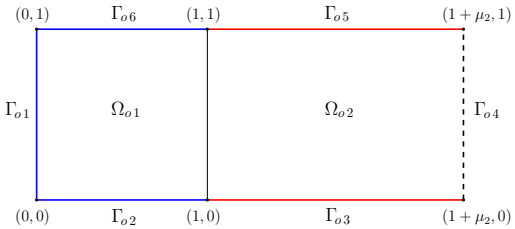


Figure 2.18: Graetz problem. parametrized domain. Buondary conditions: homogeneous Dirichlet on the blue sides, $u = 1$ on the red sides, homogeneous Neumann on the dashed side.

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, as we have seen in section 1.1.1, we need the affine transformation that maps the reference domain into the original one [45, 50], so we define:

$$\begin{aligned} &T^1(\boldsymbol{\mu})\colon \Omega^1 \to \mathbb{R}^2 \\ &T^1\left(\begin{pmatrix} x \\ y \end{pmatrix}; \boldsymbol{\mu}\right) = \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned} \tag{2.60}$$

that is the identity map, and

$$\begin{aligned} &T^2(\boldsymbol{\mu})\colon \Omega^1 \to \mathbb{R}^2 \\ &T^2\left(\begin{pmatrix} x \\ y \end{pmatrix}; \boldsymbol{\mu}\right) = \mathbf{G}^2 \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \mu_2 \\ 0 \end{pmatrix} \end{aligned} \tag{2.61}$$

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.

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

$$\begin{aligned} &\text{find } u_o(\boldsymbol{\mu}) \in V_o := \{v_o \in H^1(\Omega_o) \mid v|_{\Gamma_o^{d,1}} = 1, v|_{\Gamma_o^{d,0}} = 0\} \text{ s.t.} \\ &a(u_o(\mu), v_o; \boldsymbol{\mu}) = 0 \quad \forall v \in H_0^1(\Omega) \end{aligned} \tag{2.62}$$

where:

$$a(w_o, v_o; \boldsymbol{\mu}) := \int_{\Omega_o} \frac{1}{\mu_1} \nabla w \cdot \nabla v + 4\, y(1-y) \partial_x u\, v. \tag{2.63}$$

We set the standard FE problem, exactly as we did in (2.48), introducing then 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:

$$s(w_h, v_h; \boldsymbol{\mu}) := \sum_{K_o(\boldsymbol{\mu}) \in \mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \int_{K_o(\boldsymbol{\mu})} (4\, y(1-y) \partial_x w_h) \left(h_{K_o(\boldsymbol{\mu})} \partial_x v_h\right) \tag{2.64}$$

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

$$\begin{aligned} &\text{find } u(\boldsymbol{\mu}) \in V := \{v_h \in \mathbb{P}^1(\Omega) \mid v_h|_{\Gamma^{d,1}} = 1, v_h|_{\Gamma^{d,0}} = 0\} \text{ s.t.} \\ &a(u_h(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + s(u_h(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = 0 \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \end{aligned} \tag{2.65}$$

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

$$\begin{aligned} a(w_h, v_h; \boldsymbol{\mu}) := &\int_{\Omega^1} \frac{1}{\mu_1} \nabla w_h \cdot \nabla v_h + 4\, y(1-y) \partial_x w_h\, v_h \\ &+ \int_{\Omega^2} \frac{1}{\mu_1 \mu_2} \partial_x w_h \partial_y v_h + \frac{\mu_2}{\mu_1} \partial_x w_h \partial_y v_h + 4\mu_2\, y(1-y) \partial_x w_h\, v_h \end{aligned} \tag{2.66}$$

and s is:

$$\begin{aligned} s(w_h, v_h; \boldsymbol{\mu}) := &\sum_{K \in \mathcal{T}_h^1} h_K \int_K (4\, y(1-y) \partial_x w_x)\, \partial_x v_h \\ &+ \sum_{K \in \mathcal{T}_h^2} \frac{h_K}{\sqrt{\mu}_2} \int_K (4\, y(1-y) \partial_x w_x)\, \partial_x v_h. \end{aligned} \tag{2.67}$$

By introducing a lifting of the Dirichlet boundary condition we can obtain the stabilized FE formulation (2.54). 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 $h_{K_o(\boldsymbol{\mu})} \approx h_K \sqrt{J(\boldsymbol{\mu})} = h_K \sqrt{\mu_2}$. 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 not make any 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 forgets about the triangulation.

Recalling remark 2.1.2, we want to observe that by using a weighting that depends on both parameter and element size we lose the affinity assumption (1.9) 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 (2.5) 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 [14]) we probably need to exploit some interpolation techniques involving the empirical interpolation [1][2]. In this case it would be also worth to be checked if it were possible to define a weighting that does not depend on each h_K, but on the mesh size h, under suitable regularity assumptions [31].

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 arise in an area in which the norm of the advection field

[2]We have to remark that the weighting proposed in [14] is discontinuous in both $\boldsymbol{x}$ and $\boldsymbol{\mu}$ even if the coefficients ε and $\boldsymbol{\beta}$ are smooth.

(an thus the value of the local Péclet number) is relatively small. In figure 2.19 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.

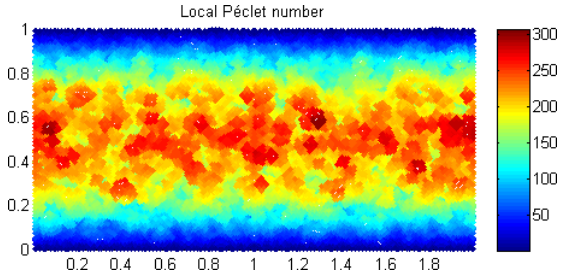


Figure 2.19: Local Péclet number computed in the quadrature points of the reference domain ($\mu_2 = 1.$)

In figure 2.20 we show a solution computed using the standard FE method wich shows instability, while in figures 2.21 and 2.22 we show some solutions obtained respectively by *Offline-Online* stabilization and *Offline-only* stabilization.

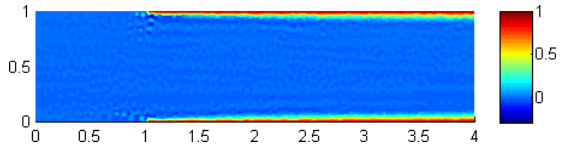
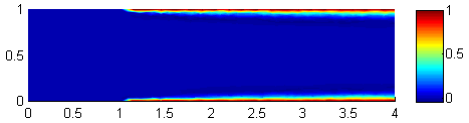
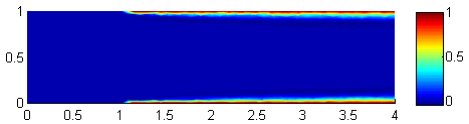


Figure 2.20: Graetz-Poiseuille test case. FE solution (non-stabilized) for $\boldsymbol{\mu} = (15000, 3)$



(a) $\mu_1 = 12500$



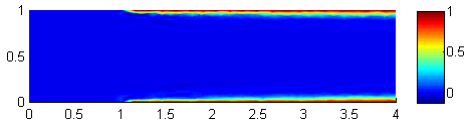
(b) $\mu_1 = 17500$

Figure 2.21: Graetz-Poiseuille test case. Representative *Offline-Online* RB solutions for $\mu_2 = 3$ and several values of μ_1.

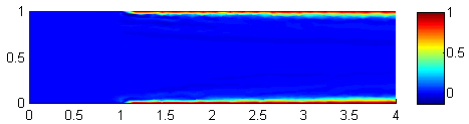
Finally, in figure 2.23 we show the error curves of the two methods. As in the previous test-case, we can see that only the *Offline-Online* stabilization produces satisfactory results, even if in this case the *Offline-only* methods has slightly better performances. Here we used a tolerance[3] on the greedy algorithm $\varepsilon^*_{tol} = 10^{-3}$.

In table 2.2, we report some figures about the numerical tests performed using the parameter space $\mathcal{D} = [10000, 20000] \times [0.5, 4]$. The tolerance for the greedy algorithm is $\varepsilon^*_{tol} = 10^{-3}$.

[3]The tolerance is on the stabilized energy norm, that is greater than the non-stabilized one.

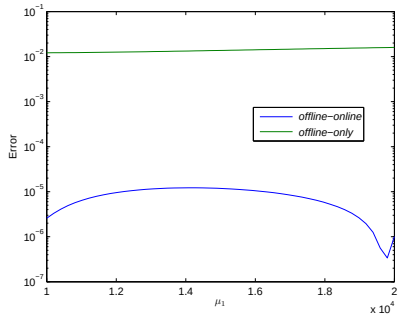


(a) $\mu_1 = 12500$

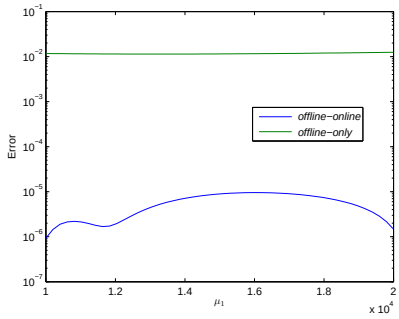


(b) $\mu_1 = 17500$

Figure 2.22: Graetz-Poiseuille test case. Representative *Offline-only* RB solutions for $\mu_2 = 3$ and several values of μ_1.



(a) $\mu_2 = 2.5$



(b) $\mu_2 = 3.5$

Figure 2.23: Graetz-Poiseuille 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.

$\mathcal{N}$	Offline stab.	Offline time (s)	N	Online stab.	Online time (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 2.2: First time dependent test case. Numerical tests

2.2.3 Discussion on the results

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

Let us introduce some notation. Let us call *energy norm* the norm on $H_0^1(\Omega(\boldsymbol{\mu}))$ 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:

$$|||\cdot|||_{\boldsymbol{\mu}} = \sqrt{a^{sym}(\cdot,\cdot;\boldsymbol{\mu})}$$
$$|||\cdot|||_{\boldsymbol{\mu},stab} = \sqrt{a^{sym}_{stab}(\cdot,\cdot;\boldsymbol{\mu})}.$$

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* one. This is because of the fact the greedy algorithm guarantees that

$$|||u_N^s(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu},stab} \leq \varepsilon_{tol}^* \quad \forall \boldsymbol{\mu} \in \Xi_{train}{}^4$$

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

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:

$$|||u_N(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq |||u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} + |||u_N^s(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}. \tag{2.68}$$

Of course, it holds that

$$|||u_N^s(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq |||u_N^s(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu},stab} \leq \varepsilon_{tol}^*$$

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

$$\begin{aligned} |||u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}^2 =& a(u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu}), u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ =& F^s(u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})) - s(u_N^s(\boldsymbol{\mu}), u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ =& -s(u_N^s(\boldsymbol{\mu}) + g_h, u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu}); \boldsymbol{\mu}) \end{aligned} \tag{2.69}$$

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

$$\begin{aligned} |s(u_N^s(\boldsymbol{\mu}) + g_h, u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu}))| \leq& h_{max} \|\beta \cdot \nabla(u_N^s(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ & \cdot \left\| \frac{\beta}{|\beta|} \cdot \nabla(u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})) \right\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ \leq& h_{max} \|\beta \cdot \nabla(u_N^s(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ & \cdot |u_N(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})|_{H_0^1(\Omega_o(\boldsymbol{\mu}))}. \end{aligned}$$

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 $|\cdot|_{H_0^1}$, we have that:

$$|v|_{H_0^1(\Omega_o(\boldsymbol{\mu}))} \leq C(\mu) \, |||v|||_{\boldsymbol{\mu}} \quad \forall v \in H_0^1(\Omega_o(\boldsymbol{\mu})). \tag{2.70}$$

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:

$$\begin{aligned} \|\beta \cdot \nabla(u_N^s(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \leq& \|\beta \cdot \nabla(u_h^s(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ & + \|\beta \cdot \nabla(u_N^s(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ \leq& \|\beta \cdot \nabla(u_h^s(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} + C(\boldsymbol{\mu}) \|\beta\|_{L^\infty(\Omega_o(\boldsymbol{\mu}))} \varepsilon_{tol}^*. \end{aligned} \tag{2.71}$$

Putting together the results obtained above, we can write a proper upper bound for the distance with respect to the energy norm between $u_N(\boldsymbol{\mu})$ and $u^{s\mathcal{N}}(\boldsymbol{\mu})$:

$$\begin{aligned} |||u_N(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq& h_{max}(\boldsymbol{\mu}) C(\boldsymbol{\mu}) \|\beta \cdot \nabla(u_h^s(\boldsymbol{\mu}))\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ & + \left(1 + h_{max}(\boldsymbol{\mu}) C(\boldsymbol{\mu})^2 \|\beta\|_{L^\infty(\Omega_o(\boldsymbol{\mu}))}\right) \varepsilon_{tol}^*. \end{aligned} \tag{2.72}$$

[4] Ξ_{train} is the set of values of $\boldsymbol{\mu}$ used by the greedy algorithm to choose the basis. See section 1.2.2.

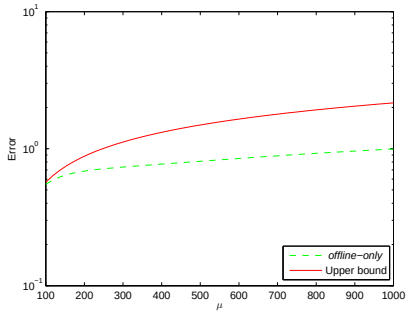


Figure 2.24: First test case. upper bound (2.73) compared to the true error

Remark 2.2.1. We point out that the bound in (2.72) depends on the L^2 norm of the streamline derivative. This means that the *Offline-only* 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 Graetz-Poiseuille problem. The "improvement" of the approximation is confirmed by the numerical results shown in figures 2.15 and 2.23.

Remark 2.2.2. A very similar computation shows that the error $|||u_N^s(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$ is actually the same as $|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}$.

We can also note that the argument used to prove the a priori estimate (2.72) actually proves the following, sharper but not properly "a priori", result:

$$\begin{aligned} |||u_N(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \leq h_{max}(\boldsymbol{\mu})\, C(\boldsymbol{\mu}) \|\beta \cdot \nabla(u_N^s(\boldsymbol{\mu}) + g_h)\|_{L^2(\Omega_o(\boldsymbol{\mu}))} \\ + |||u_N^s(\boldsymbol{\mu}) - u^{s\,\mathcal{N}}(\boldsymbol{\mu})|||_{\boldsymbol{\mu}}. \end{aligned} \tag{2.73}$$

We performed some numerical tests for the bound in (2.73). The results are shown in figures 2.24 and 2.25. Concerning the first test case we set:

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

and for the Graetz-Poiseuille problem:

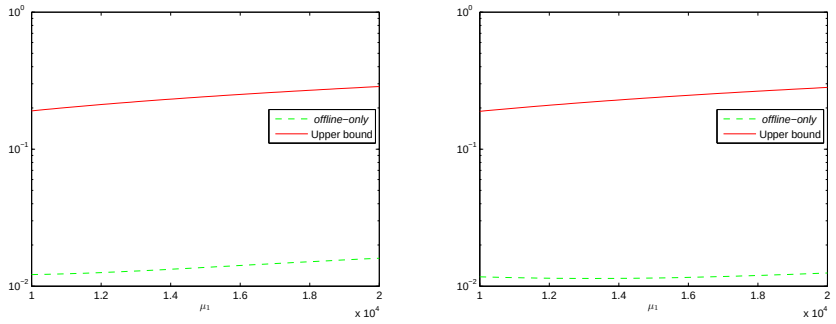
$$C(\boldsymbol{\mu}) = \sqrt{\mu_1}.$$

With these choices, (2.70) is actually an equality.

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

The sharpness obtained for at least one test case leads us to state that the *Offline-only* approach is not a good approximation method. One problem is that the Offline stage, as described in chapter 1, is tailored to minimize the error between the *Offline-Online* solution and the stabilized FE one.

At last, we want to show that the upper bound for the first test case remains sharp even if we reduce the mesh size, as we did at the end of the first section. Even if it would not be needed, we can use a stabilized method to perform the Offline stage with a very fine mesh. This situation is formally the same as in the case "coarse mesh - Offline stabilization", so it make sense to test if the upper bound (2.73) remains sharp. In figure 2.26 we show the error between the *Offline-only* RB solution and the the stabilized FE one.



(a) $\mu_2 = 2.5$ (b) $\mu_2 = 3.5$

Figure 2.25: Graetz-Poiseuille test case. upper bound (2.73) compared to the true error.

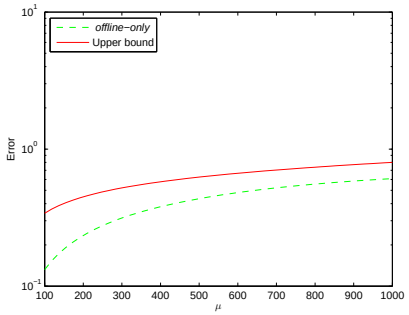


Figure 2.26: First test case. upper bound (2.73) compared to the true error

2.3 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 an effective 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 [21]. Let Ω be the unit square in $\mathbb{R}^2$, as sketched in figure 2.27, and let us define $\boldsymbol{\mu} = (\mu_1, \mu_2)$, where $\mu_1, \mu_2 \in \mathbb{R}$. The problem is the following one:

$$\begin{cases} -\dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + (\cos\mu_2, \sin\mu_2)\cdot\nabla u(\boldsymbol{\mu}) = 0 & \text{in } \Omega \\ u(\boldsymbol{\mu}) = 1 & \text{on } \Gamma_1 \cup \Gamma_2 \\ u(\boldsymbol{\mu}) = 0 & \text{on } \Gamma_3 \cup \Gamma_4 \cup \Gamma_5. \end{cases} \tag{2.74}$$

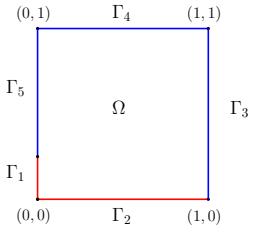


Figure 2.27: Domain of the problem (2.74). On the blue sides we impose $u = 0$, while on the red ones $u = 1$.

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:

$$a(w, v; \boldsymbol{\mu}) = \int_\Omega \frac{1}{\mu_1}\nabla w \cdot \nabla v + (\cos\mu_2\, \partial_x w + \sin\mu_2\, \partial_y w)v. \tag{2.75}$$

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:

$$\begin{aligned} s^r(w_h, v_h; \boldsymbol{\mu}) = &- \left(\sum_{K\in\mathcal{T}_h} \delta_K^r \int_K \frac{1}{\mu_1}\Delta w_h \ (\cos\mu_2, \sin\mu_2)\cdot\nabla v_h \right)^{r-1} \\ &+ \sum_{K\in\mathcal{T}_h} \delta_K^r \int_K (\cos\mu_2, \sin\mu_2)\cdot\nabla w_h \ (\cos\mu_2, \sin\mu_2)\cdot\nabla v_h. \end{aligned} \tag{2.76}$$

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

Acting as we did in section 2.2, 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:

$$\begin{aligned} &\text{find } u^{s\mathcal{N}_r}(\boldsymbol{\mu}) \in X^{\mathcal{N}_r} \text{ s.t.} \\ &a^r_{stab}(u^{s\mathcal{N}_r}(\boldsymbol{\mu}), v^{\mathcal{N}}) = F^r_{stab}(v^{\mathcal{N}_r}) \quad \forall v \in X^{\mathcal{N}_r}. \end{aligned} \tag{2.77}$$

where $X^{\mathcal{N}_r}$, a^r_{stab} and F^r_{stab} are defined as in (2.54) and (2.56) (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 2.1.2, if $r = 1$ we do not have any restriction on the weights, so we choose

$$\delta_K^1 = 1 \quad \forall K \in \mathcal{T}_h. \tag{2.78}$$

On the contrary, if $r = 2$, the stability result we saw in section 2.1 (Theorem 2.1.3) shows that δ_K^2 has to be sufficiently small. To set properly the weights, we follow the choice proposed in [14]. First of all we need to slightly redefine the "element size" h_K, as suggested in [20]:

$$h_K^2 = \frac{4A_K}{\sqrt{3\sum_{i=1}^3 |\boldsymbol{x}_{i,K} - \boldsymbol{x}_{c,K}|^2}} \quad \forall K \in \mathcal{T}_h \tag{2.79}$$

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:

$$\mathbb{P}\mathrm{e}_K(\boldsymbol{x}, \boldsymbol{\mu}) = \frac{|\boldsymbol{\beta}(\boldsymbol{x}, \boldsymbol{\mu})| h_K}{C_2\, \varepsilon(\boldsymbol{x})} \quad \forall \boldsymbol{x} \in \tag{2.80}$$

where C_2 is the constant of the inverse inequality (2.23). It can be proved that, by defining the element size as in (2.79), the best value for the constant C_2 is 48 [20]. Finally, we set:

$$\delta_K^2 = \frac{1}{2} \quad \forall K \in \mathcal{T}_h. \tag{2.81}$$

It can also be proved that a stability condition like (2.26) and theorem 2.1.3 still hold, even with the latter definitions [14].

We did different choices also for the tolerance ε_{tol}^* of the greedy algorithm. We recall, at first, that theorem 2.1.3 states 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 $h^{\frac{5}{2}}$. Here h stands for the mesh size

$$h = \max_{K \in \mathcal{T}_h} h_K \tag{2.82}$$

using, as h_K, either the element diameter or the quantity defined in (2.79) 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

$$\begin{aligned} \|u(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})\|_{SUPG} &\leq \|u(\boldsymbol{\mu}) - u^{s\mathcal{N}}(\boldsymbol{\mu})\|_{SUPG} + |||u^{s\mathcal{N}}(\boldsymbol{\mu}) - u_N^s(\boldsymbol{\mu})|||_{\boldsymbol{\mu}} \\ &\leq C(u(\boldsymbol{\mu}), \boldsymbol{\mu}) h^{k+\frac{1}{2}} + \varepsilon_{tol}^*. \end{aligned} \tag{2.83}$$

Therefore we choose a value for the tolerance ε_{tol}^* of the same order of magnitude as the FE approximation error.

In table 2.3 we report some figures about the tests we have performed. The first observation is that the variations of the advection direction have more influence on the dimension N of the reduced basis than the variations of the Péclet number. This is because by varying the direction of the advection field, the solution shows strong variations in energy norm. We think that it is worth to be noted also that if we reduce the mesh size, increasing then the number of the degrees of freedom $\mathcal{N}$, the number N of basis function that we need to achieve the same tolerance increases too. Our (heuristic) explanation of this phenomenon is that, by

r	$\mu_1 \in$	$\mu_2 \in$	$\mathcal{N}$	Offline time (s)	h	ε^*_{tol}	N
1	$\{10^4\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2605	458	0.03	10^{-3}	21
1	$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2605	649	0.03	10^{-3}	21
1	$[10^4, 10^5]$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2605	224	0.03	10^{-3}	37
1	$[10^4, 10^5]$	$\{\frac{\pi}{4}\}$	2605	131	0.03	10^{-3}	3
1	$\{10^4\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	10577	1170	0.015	10^{-3}	30
1	$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	10577	2036	0.015	10^{-3}	33
2	$\{10^4\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2605	829	0.06	10^{-4}	28
2	$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2605	3701	0.06	10^{-4}	29
2	$[10^4, 10^5]$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2605	2776	0.06	10^{-4}	65
2	$[10^4, 10^5]$	$\{\frac{\pi}{4}\}$	2605	189	0.06	10^{-4}	4
2	$\{10^4\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	10577	2188	0.03	10^{-4}	39
2	$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	10577	7382	0.03	10^{-4}	44

Table 2.3: Numerical tests

r	$\mathcal{N}$	N	RB Offline time (s)	RB Online time (s)	FE time (s)
1	2605	21	649	$2.15 \cdot 10^{-3}$	1.31
1	10577	33	2036	$2.56 \cdot 10^{-3}$	4.41
2	2605	29	3701	$2.48 \cdot 10^{-3}$	0.63
2	10577	44	7382	$3.57 \cdot 10^{-3}$	2.41

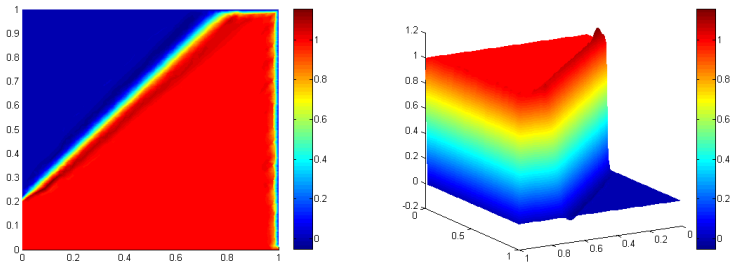
Table 2.4: Comparison of the computational times.

reducing the mesh size, we are able to capture more information about the sharp layers, but this means in turn that the "number" of the possible configurations of the system, depending on the parameter, rises. As a consequence, we will need more basis functions to obtain the same accuracy. This behaviour of the stabilized method have been highlighted also in [8].

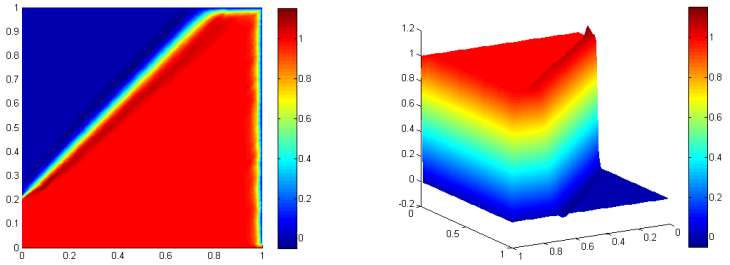
We note also that the Offline time is often much higher for the $\mathbb{P}^2$ approximation. This is because of the SCM, which needs more iterations in the $\mathbb{P}^2$ case than in the $\mathbb{P}^1$ one.

In figure 2.28, 2.29, 2.30 and 2.31 we show a visual comparison among the tested methods. All the RB approximations are obtained using a parameter space $\mathcal{D} = \{10^5\} \times [\frac{\pi}{6}, \frac{\pi}{3}]$. In table 2.4 we compare the RB and FE computational costs of the considered representative solutions[5].

[5]For the FE computational cost, we are not taking into account the meshing time. We are considering only the assembling and the resolution phases of the linear system.

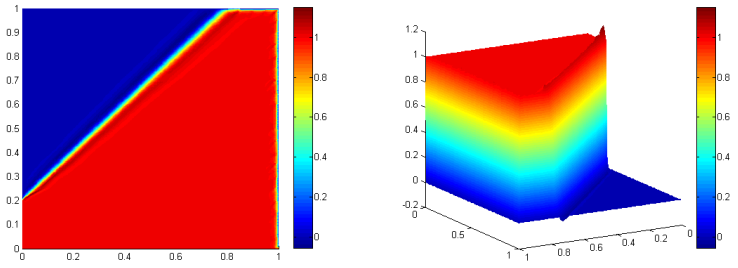


(a) $\mathbb{P}^1$-RB solution.

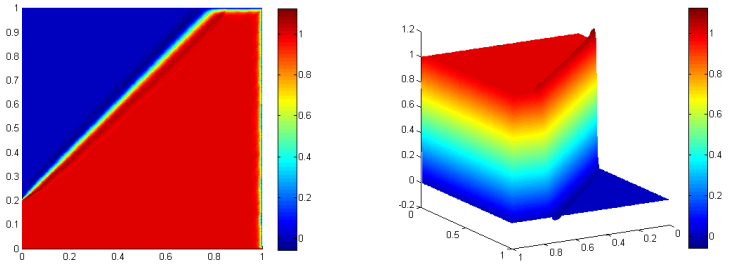


(b) $\mathbb{P}^1$-FE solution

Figure 2.28: $\mathbb{P}^1$-RB and $\mathbb{P}^1$-FE approximated solution of (2.74), with $\boldsymbol{\mu} = (10^5, \frac{\pi}{4})$ and mesh size $h = 0.03$ ($\mathcal{N} = 2605$).

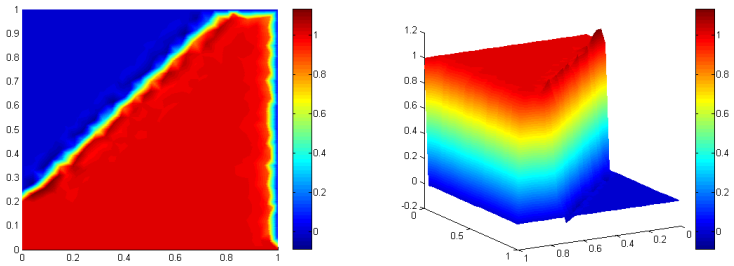


(a) $\mathbb{P}^1$-FRB solution

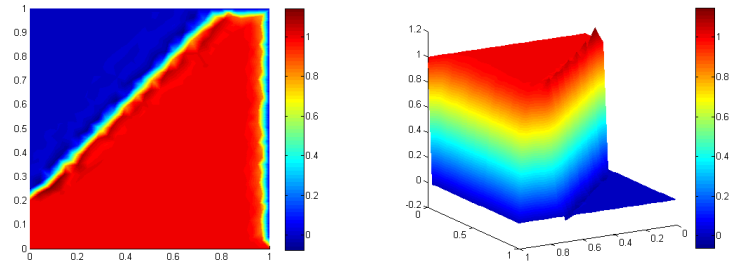


(b) $\mathbb{P}^1$-FE solution

Figure 2.29: $\mathbb{P}^1$-RB and $\mathbb{P}^1$-FE approximated solution of (2.74), with $\boldsymbol{\mu} = (10^5, \frac{\pi}{4})$ and mesh size $h = 0.015$ ($\mathcal{N} = 10577$).

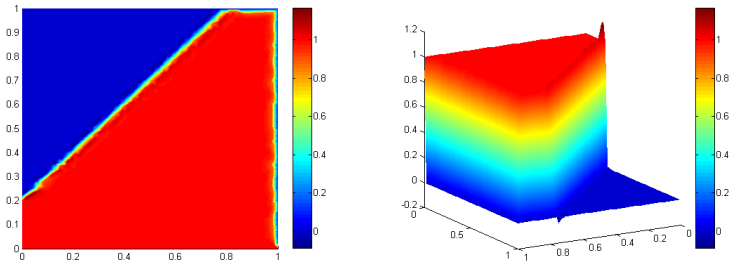


(a) $\mathbb{P}^1$-RB solution

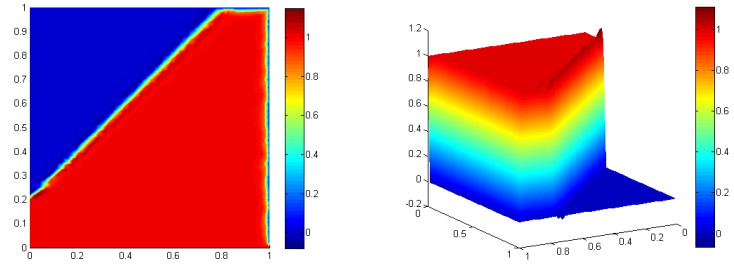


(b) $\mathbb{P}^1$-FE solution

Figure 2.30: $\mathbb{P}^1$-RB and $\mathbb{P}^1$-FE approximated solution of (2.74), with $\boldsymbol{\mu} = (10^5, \frac{\pi}{4})$ and mesh size $h = 0.06$ ($\mathcal{N} = 2605$).



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



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

Figure 2.31: $\mathbb{P}^1$-RB and $\mathbb{P}^1$-FE approximated solution of (2.74), with $\boldsymbol{\mu} = (10^5, \frac{\pi}{4})$ and mesh size $h = 0.03$ ($\mathcal{N} = 10577$).

Chapter 3

Stabilized reduced basis method for time-dependent problems

In this chapter we want to apply to time dependent problems the stabilized RB method introduced for steady problems in the previous chapter. The RB method for time dependent problem has been already studied in several works, e.g. [15, 45, 49, 51], 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.

It section 3.1 we introduce the general RB setting for parabolic problems, highlighting the *a posteriori* error estimation techniques and the sampling strategies, while in section 3.2 we show the SUPG stabilization method for parabolic problems. Finally, in section, 3.3, we show and discuss some numerical tests.

3.1 Reduced basis method for linear parabolic equations

Like in section 1.1, 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 $H_0^1(\Omega) \subset X \subset H^1(\Omega)$. For each admissible value of the parameter, i.e. for each $\boldsymbol{\mu} \in \mathcal{D}$, we define the continuous bilinear forms

$$\begin{aligned} &a(\cdot, \cdot; \boldsymbol{\mu})\colon X \times X \to \mathbb{R}, \\ &m(\cdot, \cdot; \boldsymbol{\mu})\colon L^2(\Omega) \times L^2(\Omega) \to \mathbb{R}. \end{aligned} \tag{3.1}$$

We suppose that the form a satisfies the coercivity and *affinity* assumptions (1.7) and (1.9), respectively. We assume also that the *mass* form m satisfies an *affinity* assumption like the following one:

$$m(v, w; \boldsymbol{\mu}) = \sum_{q=1}^{Q_m} \Theta_q^m(\boldsymbol{\mu}) m^q(v, w) \tag{3.2}$$

where, like in (1.9), $\Theta_q^m\colon \mathcal{D} \to \mathbb{R}$, $q = 1, \ldots, Q_m$, are smooth functions whereas $m^q\colon L^2(\Omega) \times L^2(\Omega) \to \mathbb{R}$, $q = 1, \ldots, Q_m$, 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 (1.10). Let us finally denote our time domain with $I = [0, T]$, where T is the final time.

We want to spend some words to explain why we need the *affine* assumption (3.2) also on the mass term. As we saw in chapter 1, the parameter can be *geometrical*, that is the original domain of the problem $\Omega_o(\boldsymbol{\mu})$ might depend on the parameter. As in section 1.1.1, let us then suppose that both the original domain and the reference one are divided in subdomains, like in (1.17) and (1.18). The original *mass* form, that is the L^2 scalar product on the original domain, becomes:

$$m_o(v_o, w_o; \boldsymbol{\mu}) = \sum_{l=1}^{L_{dom}} \int_{\Omega_o^l(\boldsymbol{\mu})} v_o \, w_o \quad \forall \, v_o, w_o \in X_o(\boldsymbol{\mu}) \tag{3.3}$$

where $X_o(\boldsymbol{\mu})$ is the original test function space. Tracking back the latter integrals on the reference domain Ω trough the map T defined in (1.20) and (1.21), as we did in section 1.1.2 for the bilinear form a_o, we obtain:

$$m(v, w; \boldsymbol{\mu}) = \sum_{l=1}^{L_{dom}} \int_{\Omega^l} v \, w \, J^l(\boldsymbol{\mu}) \quad \forall v, w \in X. \tag{3.4}$$

where $J^l(\boldsymbol{\mu})$ is the (local) Jacobian of the transformation T.

We can now define our continuous problem:

$$\begin{aligned} &\text{find } u(\cdot; \boldsymbol{\mu}) \in C^0(I; L^2(\Omega)) \cap L^2(I; X) \text{ s.t.} \\ &m(\partial_t u(t; \boldsymbol{\mu}), v) + a(u(t; \boldsymbol{\mu}), v; \boldsymbol{\mu}) = g(t) F(v; \boldsymbol{\mu}) \quad \forall v \in X, \quad \forall t \in I \\ &\text{given the initial value } u(0; \boldsymbol{\mu}) = u_0 \in L^2(\Omega). \end{aligned} \tag{3.5}$$

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:

$$\begin{cases} \partial_t u(\boldsymbol{\mu}) + L u(\boldsymbol{\mu}) = h & \text{in } \Omega \\ u(\cdot, t; \boldsymbol{\mu}) = 0 & \text{on } \partial\Omega, \, \forall t \in I \\ & + \quad \text{initial conditions} \end{cases} \tag{3.6}$$

where L is a differential operator and $h \in L^2(\Omega \times I)$. If we suppose that $h(x, t) = g(t) f(x)$ for each $(x, t) \in \Omega \times I$, with $g \in L^2(I)$ and $f \in L^2(\Omega)$, we obtain a weak formulation like (3.5). Another situation can be the following one:

$$\begin{cases} \partial_t u(\boldsymbol{\mu}) + L u(\boldsymbol{\mu}) = 0 & \text{in } \Omega \\ u(\cdot, t; \boldsymbol{\mu}) = h(\cdot, t) & \text{on } \partial\Omega, \, \forall t \in I \\ & + \quad \text{initial conditions} \end{cases} \tag{3.7}$$

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 $h(x, t) = g(t) f(x)$ for each $(x, t) \in \Omega \times I$, 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 (3.5) in which the functional F is given by:

$$F(v) = -a(\tilde{f}, v) \quad \forall v \in X, \tag{3.8}$$

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

3.1.1 Discretization and RB formulation

To discretize the time-dependent problem (3.5) we follow the approach used in [17, 40, 45] that is to use finite differences in time and FE in space discretization [48].

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

$$\begin{aligned}
&\text{for each } t \in I, \text{ find } u^{\mathcal{N}}(t;\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\
&m(\partial_t u^{\mathcal{N}}(t;\boldsymbol{\mu}), v^{\mathcal{N}};\boldsymbol{\mu}) + a(u^{\mathcal{N}}(t;\boldsymbol{\mu}), v^{\mathcal{N}};\boldsymbol{\mu}) = g(t)F(v^{\mathcal{N}}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.} \\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.
\end{aligned} \tag{3.9}$$

To obtain a fully discretized problem, we subdivide the time interval I into J subintervals of length $\Delta t = T/J$ and we define $t^j = j\Delta t$, $j = 1, \dots, J$. We then replace the time derivative in (3.9) with a backward finite difference approximation. The fully discretized problem we are considering is:

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\
&\frac{1}{\Delta t} m(u^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}};\boldsymbol{\mu}) + a(u^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}};\boldsymbol{\mu}) = g(t^j)F(v^{\mathcal{N}}) \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.} \\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.
\end{aligned} \tag{3.10}$$

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

$$\boldsymbol{u}^{\mathcal{N}}(\boldsymbol{\mu}) = (u^{\mathcal{N}\,1}(\boldsymbol{\mu}), \dots, u^{\mathcal{N}\,J}(\boldsymbol{\mu})) \in \left(X^{\mathcal{N}}\right)^J. \tag{3.11}$$

The latter problem is the *Backward Euler-Galerkin* discretization of (3.5). Of course, this is not the only way to discretize the time-dependent problem (3.5), for example we can resort to other theta-methods (e.g. Crank-Nicholson) or to higher order methods [48].

The RB formulation of the problem (3.10) is based on hierarchical RB spaces as we did in the steady case in section 1.2.1, that is: given an integer N_{max} we define a finite sequence $\{X_N^{\mathcal{N}}\}_{N=1}^{N_{max}}$ of subspaces of $X^{\mathcal{N}}$ such that (1.31) holds. The way these subspaces are generated will be described in detail in section 3.1.2, for the moment let us say that the basis functions of $X_N^{\mathcal{N}}$ are built by properly combining *snapshots* in time and space. As in chapter 1, we use the following notation

$$X_N^{\mathcal{N}} = \operatorname{span}\{\zeta_n^{\mathcal{N}} \mid 1 \le n \le N\}. \tag{3.12}$$

We assume also that the functions $\zeta_n^{\mathcal{N}}$ are mutually orthonormal with respect to the scalar product $(\cdot,\cdot)_X$.

The RB problem is then:

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}} \text{ s.t.} \\
&\frac{1}{\Delta t} m(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u_N^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v_N;\boldsymbol{\mu}) + a(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}), v_N;\boldsymbol{\mu}) = g(t^j)F(v_N) \quad \forall v_N \in X_N^{\mathcal{N}}, \\
&\text{given the initial condition } u_N^{\mathcal{N}\,0} \text{ s.t.} \\
&(u_N^{\mathcal{N}\,0}, v_N)_{L^2(\Omega)} = (u^{\mathcal{N}\,0}, v_N)_{L^2(\Omega)} \quad \forall v_N \in X_N^{\mathcal{N}}.
\end{aligned} \tag{3.13}$$

Again, as in (3.11), we define

$$\boldsymbol{u}_N^{\mathcal{N}}(\boldsymbol{\mu}) = (u_N^{\mathcal{N}\,1}(\boldsymbol{\mu}), \dots, u_N^{\mathcal{N}\,J}(\boldsymbol{\mu})) \in \left(X^{\mathcal{N}}\right)^J. \tag{3.14}$$

Let us try to obtain the matrix formulation of the RB problem (3.13). First of all, we recall that, for each $j = 1, \dots, J$, the RB solution $u_N^{\mathcal{N}\,j} \in X_N^{\mathcal{N}}$ can be expressed as:

$$u_N^{\mathcal{N}\,j} = \sum_{m=1}^{N} u_{N\,m}^{\mathcal{N},j}(\boldsymbol{\mu})\zeta_m^{\mathcal{N}}. \tag{3.15}$$

Then, by taking $v_N = \zeta_n^N$, $n = 1, \dots, N$, in the RB formulation (3.13) we have

$$\frac{1}{\Delta t} m(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}), \zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) + a(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}), \zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) = g(t^j)F(\zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) + \frac{1}{\Delta t} m(u_N^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), \zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) \tag{3.16}$$

that is, recalling the affine assumptions (1.9), (1.10) and (3.2):

$$\begin{aligned}\sum_{m=1}^{N} \left(\frac{1}{\Delta t} \sum_{q=1}^{Q_m} \Theta_m^q(\boldsymbol{\mu}) m(\zeta_m^N, \zeta_n^N) + \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a(\zeta_m^N, \zeta_n^N) \right) u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}) \\ = g(t^j) \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) F^q(\zeta_n^N) + \sum_{m=1}^{N} \left(\frac{1}{\Delta t} \sum_{q=1}^{Q_m} \Theta_m^q(\boldsymbol{\mu}) m^q(\zeta_m^N, \zeta_n^N) \right) u_N^{\mathcal{N}\,j-1}(\boldsymbol{\mu}).\end{aligned} \tag{3.17}$$

We can thus obtain the matrix formulation:

$$\begin{aligned}\left(\frac{1}{\Delta t} \sum_{q=1}^{Q_m} \Theta_m^q(\boldsymbol{\mu}) \mathbf{M}_N^q + \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbf{A}_N^q \right) \mathbf{u}_N^j(\boldsymbol{\mu}) \\ = g(t^j) \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) \mathbf{F}^q + \left(\frac{1}{\Delta t} \sum_{q=1}^{Q_m} \Theta_m^q(\boldsymbol{\mu}) \mathbf{M}_N^q \right) \mathbf{u}_N^{j-1}(\boldsymbol{\mu}).\end{aligned} \tag{3.18}$$

where $\mathbf{A}_N$, $\mathbf{F}_N$ are defined in (1.42) and (1.44), while

$$\left(\mathbf{u}_N^{\mathcal{N},j}(\boldsymbol{\mu})\right)_m = u_{N\,m}^j(\boldsymbol{\mu}), \quad \left(\mathbf{M}_N^q\right)_{nm} = m^q(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}), \tag{3.19}$$

for $n, m = 1, \dots, N$ and $j = 1, \dots, J$. Denoting again with $\mathcal{Z}$ the $\mathcal{N} \times \mathcal{N}$ matrix whose columns are the coordinates of the reduced basis $\zeta_1^{\mathcal{N}}, \dots, \zeta_N^{\mathcal{N}}$ with respect to $\{\varphi_i\}_{i=1}^{\mathcal{N}}$, it holds that:

$$\mathbf{M}_N^q = \mathcal{Z}^{\mathrm{T}} \mathbf{M}_{\mathcal{N}}^q \mathcal{Z} \quad 1 \le q \le Q_m \tag{3.20}$$

where $\mathbf{M}_{\mathcal{N}}^q$ is the q-th affine term of the FE mass matrix, that is

$$\left(\mathbf{M}_{\mathcal{N}}^q\right)_{ij} = m^q(\varphi_j, \varphi_i) \quad 1 \le i, j \le \mathcal{N}, \quad 1 \le q \le Q_m. \tag{3.21}$$

Like in the steady case, during the Offline stage we have to compute and store the FE matrices, the *snapshots* solutions and the RB matrices. The only difference between the time-dependent case and the steady case is that in the former we have also to deal with the matrices associated to the mass term, which arise as a consequence of the time-dependency.

The Online operation count is the following [40, 45]:

- $O((Q_a + Q_m)N^2)$ to get the left-hand side matrix;
- $O(Q_F N + Q_m N^2)$ to compute the right-hand side;
- $O(N^3 + JN^2)$ to perform a factorization of the left-hand side matrix (e.g. LU factorization [47]) and to solve the J linear systems (3.18);
- $O(JN)$ to perform the scalar products (3.15).

Once again, we stress the point that the Online computational cost is independent of the dimension $\mathcal{N}$ of the underlying FE element *truth* approximation.

3.1.2 Sampling strategy and *a posteriori* error estimates

To construct the reduced basis in the time-dependent case, we will follow the so called *POD-greedy* approach [19, 40, 45]. It consists in using a greedy technique to explore the parameter space $\mathcal{D}$ and the POD (Proper Orthogonal Decomposition) method to deal with the time evolution.

Before describing in detail such a sampling strategy, for the sake of completeness, we will give a brief general introduction to the POD method.

POD method

As described for example in [32], given K elements w_k, $k = 1, \dots, K$, in a linear space W, the POD method returns M functions $\chi_m \in W$, $m = 1, \dots, M$, with $M < K$,that are orthonormal with respect to a given scalar product $(\cdot, \cdot)$ and such that the space

$$\mathcal{P}_M = \text{span}\{\chi_m \,|\, 1 \leq m \leq M\} \tag{3.22}$$

is optimal in the sense that:

$$\mathcal{P}_M = \underset{Y_M \subset \text{span}\{w_k \,|\, 1 \leq k \leq K\}}{\text{argmin}} \left(\frac{1}{K} \sum_{k=1}^{K} \inf_{v \in Y_M} \|w_k - v\|^2 \right)^{\frac{1}{2}}, \tag{3.23}$$

where Y_M denotes a M-dimensional linear space and $\|\cdot\|$ is the norm induced by the scalar product $(\cdot, \cdot)$. We note that it also holds:

$$\inf_{v \in Y_M} \|w_k - v\|^2 = \|w_k - \pi_{Y_M} v\|^2 \tag{3.24}$$

where π_{Y_M} is the orthogonal projection on Y_M, with respect to the scalar product $(\cdot, \cdot)$.

We show now an effective procedure to compute the orthonormal basis $\{\chi_m \,|\, 1 \leq m \leq M\}$ [18, 32].

1. Given the vector system $\{w_k \,|\, 1 \leq k \leq K\}$, we compute the $K \times K$ symmetric and positive definite matrix C^{POD} defined by

$$C_{ij}^{POD} = \frac{1}{K}(w_i, w_j). \tag{3.25}$$

2. We compute the first M eigenvalues λ_m, $m = 1, \dots, M$, of C^{POD} and the correspondent eigenvectors $\{\psi_m, 1 \leq m \leq M\}$.

3. We obtain the orthonormal basis $\{\chi_m \,|\, 1 \leq m \leq M\}$ using the formula:

$$\chi_m = \frac{1}{\sqrt{\lambda_m}} \sum_{k=1}^{K} (\psi_m)_k \, w_k \quad 1 \leq m \leq M, \tag{3.26}$$

where $(\psi_m)_k$ is the k-th component of the eigenvector ψ_m.

As regards the error, we can define

$$\mathcal{E}_M := \frac{1}{K} \sum_{k=1}^{K} \|w_k - \pi_{\mathcal{P}_M} w_k\|^2 \tag{3.27}$$

and the following equality holds [32]:

$$\mathcal{E}_M = \sum_{k=M+1}^{K} \lambda_k. \tag{3.28}$$

It was already obvious from the definition that $\mathcal{E}_M \to 0$ as M increases, but the equality (3.28) turns out to be useful for the choice of M. If we fix a tolerance ε_{tol}^{POD}, we can set M as:

$$M = \min \left\{ \widetilde{M} \;\middle|\; \left(\mathrm{tr}(C^{POD}) - \sum_{k=1}^{\widetilde{M}} \lambda_k \right)^{\frac{1}{2}} \leq \varepsilon_{tol}^{POD} \right\} \tag{3.29}$$

where $\mathrm{tr}(C^{POD})$ is the trace of the matrix C^{POD}.

To indicate the POD procedure, we adopt the following compact notation:

$$\{\chi_m \,|\, 1 \leq m \leq M\} = POD(\{w_k \,|\, 1 \leq k \leq K\}, M). \tag{3.30}$$

POD-greedy method

We introduce now the POD-greedy method [19, 45] used to build the reduced basis for the time dependent problem (3.10). First of all, let us define the norm:

$$|||\boldsymbol{v}^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} = \left(m(v^{\mathcal{N}\,J}(\boldsymbol{\mu}), v^{\mathcal{N}\,J}(\boldsymbol{\mu}); \boldsymbol{\mu}) + \sum_{j=1}^{J} a(v^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}\,j}(\boldsymbol{\mu}); \boldsymbol{\mu}) \Delta t \right)^{\frac{1}{2}} \tag{3.31}$$

for all sequences $\boldsymbol{v}^{\mathcal{N}}(\boldsymbol{\mu}) = (v^{\mathcal{N}\,1}(\boldsymbol{\mu}), \ldots, v^{\mathcal{N}\,J}(\boldsymbol{\mu})) \in \left(X^{\mathcal{N}}\right)^J$.

Let us denote with $\boldsymbol{e}(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in \mathcal{D}$, the difference between the *truth* solution $\boldsymbol{u}^{\mathcal{N}}(\boldsymbol{\mu})$ and the RB one $\boldsymbol{u}_N^{\mathcal{N}}(\boldsymbol{\mu})$. In order to pursue an effective greedy strategy, as we did in section 1.2.2, we assume that we have a sharp and computationally inexpensive *a posteriori* error estimator $\boldsymbol{\mu} \mapsto \Delta_N^t(\boldsymbol{\mu})$ such that

$$|||\boldsymbol{e}(\boldsymbol{\mu})|||_{t-dep} \leq \Delta_N^t(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \quad 1 \leq N \leq N_{max}. \tag{3.32}$$

Like in the steady case, section 1.2.2, we define a finite subset Ξ_{train} of $\mathcal{D}$, large enough to be considered an approximation of the parameter space $\mathcal{D}$.

The N-th step of the POD algorithm can be roughly described as follows:

1. find the value $\tilde{\boldsymbol{\mu}} \in \Xi_{train}$ that maximize the estimator Δ^t_{N-1} ;
2. compute the FE solution of the problem (3.10), taking $\boldsymbol{\mu} = \tilde{\boldsymbol{\mu}}$;
3. apply the POD method to $\{u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \,|\, 1 \le j \le J\}$ and obtain $\{\chi_m \,|\, 1 \le m \le M\}$;
4. apply the POD method to $\{\chi_m \,|\, 1 \le m \le M\} \cup \{\zeta^{\mathcal{N}}_n \,|\, 1 \le n \le N-1\}$, where $\{\zeta^{\mathcal{N}}_n \,|\, 1 \le n \le N-1\}$ is the already computed reduced basis;
5. set the output of the latter POD as the new reduced basis.

Concerning the stopping criteria, we will use the same used in the steady case, that is:

- *Prescribed tolerance on the greedy.* Given a tolerance ε^*_{tol}, the POD-greedy stops if

$$\Delta^t_N(\boldsymbol{\mu}) \le \varepsilon^*_{tol}. \tag{3.33}$$

- *Maximum RB dimension.* The algorithm stops if N reaches the value N_{max}, even if the tolerance ε^*_{tol} is not satisfied.

The POD-greedy procedure [40, 45] is shown in algorithm 3. In this algorithm there are two "tuning" values, M_1 and M_2, to be chosen. As regards M_1, we can set a POD tolerance ε^{POD}_{tol} and then use (3.29). The parameter M_2 has to be set less than M_1, otherwise, at each step, the second POD would not be effective (the dimension of the "reduced" space would be greater then the initial one).

Algorithm 3 POD-greedy algorithm

$\mathcal{Z} = \emptyset$;
$S = \{\boldsymbol{\mu}_0\}$; $\boldsymbol{\mu}* = \boldsymbol{\mu}_0$;
while $N \le N_{max}$ **do**
  $\{\chi_m \,|\, 1 \le m \le M_1\} = POD(\{u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \,|\, 1 \le j \le J\}, M_1)$;
  $\mathcal{Z} = \mathcal{Z} \cup \{\chi_m \,|\, 1 \le m \le M\}$;
  $N = N + M_2$;
  $\{\xi_n \,|\, 1 \le n \le N\} = POD(\mathcal{Z}, N)$;
  $X_N = \text{span}\{\zeta_n \,|\, 1 \le n \le N\}$;
end while.

For the sake of completeness, we must observe that the *a posteriori* error estimator Δ^t_N should depend also on the control function g. Then, we have to choose a particular control function with which the greedy algorithm is performed [17]. Recalling that our problem is linear, for every input control function g we can recover the solution $\boldsymbol{u}^{\mathcal{N}}(\boldsymbol{\mu})$ by convolution

$$u^{\mathcal{N}\,\tilde{\jmath}}(\boldsymbol{\mu}) = \sum_{j=1}^{\tilde{\jmath}} g(t^j)\tilde{u}^{\mathcal{N}\,\tilde{\jmath}-j}(\boldsymbol{\mu}) \tag{3.34}$$

where $\tilde{\boldsymbol{u}}^{\mathcal{N}}(\boldsymbol{\mu})$ is the *impulse response*, that is the solution of (3.10) in which it is used an *impulse control* $\tilde{g}$, such that:

$$\tilde{g}(t^0) = 1, \quad \tilde{g}(t^j) = 0 \quad 1 \le j \le J.^1 \tag{3.35}$$

[1] In the continuous problem, an *impulse conrol function* is a Dirac delta centered in $t = 0$.

It is important to note, in (3.34), that the function $u^{\mathcal{N}\,\tilde{\jmath}}(\boldsymbol{\mu})$, for $\tilde{\jmath} = 1, \dots, J$, is a linear combination of the impulse response $\tilde{u}^j(\boldsymbol{\mu})$, $j = 1, \dots, J$. This means that, to obtain a good approximation of the solution corresponding to any control function g, it is sufficient that the RB method approximates well the (parametric) impulse response [17].

As a consequence of the presence of an "unknown" control function that can be set Online, theoretically the error $\boldsymbol{e}(\boldsymbol{\mu})$ should depend also on g. We must point out that the following inequality is not valid:

$$|||\boldsymbol{e}(\boldsymbol{\mu}; g)|||_{t-dep} \leq \varepsilon^*_{tol} \quad \forall \boldsymbol{\mu} \in \Xi_{train} \tag{3.36}$$

for every choice of g, but we can provide error estimators such that:

$$|||\boldsymbol{e}(\boldsymbol{\mu}; g)|||_{t-dep} \leq \Delta^t_N(\boldsymbol{\mu}; g) \quad \forall \boldsymbol{\mu} \in \Xi_{train}, \tag{3.37}$$

for every choice of g.

Improvements to the POD-greedy algorithm have been proposed in e.g. [12].

A posteriori error estimates

We are dealing now with the *a posteriori* error estimators to be used in the greedy algorithm. In our work we will follow the choice presented in [17], but other possibility have recently been proposed [54, 55].

The first ingredient we have to introduce is the dual norm of the residual:

$$\varepsilon_N(t^j; \boldsymbol{\mu}; g) := \sup_{v^{\mathcal{N}} \in X^{\mathcal{N}}} \frac{r_N(v^{\mathcal{N}}; t^j; \boldsymbol{\mu}; g)}{\|v^{\mathcal{N}}\|_X}, \quad 1 \leq j \leq J, \tag{3.38}$$

where r_N is the residual of the RB approximation, that is:

$$\begin{aligned} r_N(v^{\mathcal{N}}; t^j; \boldsymbol{\mu}; g) = {} & g(t^j) f(v^{\mathcal{N}}) - \frac{1}{\Delta t} m(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u_N^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) - a(u_N^{\mathcal{N}\,j}(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) \\ & \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, 1 \leq j \leq J. \end{aligned} \tag{3.39}$$

Exploiting the affine assumptions, (1.9), (1.10) and (3.2), it can be shown [17] that

$$\begin{aligned} \varepsilon_N(t^j; \boldsymbol{\mu}; g)^2 = {} & \sum_{q,q'=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) \Theta_F^{q'}(\boldsymbol{\mu}) g(t^j)^2 \Lambda_{qq'}^{FF} \\ & + \sum_{q=1}^{Q_F} \sum_{n=1}^{N} \Theta_F^q(\boldsymbol{\mu}) g(t^j) \Bigg(\sum_{q'=1}^{Q_a} \Theta_a^{q'}(\boldsymbol{\mu}) u_{N\,n}^j(\boldsymbol{\mu}) \Lambda_{qq'\,n}^{aF} \\ & + \sum_{q'=1}^{Q_m} \Theta_m^{q'}(\boldsymbol{\mu}) \left(u_{N\,n}^j(\boldsymbol{\mu}) - u_{N\,n}^{j-1}(\boldsymbol{\mu}) \right) \Lambda_{qq'\,n}^{mF} \Bigg) \\ & + \sum_{n,n'=1}^{N} \Bigg\{ \sum_{q,q'=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \Theta_a^{q'}(\boldsymbol{\mu}) u_{N\,n}^j(\boldsymbol{\mu}) u_{N\,n'}^j(\boldsymbol{\mu}) \Lambda_{qnq'n'}^{aa} \\ & + \sum_{q,q'=1}^{Q_m} \left(u_{N\,n}^j(\boldsymbol{\mu}) - u_{N\,n}^{j-1}(\boldsymbol{\mu}) \right) \left(u_{N\,n'}^j(\boldsymbol{\mu}) - u_{N\,n'}^{j-1}(\boldsymbol{\mu}) \right) \Lambda_{qnq'n'}^{mm} \\ & + \sum_{q=1}^{Q_a} \sum_{q'=1}^{Q_m} \Theta_a^q(\boldsymbol{\mu}) \Theta_m^{q'}(\boldsymbol{\mu}) u_{N\,n}^j(\boldsymbol{\mu}) \left(u_{N\,n'}^j(\boldsymbol{\mu}) - u_{N\,n'}^{j-1}(\boldsymbol{\mu}) \right) \Lambda_{qnq'n'}^{am} \Bigg\} \end{aligned} \tag{3.40}$$

where the $\boldsymbol{\mu}$-independent quantities $\Lambda_{\cdot}^{\cdot}$ are defined as:

$$\begin{aligned}
&\Lambda_{qq'}^{FF} = (\mathcal{B}_q, \mathcal{B}_{q'})_X, && 1 \le q, q' \le Q_F;\\
&\Lambda_{qq'n}^{aF} = -2(\mathcal{B}_q, \mathcal{A}_{q',n})_X, && 1 \le q \le Q_F, 1 \le q' \le Q_a, 1 \le n \le N;\\
&\Lambda_{qq'n}^{mF} = -\tfrac{2}{\Delta t}(\mathcal{B}_q, \mathcal{M}_{q',n})_X, && 1 \le q \le Q_F, 1 \le q' \le Q_m, 1 \le n \le N;\\
&\Lambda_{qnq'n'}^{aa} = (\mathcal{A}_{q,n}, \mathcal{A}_{q',n'})_X, && 1 \le q, q' \le Q_a, 1 \le n, n' \le N;\\
&\Lambda_{qnq'n}^{mF} = \tfrac{2}{\Delta t}(\mathcal{A}_{q,n}, \mathcal{M}_{q',n})_X, && 1 \le q \le Q_a, 1 \le q' \le Q_m, 1 \le n, n' \le N;\\
&\Lambda_{qnq'n'}^{mm} = (\mathcal{M}_{q,n}, \mathcal{M}_{q',n'})_X, && 1 \le q, q' \le Q_m, 1 \le n, n' \le N;
\end{aligned} \tag{3.41}$$

in which $\mathcal{B}_q \in X^{\mathcal{N}}$, $\mathcal{A}_{q,n} \in X^{\mathcal{N}}$ and $\mathcal{M}_{q,n} \in X^{\mathcal{N}}$ are the solutions of the following $\boldsymbol{\mu}$-independent problems:

$$\begin{aligned}
&(\mathcal{B}_q, v^{\mathcal{N}})_X = F^q(v^{\mathcal{N}}), && \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \text{for } 1 \le q \le Q_F,\\
&(\mathcal{A}_{q,n}, v^{\mathcal{N}})_X = a^q(\zeta_n^{\mathcal{N}}, v^{\mathcal{N}}), && \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \text{for } 1 \le q \le Q_a, 1 \le n \le N,\\
&(\mathcal{M}_{q,n}, v^{\mathcal{N}})_X = m^q(\zeta_n^{\mathcal{N}}, v^{\mathcal{N}}), && \forall v^{\mathcal{N}} \in X^{\mathcal{N}} \text{for } 1 \le q \le Q_m, 1 \le n \le N,
\end{aligned} \tag{3.42}$$

We then need again a lower bound $\boldsymbol{\mu} \mapsto \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})$ for the coercivity constant of the (discretized) bilinear form a, as in (1.51). To do so we can resort to the SCM [25] introduced in section 1.2.4 and used in the steady case.

After these preliminaries, we are finally able to define the *a posteriori* error estimator, which satisfies (3.32):

$$\Delta_N^t(\boldsymbol{\mu}; g) = \left(\frac{\Delta t}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})} \sum_{j=1}^{J} \varepsilon_N(t^j; \boldsymbol{\mu}; g)^2 \right)^{\frac{1}{2}}. \tag{3.43}$$

The *a posteriori* error estimator used during the greedy is actually

$$\Delta_N^t(\boldsymbol{\mu}) := \Delta_N^t(\boldsymbol{\mu}; \tilde{g}), \tag{3.44}$$

where $\tilde{g}$ is the impulse control defined in (3.35).

3.2 SUPG stabilization method for time dependent problems

In this section we briefly introduce the SUPG method for time-dependent problems [3, 4, 31]. The idea is the same of the steady case: we have to add terms like $s^{(0)}$ and $\phi^{(0)}$, defined in (2.18) to the left-hand side and to the right-hand one of (3.9), respectively. More precisely, the right-hand side term is the very same, whereas we have to slightly redefine the term $s^{(0)}$ in order to consider the time dependence and to guarantee the strong consistency. We thus set

$$s^{(0)}(v^{\mathcal{N}}(t), w^{\mathcal{N}}) = \sum_{K \in \mathcal{T}_h} \delta_K \left(\partial_t v^{\mathcal{N}}(t) + L v^{\mathcal{N}}(t), \frac{h_K}{|\boldsymbol{\beta}|}(L_{SS} + \rho L_S) w^{\mathcal{N}} \right)_K \tag{3.45}$$

where $v^{\mathcal{N}}(t) \in X^{\mathcal{N}}$ for each $t \in I$ and $w^{\mathcal{N}} \in X^{\mathcal{N}}$.

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 2.2.

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

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\
&\frac{1}{\Delta t} m_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g(t^j) F_{stab}(v^{\mathcal{N}}) \\
&\qquad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.} \\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}.
\end{aligned} \tag{3.46}$$

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

$$\begin{aligned}
m_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= m(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) \; + \sum_{K_o(\boldsymbol{\mu}) \in \mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \left(v^{\mathcal{N}}, \frac{h_{K_o(\boldsymbol{\mu})}}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L_{SS}\, w^{\mathcal{N}} \right)_{K_o(\boldsymbol{\mu})} \\
a_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= a(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) \; + \sum_{K_o(\boldsymbol{\mu}) \in \mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \left(L v^{\mathcal{N}}, \frac{h_{K_o(\boldsymbol{\mu})}}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L_{SS}\, w^{\mathcal{N}} \right)_{K_o(\boldsymbol{\mu})} \\
F_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= F(v^{\mathcal{N}}; \boldsymbol{\mu}) \; + \sum_{K_o(\boldsymbol{\mu}) \in \mathcal{T}_{h,o}(\boldsymbol{\mu})} \delta_{K_o(\boldsymbol{\mu})} \left(f, \frac{h_{K_o(\boldsymbol{\mu})}}{|\boldsymbol{\beta}(\boldsymbol{\mu})|} L_{SS}\, w^{\mathcal{N}} \right)_{K_o(\boldsymbol{\mu})}
\end{aligned} \tag{3.47}$$

where $K_o(\boldsymbol{\mu})$ are the elements which form the mesh $\mathcal{T}_{h,o}$ defined on the original domain Ω_o (see sections 1.1.1 and 2.2.2).

For the analysis of stability and convergence of this method, we refer to [3, 5, 29].

3.3 Numerical results

We are showing now some numerical tests of the stabilized RB method for parabolic PDEs. The first one, discussed in section 3.3.1 is the time dependent version of the problem studied in section 2.3, while the second test case, section 3.3.2, is a time-dependent Poiseuille-Graetz problem.

3.3.1 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 3.1. Moreover, let us denote with I the time interval $[0, T]$.

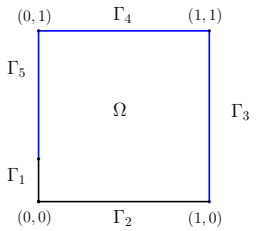


Figure 3.1: Domain of the problem (3.48). On the blue sides we impose $u = 0$, while on the black ones $u = g$.

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:

$$\begin{cases} \partial_t u - \dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + (\cos\mu_2, \sin\mu_2)\cdot\nabla u(\boldsymbol{\mu}) = 0 & \text{in } \Omega \times I \\ u(\cdot, t; \boldsymbol{\mu}) = g(t) & \text{on } \Gamma_1 \cup \Gamma_2,\ \forall t \in I \\ u(\cdot, t; \boldsymbol{\mu}) = 0 & \text{on } \Gamma_3 \cup \Gamma_4 \cup \Gamma_5,\ \forall t \in I, \\ u(\cdot, 0; \boldsymbol{\mu}) = 0 & \text{on } \Omega, \end{cases} \tag{3.48}$$

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 (2.17)). 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 (3.46) in which, for all $v^{\mathcal{N}}, w^{\mathcal{N}} \in X^{\mathcal{N}}$, we have:

$$\begin{aligned} m_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= \int_\Omega v^{\mathcal{N}} w^{\mathcal{N}} + \sum_{K\in\mathcal{T}_h} h_K \left(v^{\mathcal{N}}, (\cos\mu_2, \sin\mu_2)\cdot\nabla w^{\mathcal{N}}\right)_K \\ a_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) &= \int_\Omega \frac{1}{\mu_1}\nabla v^{\mathcal{N}}\cdot\nabla w^{\mathcal{N}} + (\cos\mu_2, \sin\mu_2)\cdot\nabla v^{\mathcal{N}}\, w^{\mathcal{N}} \\ &\quad + \sum_{K\in\mathcal{T}_h} h_K \left((\cos\mu_2, \sin\mu_2)\cdot\nabla v^{\mathcal{N}}, (\cos\mu_2, \sin\mu_2)\cdot\nabla w^{\mathcal{N}}\right)_K \\ F_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= \sum_{K\in\mathcal{T}_h} h_K \left(f_h, (\cos\mu_2, \sin\mu_2)\cdot\nabla\, w^{\mathcal{N}}\right)_K \end{aligned} \tag{3.49}$$

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

$$\delta_K = 1 \quad \forall\, K \in \mathcal{T}_h. \tag{3.50}$$

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 "tuning" parameter M_1 in the POD-greedy has been set using (3.29), assuming a tolerance $\varepsilon_{tol}^{POD} = 0.1$. The other "tuning" parameter, M_2, has been chosen equal to 1. In table 3.1 we report informations about the computational time. As in the steady case, 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. In figures 3.2 and 3.3 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 3.2, we show the RB solution (computed at some time-steps) of (3.48) obtained using a constant control function $g \equiv 1$. In figure 3.3 we show the solution corresponding to the control function $g(t) = \sin(\frac{4}{5}\pi t)$, for all $t \in [0, T]$.

3.3.2 Time dependent Poiseuille-Graetz problem

In this section we want to test the stabilized RB method for a time dependent Poiseuille-Graetz problem [15, 27, 45, 51]. We have already dealt with the steady case of this problem in section 2.2.2.

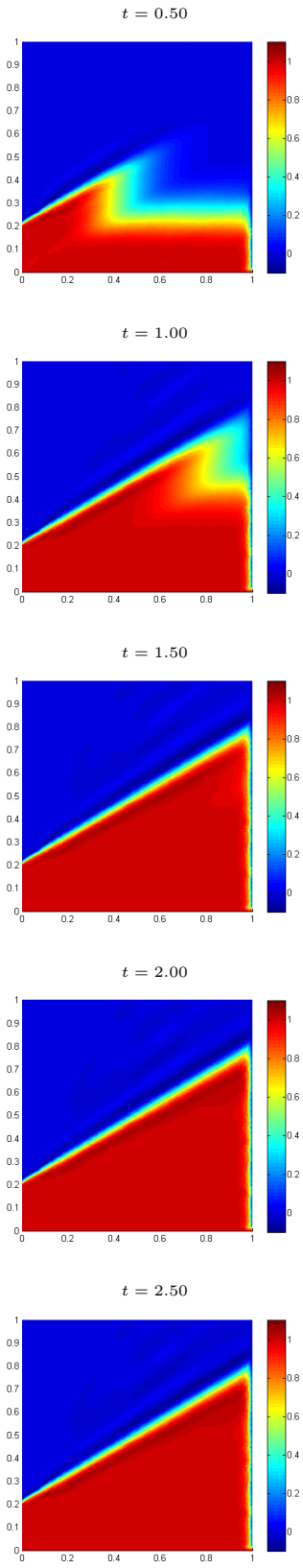


Figure 3.2: RB solution of (3.48), with $g(t) = 1$ for all $t \in [0, T]$, for a parameter value $\boldsymbol{\mu} = (10^5, \frac{\pi}{6})$.

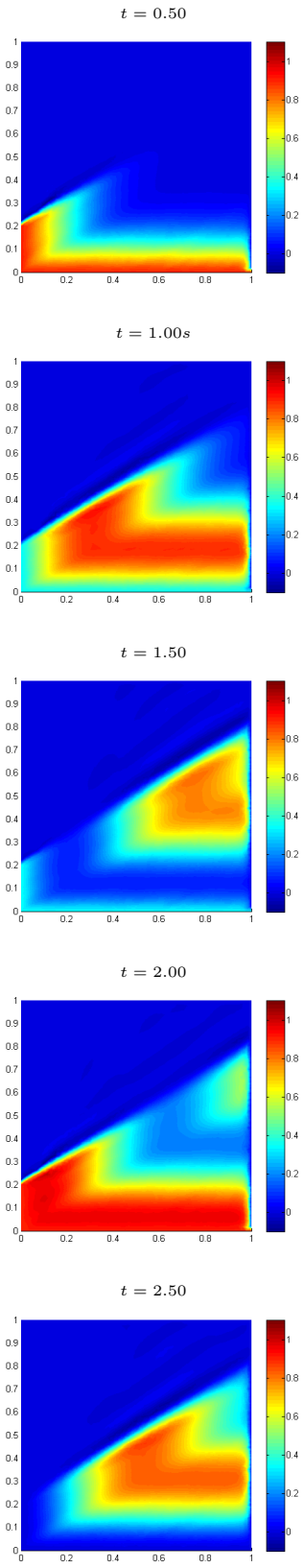


Figure 3.3: RB solution of (3.48), with $g(t) = \sin(\frac{4}{5}\pi t)$ for all $t \in [0, T]$, for a parameter value $\boldsymbol{\mu} = (10^5, \frac{\pi}{6})$.

$\mu_1 \in$	$\mu_2 \in$	Offline time (s)	N	Online time (s)
$\{10^5\}$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2346	28	0.100
$[10^4, 10^5]$	$[\frac{\pi}{6}, \frac{\pi}{3}]$	2857	69	0.191
$[10^4, 10^5]$	$\{\frac{\pi}{4}\}$	339	15	0.067

Table 3.1: First time dependent test case. Numerical tests

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 3.4. 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 $\Gamma_{o\,i}$, $i = 1, \dots, 6$. We then define I the time interval $[0, T]$.

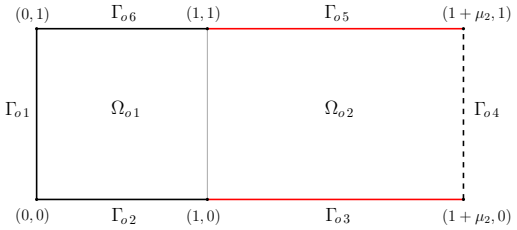


Figure 3.4: Domain of the problem (3.48). On the black sides we impose $u = g_1$, whereas on the red ones $u = g_2$.

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

$$\begin{cases} \partial_t u(\boldsymbol{\mu}) - \dfrac{1}{\mu_1}\Delta u(\boldsymbol{\mu}) + 4\,y(1-y)\partial_x u(\boldsymbol{\mu}) = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \\ u(\cdot, t; \boldsymbol{\mu}) = g_1(t) & \text{on } \Gamma_{o\,1}(\boldsymbol{\mu}) \cup \Gamma_{o\,2(\boldsymbol{\mu})} \cup \Gamma_{o\,6}(\boldsymbol{\mu}),\ \forall\, t \in I, \\ u(\cdot, t; \boldsymbol{\mu}) = g_2(t) & \text{on } \Gamma_{o\,3}(\boldsymbol{\mu}) \cup \Gamma_{o\,5}(\boldsymbol{\mu}),\ \forall\, t \in I, \\ \dfrac{\partial u}{\partial \nu}(\cdot, t; \boldsymbol{\mu}) = 0 & \text{on } \Gamma_{o\,4}(\boldsymbol{\mu}),\ \forall\, t \in I, \\ u(\cdot, 0; \boldsymbol{\mu}) = 1 & \text{on } \Omega_o(\boldsymbol{\mu}). \end{cases} \tag{3.51}$$

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 $\Gamma_i = \Gamma_{o\,i}(\boldsymbol{\mu})$, $i = 1, \dots, 6$. 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 2.2.2 by defining its restrictions on the subdomains Ω_i, $i = 1, 2$ (see (2.60) and (2.61)). 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:

$$f_h^1|_{\Gamma_1 \cap \Gamma_2 \cap \Gamma_6} \equiv 1 \qquad f_h^2|_{\Gamma_3 \cap \Gamma_5} \equiv 1 \tag{3.52}$$

Like in section 2.2.2, we can write the weak formulation of the problem (3.51) and then track it back on the reference domain. We can thus obtain the following Backward-Euler/stabilized

FE problem:

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } u^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\
&\frac{1}{\Delta t} m_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}) - u^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(u^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) \\
&\qquad = g_1(t^j) F^1_{stab}(v^{\mathcal{N}}) + g_2(t^j) F^2_{stab}(v^{\mathcal{N}}) \\
&\qquad\quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \\
&\text{given the initial condition } u^{\mathcal{N}\,0} \text{ s.t.} \\
&(u^{\mathcal{N}\,0}, v^{\mathcal{N}})_{L^2(\Omega)} = (u_0, v^{\mathcal{N}})_{L^2(\Omega)} \quad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}
\end{aligned} \tag{3.53}$$

where, in the left-hand side:

$$\begin{aligned}
m_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) = &\int_{\Omega^1} v^{\mathcal{N}} w^{\mathcal{N}} + \sum_{K \in \mathcal{T}_h^1} h_K \int_K v^{\mathcal{N}} \partial_x w^{\mathcal{N}} \\
&+ \int_{\Omega^2} \frac{\mu_2}{\mu_1} v^{\mathcal{N}} w^{\mathcal{N}} + \sum_{K \in \mathcal{T}_h^1} \frac{h_K}{\sqrt{\mu_2}} \int_K v^{\mathcal{N}} \partial_x w^{\mathcal{N}} \\
a_{stab}(v^{\mathcal{N}}, w^{\mathcal{N}}; \boldsymbol{\mu}) = &\int_{\Omega^1} \frac{1}{\mu_1} \nabla v^{\mathcal{N}} \cdot \nabla w^{\mathcal{N}} + 4\, y(1-y) \partial_x v^{\mathcal{N}}\, w^{\mathcal{N}} \\
&+ \sum_{K \in \mathcal{T}_h^1} h_K \int_K \left(4\, y(1-y) \partial_x v^{\mathcal{N}}\right) \partial_x w^{\mathcal{N}} \\
&+ \int_{\Omega^2} \frac{1}{\mu_1 \mu_2} \partial_x v^{\mathcal{N}} \partial_y w^{\mathcal{N}} + \frac{\mu_2}{\mu_1} \partial_x v^{\mathcal{N}} \partial_y w^{\mathcal{N}} + 4\mu_2\, y(1-y) \partial_x v^{\mathcal{N}}\, w^{\mathcal{N}} \\
&+ \sum_{K \in \mathcal{T}_h^2} \frac{h_K}{\sqrt{\mu}_2} \int_K \left(4\, y(1-y) \partial_x w^{\mathcal{N}}\right) \partial_x v^{\mathcal{N}}.
\end{aligned} \tag{3.54}$$

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

$$\begin{aligned}
F^1_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= - a_{stab}(f_h^1, v^{\mathcal{N}}; \boldsymbol{\mu}) \\
F^2_{stab}(v^{\mathcal{N}}; \boldsymbol{\mu}) &= - a_{stab}(f_h^2, v^{\mathcal{N}}; \boldsymbol{\mu}),
\end{aligned} \tag{3.55}$$

for all $v^{\mathcal{N}}, w^{\mathcal{N}} \in X^{\mathcal{N}}$. The weighting has been chosen as in section 2.2.2.

In order to apply the RB method exposed in section 3.1, we exploit the linearity of the problem and we consider the two problems:

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } \varphi^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\
&\frac{1}{\Delta t} m_{stab}(\varphi^{\mathcal{N}\,j}(\boldsymbol{\mu}) - \varphi^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(\varphi^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g_1(t^j) F^1_{stab}(v^{\mathcal{N}}) \\
&\qquad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \\
&\text{given the initial condition } \theta u^{\mathcal{N}\,0}
\end{aligned} \tag{3.56}$$

and

$$\begin{aligned}
&\text{for each } 1 \le j \le J, \text{ find } \psi^{\mathcal{N}\,j}(\boldsymbol{\mu}) \in X^{\mathcal{N}} \text{ s.t.} \\
&\frac{1}{\Delta t} m_{stab}(\psi^{\mathcal{N}\,j}(\boldsymbol{\mu}) - \psi^{\mathcal{N}\,j-1}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) + a_{stab}(\psi^{\mathcal{N}\,j}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = g_1(t^j) F^1_{stab}(v^{\mathcal{N}}) \\
&\qquad \forall v^{\mathcal{N}} \in X^{\mathcal{N}}, \\
&\text{given the initial condition } (1-\theta) u^{\mathcal{N}\,0},
\end{aligned} \tag{3.57}$$

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 (3.56) and (3.57), respectively, then $\boldsymbol{\varphi}^{\mathcal{N}}(\boldsymbol{\mu}) + \boldsymbol{\psi}^{\mathcal{N}}(\boldsymbol{\mu})$ is a solution of (3.53).

Once we have this "separation" of the problem, we can apply the RB method to (3.56) and (3.57) separately. We then define the RB solution of (3.53) $u_N^{\mathcal{N}}(\boldsymbol{\mu}) := \varphi_N^{\mathcal{N}}(\boldsymbol{\mu}) + \psi_N^{\mathcal{N}}(\boldsymbol{\mu})$. Concerning the RB approximation error, the triangular inequality implies that:

$$|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \leq |||\varphi^{\mathcal{N}}(\boldsymbol{\mu}) - \varphi_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} + |||\psi^{\mathcal{N}}(\boldsymbol{\mu}) - \psi_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep}. \quad (3.58)$$

In our numerical tests we have used $\mathcal{D} = [10000, 20000] \times [0.5, 4]$, $T = 5$, $J = 100$ and $\theta = 0$. 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 (3.56) (Offline computational time: $5773\,s$) and $N^2 = 50$ basis for the problem (3.57) (Offline computational time: $1658\,s$). The tolerance on the greedy algorithm is $\varepsilon_{tol}^* = 10^{-2}$.

In figure 3.5, we show the RB solution of (3.53) for $\boldsymbol{\mu} = (15000, 2)$, computed at some time steps. Here we used the following control functions:

$$\begin{aligned} g_1(t) &= e^{-t} \quad \forall t \in I, \\ g_2(t) &= 1 \quad \forall t \in I. \end{aligned} \quad (3.59)$$

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

$$|||\varphi^{\mathcal{N}}(\boldsymbol{\mu}) - \varphi_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \leq 0.058, \qquad |||\psi^{\mathcal{N}}(\boldsymbol{\mu}) - \psi_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \leq 0.047, \quad (3.60)$$

then for the total RB approximation error holds

$$|||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})|||_{t-dep} \leq 0.105. \quad (3.61)$$

This error has the same order of magnitude as the time dependent SUPG approximation error, which is bounded by $C(h^3 + \Delta t^2)^{\frac{1}{2}}$ [29]. The computational time of the Online stage is $0.255\,s$.

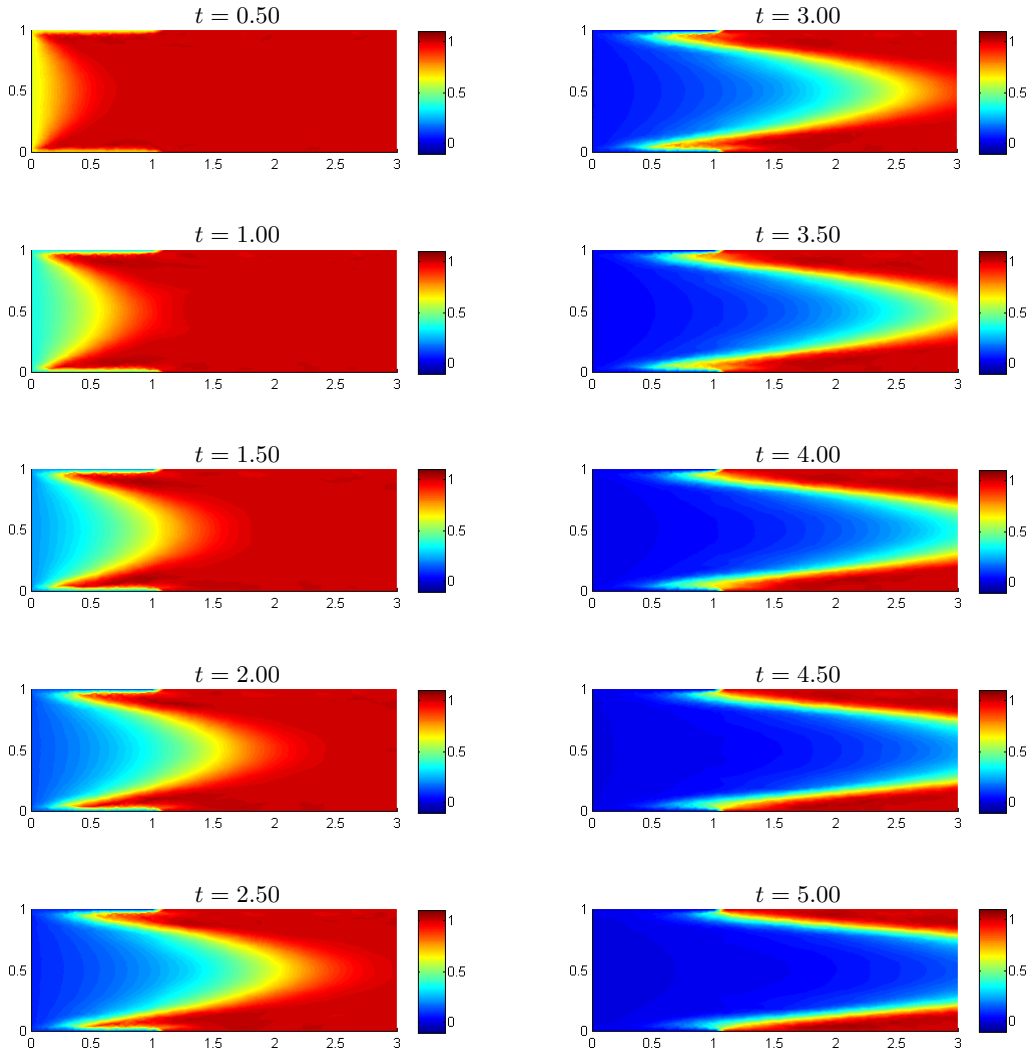


Figure 3.5: Time dependent Poiseuille-Graetz problem. RB solution computed at some time steps.

Conclusions

In this thesis 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.

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. In the former, we have used the same SUPG [48] stabilized bilinear form in both Offline and Online stages, whereas in the latter we have performed the Offline stage using the SUPG stabilized form and the Offline stage with the original (unstabilized) form. The underlying idea was to obtain stable RB solution combining the stable reduced basis produced in the Offline stage.

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 introduced in section 1.2.3 are still effective. As regards the *Offline-only* method, we have observed strong instability phenomena in the RB solution and we have showed that this is because of "inconsistency" problems arising from the use of different bilinear forms in the two stages of the RB method. To carry out this analysis, we have tested both method using test cases whose FE approximated solutions show significant instability effects.

Having determined which stabilization strategy is the best one, 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 [3, 4]. Considering what we have showed in the steady case, we have proposed to use the same stabilized form in both Offline and Online stage. The method have been successfully tested on some test problem, in particular on an unsteady Poiseuille-Graetz problem with time dependent boundary conditions.

A natural continuation of this work can be the application of these stabilization strategies to problems with more complex affine geometries, in order to understand if strong 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, 34], in order to obtain a suitable RB formulation.

The proposed tests and methodology could be also used as the first step to study non-linear transport problems, i.e. Burgers' equation.

Bibliography

[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. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.*, 43(3):1457–1472, 2011.

[3] 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.

[4] 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).

[5] 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.

[6] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodríguez. Improved successive constraint method based a posteriori error estimate for reduced basis approximation of 2D Maxwell's problem. *M2AN Math. Model. Numer. Anal.*, 43(6):1099–1116, 2009.

[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.W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[10] 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.

[11] J.L. Eftang, D.B.P. Huynh, D.J. Knezevic, and A.T. Patera. A two-step certified reduced basis method. *J. Sci. Comput.*, 51(1):28–58, 2012.

[12] J.L. Eftang, D.J. Knezevic, and A.T. Patera. An *hp* certified reduced basis method for parametrized parabolic partial differential equations. *Math. Comput. Model. Dyn. Syst.*, 17(4):395–422, 2011.

[13] J.L. Eftang, A.T. Patera, and E.M. Rønquist. An "*hp*" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.*, 32(6):3170–3200, 2010.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] M.D. Gunzburger. *Perspectives in flow control and optimization*, volume 5 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003.

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

[20] 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.

[21] 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.

[22] T.J.R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics. VIII. The Galerkin/least-squares method for advective-diffusive equations. *Comput. Methods Appl. Mech. Engrg.*, 73(2):173–189, 1989.

[23] D.B.P. Huynh, D.J. Knezevic, Y. Chen, J.S. Hesthaven, and A.T. Patera. A natural-norm successive constraint method for inf-sup lower bounds. *Comput. Methods Appl. Mech. Engrg.*, 199(29-32):1963–1975, 2010.

[24] 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.

[25] 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.

[26] 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.

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

[28] K. Ito and S.S. Ravindran. A reduced-order method for simulation and control of fluid flows. *J. Comput. Phys.*, 143(2):403–425, 1998.

[29] 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.

[30] 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.

[31] 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.

[32] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numer. Math.*, 90(1):117–148, 2001.

[33] T. Lassila, A. Manzoni, and G. Rozza. On the approximation of stability factors for general parametrized partial differential equations with a two-level affine decomposition. *M2AN Math. Model. Numer. Anal.*, 46:1555–1576, 2012.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

[41] 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`,.

[42] T.A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Math. Comp.*, 45(172):487–496, 1985.

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

[44] 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.

[45] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *J. Math. Ind.*, 1:Art. 3, 44, 2011.

[46] 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.

[47] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer-Verlag, Berlin, second edition, 2007.

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

[49] 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*, S. Francisco, CA, USA, 2009.

[50] 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.

[51] 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*, S. Francisco, CA, USA, 2009.

[52] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.

[53] 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.

[54] 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.

[55] K. Urban and A.T. Patera. An improved error bound for reduced basis approximation of linear parabolic problems. Submitted to Mathematics of Computation. June 2012.

Desidero innanzitutto ringraziare la Prof. Ilaria Perugia ed il Dr. Ing. Gianluigi Rozza per il loro supporto nella preparazione di questa tesi. Ringrazio il gruppo del CMCS-MATHICSE dell'EPFL per l'accoglienza e la collaborazione durante il mio periodo di permanenza a Losanna, reso possibile grazie al progetto Erasmus Student Placement dell'Università di Pavia.

Un ringraziamento speciale va ai miei compagni di corso, per essere stati degli amici più che semplici "colleghi di lavoro". Grazie a Denver per avermi sempre dato retta tutte le volte che l'ho tediato (sto usando un eufemismo) a ogni ora del giorno. Grazie ad Andrea per la sua ospitalità e per molto altro. Grazie a Matteo per la compagnia durante i lunghi e altrimenti noiosi viaggi in pullman. Una menzione particolare è per Marco, in quanto è l'unico ad avere seguito tutti i miei stessi corsi della Magistrale ed è stato un vero sostegno nella preparazione degli esami. Grazie poi a Nicolò, Kinder, Lara, Laura, Michele, Eleonora, Diletta... Con voi, persino la nave è potuta diventare un luogo piacevole.

Un ringraziamento ed un saluto vanno anche a tutte le persone che ho avuto il piacere di conoscere a Losanna. Grazie in particolare a Laura per avermi fatto da "balia" durante i miei primi giorni lì e per essere stata una fantastica compagna di ufficio. Grazie a Ricardo, altro eccezionale compagno di ufficio. Grazie poi a Federico per il supporto con LaTeX(ed i preziosissimi template). Grazie ovviamente anche a Eleonora, Tommaso, Anna, Matteo, Francesco, Francesca, Wibke, Cristiano, Luca, Simone, Andrea, Toni, Peng, Matteo, Alessandro, Laura e a tutti gli altri. Spero di rivedervi al più presto.

Grazie ai miei amici di Vidardo (e anche di Castiraga), che ho la fortuna (?) di conoscere da sempre. Grazie per essere in grado di mettere alla prova come nessun altro le mie capacità di sopportazione.

Il ringraziamento finale è per i miei genitori, per avermi "sponsorizzato", sostenuto e sopportato (soprattutto) per 24 anni. Una prova di resistenza notevole, davvero.

Ringrazio infine:

- *la biblioteca del dimat,*
- *la macchinetta del caffè del dimat,*
- *il bar della nave,*
- *gli appunti di Denver.*